SIEMENS

SINUMERIK

SINUMERIK 840D sl
NC variables

List Manual

Preface

Fundamental safety instructions 1

Introduction 2

NC variables 3

Appendix A A

Valid for
Control
SINUMERIK 840D sl / 840DE sl
Software
CNC Software, Version 4.92

06/2019
A5E40870419B AC

Legal information

Warning notice system

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

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

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

CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

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

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

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

Trademarks

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

Disclaimer of Liability

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

Copyright © Siemens AG 2005 - 2018.
All rights reserved

Preface

SINUMERIK documentation

The SINUMERIK documentation is organized into the following categories:

- General documentation/catalogs
- User documentation
- Manufacturer/service documentation

Additional information

You can find information on the following topics at the following address (https://support.industry.siemens.com/cs/de/en/view/108464614):

- Ordering documentation/overview of documentation
- Additional links to download documents
- Using documentation online (find and search in manuals/information)

If you have any questions regarding the technical documentation (e.g. suggestions, corrections), please send an e-mail to the following address (mailto:docu.motioncontrol@siemens.com).

mySupport/Documentation

At the following address (https://support.industry.siemens.com/My/ww/en/documentation), you can find information on how to create your own individual documentation based on Siemens' content, and adapt it for your own machine documentation.

Training

At the following address (http://www.siemens.com/sitrain), you can find information about SITRAIN (Siemens training on products, systems and solutions for automation and drives).

FAQs

You can find Frequently Asked Questions in the Service&Support pages under Product Support (https://support.industry.siemens.com/cs/de/en/ps/faq).

SINUMERIK

You can find information about SINUMERIK at the following address (http://www.siemens.com/sinumerik).

Target group

This publication is intended for project engineers, commissioning engineers, machine operators and service and maintenance personnel.

Benefits

The intended target group can use the Parameter Manual to test and commission the system or the plant correctly and safely.

Utilization phase: Setup and commissioning phase

Standard scope

This documentation describes the functionality of the standard scope. Extensions or changes made by the machine manufacturer are documented by the machine manufacturer.

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

Furthermore, for the sake of clarity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation or maintenance.

Note regarding the General Data Protection Regulation

Siemens observes standard data protection principles, in particular the principle of privacy by design. That means that

this product does not process / store any personal data, only technical functional data (e.g. time stamps). If a user links this data with other data (e.g. a shift schedule) or stores personal data on the same storage medium (e.g. hard drive) and thus establishes a link to a person or persons, then the user is responsible for ensuring compliance with the relevant data protection regulations.

Technical Support

Country-specific telephone numbers for technical support are provided in the Internet at the following address (https://support.industry.siemens.com/sc/ww/en/sc/2090) in the "Contact" area.

Table of contents

Preface 3

1 Fundamental safety instructions 9

1.1 General safety instructions 9

1.2 Warranty and liability for application examples 9

1.3 Industrial security 10

2 Introduction 13

2.1 Overview 13

2.2 References 13

3 NC variables 15

3.1 Explanations on the NC variables 15
3.1.1 NC areas 15
3.1.2 Data blocks 15
3.1.3 Variable types 18
3.1.4 Data types 20
3.1.5 Structure of the data tables 21

3.2 NC variable descriptions 21
3.2.1 Axis: Axis-Specific Basic Settings 21
3.2.1.1 M: Machine data 21
3.2.1.2 SE: Setting data 79
3.2.2 Bag: Mode Group Data 85
3.2.2.1 S: State data 85
3.2.3 Channel: Channel-assigned data 88
3.2.3.1 AUXFU: Active auxiliary functions 88
3.2.3.2 CP: Generic coupling 92
3.2.3.3 DIAG: Diagnostics data, available only for internal development purposes 105
3.2.3.4 DIAGN: Diagnostic module 111
3.2.3.5 EPSP: Reserved for internal use only (ePS) 117
3.2.3.6 ETP: Event types 120
3.2.3.7 FA: Active Work Offset 129
3.2.3.8 FB: Base Frame 131
3.2.3.9 FE: External Work Offset 133
3.2.3.10 FG: Frames for grinding applications 135
3.2.3.11 FS: System Frame 137
3.2.3.12 FU: Settable Work Offset 139
3.2.3.13 M: Machine data 141
3.2.3.14 NIB: Nibbling data 189
3.2.3.15 O: Internal 192
3.2.3.16 PA: Protection zones 192
3.2.3.17 RP: Arithmetic parameter 211
3.2.3.18 S: State data 212
3.2.3.19 SE: Setting data 365

3.2.3.20 SEGA: Extended state in the WCS....368
3.2.3.21 SEMA: Extended state data in MCS....390
3.2.3.22 SGA: State data in the WCS....482
3.2.3.23 SINF: Part program-specific state data....487
3.2.3.24 SMA: State data in the MCS....493
3.2.3.25 SNCF: Active G-functions....496
3.2.3.26 SPARP: Part program information....498
3.2.3.27 SPARPF: Search pointer for block search....513
3.2.3.28 SPARPI: Program pointer at interruption....518
3.2.3.29 SPARPP: Program pointer in automatic operation mode....523
3.2.3.30 SSP: State data spindle....530
3.2.3.31 SSP2: Spindle status data for spindle conversion....546
3.2.3.32 SSYNAC: Synchronous actions....562
3.2.3.33 SYNACT: Channel-specific synchronous actions....566
3.2.3.34 TO: Tool offsets....570
3.2.3.35 VSYN: NCK-specific user variable for synchronous actions....571
3.2.3.36 WAL: Working area limitation....573
3.2.3.37 Y: System data....575
3.2.4 DriveHsa: Main spindle drive data....587
3.2.4.1 S: State data....587
3.2.5 Drive: HMI data....588
3.2.5.1 S: State data....588
3.2.6 Nck: NC data....589
3.2.6.1 CP: Generic coupling....589
3.2.6.2 DIAG: Diagnostics data, available only for internal development purposes....593
3.2.6.3 DIAGN: Diagnostic module....598
3.2.6.4 ETPD: Data list for protocolling....635
3.2.6.5 FA: Active Work Offset....637
3.2.6.6 FB: Base Frame....639
3.2.6.7 FG: Frames for grinding applications....641
3.2.6.8 FU: Settable Work Offset....643
3.2.6.9 M: Machine data....645
3.2.6.10 O: Internal....687
3.2.6.11 PA: Protection zones....690
3.2.6.12 RP: Arithmetic parameter....744
3.2.6.13 S: State data....745
3.2.6.14 SALA: Alarm list, oldest alarm....866
3.2.6.15 SALAC: Alarm actions, oldest alarm....869
3.2.6.16 SALAL: Alarm list, most recent alarm....873
3.2.6.17 SALAP: Alarm list, highest priority alarm....876
3.2.6.18 SD: Servo data....879
3.2.6.19 SE: Setting data....975
3.2.6.20 SEMA: Extended state data in MCS....983
3.2.6.21 SMA: State data in the MCS....1073
3.2.6.22 SSP: State data spindle....1077
3.2.6.23 SSP2: Spindle status data for spindle conversion....1092
3.2.6.24 VSYN: NCK-specific user variable for synchronous actions....1107
3.2.6.25 Y: System data....1107
3.2.6.26 YFAFL: NCK instruction group list (Fanuc)....1136
3.2.6.27 YNCFL: NCK instruction group list....1137
3.2.7 Tool: Tool Data....1138
3.2.7.1 AD: Adapter data (not for new developments)....1138

3.2.7.2 ADN: Adapter data....1139
3.2.7.3 AEV: Working offsets: Directory....1147
3.2.7.4 MTAD: Application-specific multitool data....1150
3.2.7.5 MTAP: Application-specific multitool place data....1151
3.2.7.6 MTD: Multitool data, general data....1151
3.2.7.7 MTP: Multitool data, place data....1157
3.2.7.8 MTUD: Multitool data, user-defined data....1158
3.2.7.9 MTUP: Multitool place user data....1159
3.2.7.10 MTV: Multitool data, directory....1159
3.2.7.11 TAD: Application-specific data....1162
3.2.7.12 TAM: Application-specific magazine data....1163
3.2.7.13 TAMD: Application-specific magazine data double....1163
3.2.7.14 TAO: Application-specific tool edge data....1164
3.2.7.15 TAP: Application-specific magazine location data....1165
3.2.7.16 TAPD: Application-specific magazine location data double....1166
3.2.7.17 TAS: Application-specific monitoring data....1167
3.2.7.18 TC: Toolholder parameters....1168
3.2.7.19 TD: Tool data, general data....1190
3.2.7.20 TDC: Tool parameters of the Siemens application....1199
3.2.7.21 TF: Used as parameter and return value of the PI-Services _N_TMGETT and _N_TSEARC....1200
3.2.7.22 TG: Tool data, grinding-specific data....1211
3.2.7.23 TISO: ISO tool offset data....1215
3.2.7.24 TM: Magazine data, general data....1215
3.2.7.25 TMC: Magazine data, configuration data....1223
3.2.7.26 TMV: Magazine data, directory....1232
3.2.7.27 TO: Tool offsets....1234
3.2.7.28 TOE: Edge-related location-dependent coarse total offsets, setup offsets....1236
3.2.7.29 TOET: Edge-related location-dependent coarse total offsets, transformed setup offsets....1237
3.2.7.30 TOS: Edge-related positional fine total offsets....1239
3.2.7.31 TOST: Edge-related location-dependent total offsets, transformed....1241
3.2.7.32 TOT: Edge data, transformed offset data....1243
3.2.7.33 TP: Magazine location data....1247
3.2.7.34 TPM: Magazine data, multiple assignment of place data....1251
3.2.7.35 TS: Tool monitoring data....1252
3.2.7.36 TT: Magazine location data....1253
3.2.7.37 TU: Tool data for OEM....1254
3.2.7.38 TUE: Tool edge data for OEM....1255
3.2.7.39 TUM: Mgazine user data....1256
3.2.7.40 TUMD: User magazine data double....1256
3.2.7.41 TUP: Magazine place user data....1257
3.2.7.42 TUPD: User magazine location data double....1257
3.2.7.43 TUS: User monitoring data....1258
3.2.7.44 TV: Tool data, directory....1259
3.2.8 DriveVsa: Feed drive data....1262
3.2.8.1 M: Machine data....1262
3.2.8.2 S: State data....1263

A Appendix A....1265

A.1 List of abbreviations....1265

Fundamental safety instructions 1

1.1 General safety instructions

WARNING
Danger to life if the safety instructions and residual risks are not observed If the safety instructions and residual risks in the associated hardware documentation are not observed, accidents involving severe injuries or death can occur. • Observe the safety instructions given in the hardware documentation. • Consider the residual risks for the risk evaluation.

WARNING
Malfunctions of the machine as a result of incorrect or changed parameter settings As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death. • Protect the parameterization against unauthorized access. • Handle possible malfunctions by taking suitable measures, e.g. emergency stop or emergency off.

1.2 Warranty and liability for application examples

Application examples are not binding and do not claim to be complete regarding configuration, equipment or any eventuality which may arise. Application examples do not represent specific customer solutions, but are only intended to provide support for typical tasks.

As the user you yourself are responsible for ensuring that the products described are operated correctly. Application examples do not relieve you of your responsibility for safe handling when using, installing, operating and maintaining the equipment.

1.3 Industrial security

Note

Industrial security

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Products and solutions from Siemens constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. using firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that can be implemented, please visit:

Industrial security (https://www.siemens.com/industrialsecurity)

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they become available, and that only the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at:

Industrial security (https://www.siemens.com/industrialsecurity)

Further information is provided on the Internet:

Industrial Security Configuration Manual (https://support.industry.siemens.com/cs/ww/en/view/108862708)

WARNING
Unsafe operating states resulting from software manipulation Software manipulations, e.g. viruses, Trojans, or worms, can cause unsafe operating states in your system that may lead to death, serious injury, and property damage. • Keep the software up to date. • Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine. • Make sure that you include all installed products into the holistic industrial security concept. • Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners. • On completion of commissioning, check all security-related settings. • Protect the drive against unauthorized changes by activating the "Know-how protection" converter function.

Introduction 2

2.1 Overview

General

The NC variables can be accessed from the user interface or from the PLC via the operator panel interface (OPI).

The structure of the NC variables and their processing is described in Section Explanations on the NC variables (Page 15).

The tables of the NC variables contain references to further manuals with detailed information.

The references to manuals are listed in the list of manuals.
You will find more information in Chapter: References (Page 13).

2.2 References

References

The variable table has a dedicated "Lit" field for references to additional literature.

As a cross reference to the functional description of the data, the "Lit" field contains the short designation of the corresponding chapter of a function manual respectively the short designation of a manual.

Examples:

W1	Function Manual Tools, Submanual W1: Tool offset
FBWsl	Function Manual, Tool Management

List of manuals

Reference is made to the following chapters respectively manuals:

- Function Manual Basic Functions, Chapter: A2, A4, B1, B3, H2, K1, K2, K5, K7, K10, N2, S7, V2, TE7
- Function Manual Axes and Spindles, Chapter: B2, F1, G1, G2, H1, M3, N3, P1, P2, P5, R1, R2, R3, S1, S3, S9, T1, T4, T5, T6, V1
- Function Manual Technologies, Chapter: M5, N4, R3, T3, TE01, TE02, TE1, TE3, TE4, TE6, TE7, TE8
- Function Manual Tools, Chapter: W1, W4, W5
- Function Manual Transformations, Chapter: F2, K12, M1

- Function Manual Monitoring and Compensation, Chapter: A3, A5, K3, K6, K8, K9, K11, TE1, TE9
- Safety Integrated Commissioning Manual, FBSI
- Function Manual Tool Management, FBWsl
- Function Manual Synchronous Actions, FBSY
- Description of Functions, ISO Dialects for SINUMERIK, FBFA
- Programming Manual, PG
- Programming Manual Job Planning, PGA
- Basic Software and Operating Software Commissioning Manual, IM9, BE2, IM7, IM10
- System Manual Ctrl-Energy, SCE

NC variables 3

3.1 Explanations on the NC variables

3.1.1 NC areas

NC areas

The NC variables are organized in data blocks that are assigned to the following NC areas.

Table 3-1 Assignment of the TO areas

Area	NC variable
NC (N)	Contains all variables that apply to the entire NC, e.g. • System data (Y) • Protection areas (PA) • G groups (YNCFL), etc.
Mode group (B)	Contains all variables that apply to the mode group, e.g. • Status data (S)
Channel (C)	Contains all variables that apply to the relevant channel, e.g. • System data (Y) • Protection areas (PA) • Global status data (S)
Tool (T)	Contains all variables that apply to the tools on the machine, e.g. • Tool offset data (TO) • General tool data (TD) • Tool monitoring data (TS), etc. Each tool area (T) is assigned to a channel.
Axis (A)	Contains machine and setting data that applies to the relevant axis or spindle. See List Manual 1, Section: Axis-specific machine data
Feed/main drive (V/H)	Contains machine data or machine data as service values that apply to the relevant drive.

3.1.2 Data blocks

Overview of the available data blocks

The following table contains an overview of available data blocks for variables of the NC and their assignment to the individual areas.

Only those data blocks whose variables can be read or written with direct access are represented.

Data blocks whose variables can be freely defined by the programmer (e.g. global user data) are read from the operating software or PLC using other mechanisms.

Data block	Area						
	A	B	C	H	N	T	V
AD						x	
MTUD						x	
TAO						x	
TDC						x	
TMV						x	
TOT						x	
TUE						x	
TV						x	
AEV						x	
MTUP						x	
TAP						x	
TF						x	
TO						x	
TP						x	
TUM						x	
MTAD						x	
MTV						x	
TAPD						x	
TG						x	
TOE						x	
TPM						x	
TUMD						x	
MTAP						x	
TAD						x	
TAS						x	
TISO						x	
TOET						x	
TS						x	
TUP						x	
MTD						x	
TAM						x	
TC						x	
TM						x	
TOS						x	
TT						x	
TUPD						x	
MTP						x	

Data block	Area						
	A	B	C	H	N	T	V
TAMD						x	
TD						x	
TMC						x	
TOST						x	
TU						x	
TUS						x	
ETP			x				
ETPD					x		
DIAGN			x				
FA			x		x		
FB			x		x		
FE			x				
FU			x		x		
M	x				x		
NIB			x				
PA			x		x		
RP			x				
S		x	x	x	x		x
SALA					x		
SALAL					x		
SALAP					x		
SE	x		x		x		
SEGA			x				
SEMA			x		x		
SGA			x				
SINF			x				
SMA			x		x		
SNCF			x				
SPARP			x				
SPARPF			x				
SPARPI			x				
SPARPP			x				
SSP			x		x		
SSP2			x		x		
SSYNAC			x				
SYNACT			x				
VSYN		x					
Y		x			x		
YNCFL					x		

References

Further information on which data blocks this mechanism is used, can be found in the following manual: 840D sl PLC Function Manual.

3.1.3 Variable types

Access to an NC variable

In general, the NC variables are stored as structures or arrays of structures (tables) in the areas. This means the following details must be specified in the address for access to an NC variable:

- Area and area number
- Block
- NC variable name (or column number)
- Line number

NC variable type

Generally, a distinction can be made between three variants for the NC variables:

- NC variables that consist of one line
- NC variables that consist of several lines
- NC variables that consist of several columns and lines

Single-line NC variable

A single-line NC variable consists of just a single value. The following information is required to access an NC variable of this type:

- Area (and possibly the area number)
- Block
- NC variable name

Table 3-2 Single-line NC variant type

numMachAxes					
Number of the highest existing channel axis					
-				UWord	r
Multi-line: No					

Example for reading the number of axes in channel 1:

HMI:
/Channel/Configuration/numMachAxes[u1]

HMI:
P_C_Y_numMachAxes

PLC with NC variable selector:

Area	C[.]
Block	Y
NC variable	numMachAxes
Area number	1

Multi-line NC variable

This NC variable is generally defined as a one-dimensional array. The following information is required to access an NC variable of this type:

- Area and possibly the area number
- Block
- NC variable name
- Line number

Table 3-3 Multi-line variant type

actFeedRate					S5
Actual value axial feedrate if the axis is a positioning axis. Actual value single axis feedrate if the axis is a special axis.					
mm/min, inch/min, userdef				Double	r
Multi-line: Yes	Axis number		maxnumGlobMachAxes		

Example for reading the current speed of axis 3 in channel 1:

HMI:
/Channel/MachineAxis/actFeedRate[u1, 3]
HMI:
P_C_SEMA_actFeedRate

PLC with NC variable selector:

Area	C[.]
Block	SEMA
NC variable	actFeedRate[.]
Area number	1
Line	3

Multi-line and multi-column NC variable

This NC variable is generally defined as a two-dimensional array. The following information is required to access an NC variable of this type:

- Area and possibly the area number
- Block

- NC variable name
- Column number
- Line number

In this example, the complete data block consists only of these two-dimensional NC variables.

Table 3-4 Multi-line and multi-column variant type

cuttEdgeParam					
Parameters of the active tool cutting edge					
-	0			Double	r
Multi-line: Yes	Number of the parameter: (EdgeNo - 1) * numCuttEdgeParams + ParameterNo		35		

Example of reading and writing the current cutting edge data for cutting edge 3 / parameter 1 of tool 3 in T area 1.

The example assumes that each tool cutting edge is defined with (numCuttEdgeParams =) 35 parameters:

HMI:
/Tool/Compensation/cuttEdgeParam[u1,c3, 51]

HMI:
P_T_TO_cuttEdgeParam
[y,z] array which is queried in SINUMERIK Operate.

PLC with NC variable selector:

Area	T[.]
Block	TO
NC variable	cuttEdgeParam[.]
Area number	1
Column	3
Line	51

3.1.4 Data types

The control provides the following data types that can be used for programming:

Table 3-5 Data types

Data type	Range of values
BOOLEAN	false, true
BYTE	-128 to +127

Data type	Range of values
UBYTE	0 to +255
DWORD	-2,147,483,648 to +2,147,483,647
UDWORD	0 to +4,294,967,295
DOUBLE	$-1.7*10^{308}$ to $+1.7*10^{308}$
STRING	Sequence of characters (→ UBYTE) with any arbitrary/specified length.

3.1.5 Structure of the data tables

Table fields

Table 3-6 Meaning of table fields

NC variable name (<Range>, <Block>)	Reference to assigned machine data			Lit.
<NC variable brief description/ NC variable description>				
Units and value ranges				
Physical unit:	Data type:	Initial value:	Minimum value:	Maximum value:
Addressing				
Line index description:			Maximum line index:	
			Read and write access	

Lit.	Reference according to the list of references See: Chapter References (Page 13)
Write access	Variable may be overwritten
Read access	Variable can be read

3.2 NC variable descriptions

3.2.1 Axis: Axis-Specific Basic Settings

3.2.1.1 M: Machine data

Axis-specific machine data

AA_OFF_MODE (A, M)	MD 36750: $MA_AA_OFF_MODE	
MD 36750: $MA_AA_OFF_MODE		

AA_OFF_MODE (A, M)		MD 36750: $MA_AA_OFF_MODE		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

ACCEL_REDUCTION_FACTOR (A, M)		MD 35230: $MA_ACCEL_REDUC-TION_FACTOR		
MD 35230: $MA_ACCEL_REDUCTION_FACTOR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

ACCEL_REDUCTION_SPEED_POINT (A, M)		MD 35220: $MA_ACCEL_REDUC-TION_SPEED_POINT		
MD 35220: $MA_ACCEL_REDUCTION_SPEED_POINT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

ACCEL_REDUCTION_TYPE (A, M)		MD 35242: $MA_ACCEL_REDUC-TION_TYPE		
MD 35242: $MA_ACCEL_REDUCTION_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

ACCEL_TYPE_DRIVE (A, M)		MD 35240: $MA_ACCEL_TYPE_DRIVE		
MD 35240: $MA_ACCEL_TYPE_DRIVE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

ACT_POS_ABS (A, M)		MD 30250: $MA_ACT_POS_ABS		
MD 30250: $MA_ACT_POS_ABS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

ACT_POS_ABS (A, M)	MD 30250: $MA_ACT_POS_ABS	
	2	
	Read access	

AC_FILTER_TIME (A, M)		MD 32920: $MA_AC_FILTER_TIME		
MD 32920: $MA_AC_FILTER_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

LOAD_SMOOTH_FILTER_TIME (A, M)		MD 32925: $MA_LOAD_SMOOTH_FIL-TER_TIME		
MD 32925: $MA_LOAD_SMOOTH_FILTER_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access and write access	

POWER_SMOOTH_FILTER_TIME (A, M)		MD 32926: $MA_POWER_SMOOTH_FIL-TER_TIME		
MD 32926: $MA_POWER_SMOOTH_FILTER_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access and write access	

AUTO_GET_TYPE (A, M)		MD 30552: $MA_AUTO_GET_TYPE		
MD 30552: $MA_AUTO_GET_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

AXCONF_ASSIGN_MASTER_CHAN (A, M)		MD 30550: $MA_AXCONF_AS-SIGN_MASTER_CHAN		
MD 30550: $MA_AXCONF_ASSIGN_MASTER_CHAN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

AXIS_DIAGNOSIS (A, M)	MD 36690: $MA_AXIS_DIAGNOSIS	
MD 36690: $MA_AXIS_DIAGNOSIS		

AXIS_DIAGNOSIS (A, M)		MD 36690: $MA_AXIS_DIAGNOSIS		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

AX_EMERGENCY_STOP_TIME (A, M)		MD 36610: $MA_AX_EMERGEN-CY_STOP_TIME		
MD 36610: $MA_AX_EMERGENCY_STOP_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

AX_INERTIA (A, M)		MD 32650: $MA_AX_INERTIA		
MD 32650: $MA_AX_INERTIA				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

AX_JERK_ENABLE (A, M)		MD 32400: $MA_AX_JERK_ENABLE		
MD 32400: $MA_AX_JERK_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

AX_JERK_TIME (A, M)		MD 32410: $MA_AX_JERK_TIME		
MD 32410: $MA_AX_JERK_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

AX_MASS (A, M)		MD 32652: $MA_AX_MASS		
MD 32652: $MA_AX_MASS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

AX_MOTION_DIR (A, M)		MD 32100: $MA_AX_MOTION_DIR		
MD 32100: $MA_AX_MOTION_DIR				

AX_MOTION_DIR (A, M)		MD 32100: $MA_AX_MOTION_DIR		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

AX_VELO_LIMIT (A, M)		MD 36200: $MA_AX_VELO_LIMIT		
MD 36200: $MA_AX_VELO_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

BACKLASH (A, M)		MD 32450: $MA_BACKLASH		
MD 32450: $MA_BACKLASH				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

BERO_CYCLE (A, M)		MD 31100: $MA_BERO_CYCLE		
MD 31100: $MA_BERO_CYCLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

BERO_DELAY_TIME_MINUS (A, M)		MD 31123: $MA_BERO_DE-LAY_TIME_MINUS		
MD 31123: $MA_BERO_DELAY_TIME_MINUS				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

BERO_DELAY_TIME_MINUS (A, M)	MD 31123: $MA_BERO_DE-LAY_TIME_MINUS	
	2	
	Read access	

BERO_DELAY_TIME_PLUS (A, M)		MD 31122: $MA_BERO_DE-LAY_TIME_PLUS		
MD 31122: $MA_BERO_DELAY_TIME_PLUS				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

BERO_EDGE (A, M)		MD 31120: $MA_BERO_EDGE		
MD 31120: $MA_BERO_EDGE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

BERO_EDGE_TOL (A, M)		MD 31110: $MA_BERO_EDGE_TOL		
MD 31110: $MA_BERO_EDGE_TOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

BRAKE_MODE_CHOICE (A, M)		MD 36600: $MA_BRAKE_MODE_CHOICE		
MD 36600: $MA_BRAKE_MODE_CHOICE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

CEC_ENABLE (A, M)		MD 32710: $MA_CEC_ENABLE		
MD 32710: $MA_CEC_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

CEC_MAX_SUM (A, M)		MD 32720: $MA_CEC_MAX_SUM		
MD 32720: $MA_CEC_MAX_SUM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CEC_MAX_VELO (A, M)		MD 32730: $MA_CEC_MAX_VELO		
MD 32730: $MA_CEC_MAX_VELO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CLAMP_POS_TOL (A, M)		MD 36050: $MA_CLAMP_POS_TOL		
MD 36050: $MA_CLAMP_POS_TOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

COMPRESS_POS_TOL (A, M)		MD 33100: $MA_COMPRESS_POS_TOL		
MD 33100: $MA_COMPRESS_POS_TOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

COMP_ADD_VELO_FACTOR (A, M)		MD 32760: $MA_COMP_ADD_VE-LO_FACTOR		
MD 32760: $MA_COMP_ADD_VELO_FACTOR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CONTOUR_TOL (A, M)		MD 36400: $MA_CONTOUR_TOL		
MD 36400: $MA_CONTOUR_TOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CORR_VELO (A, M)		MD 32070: $MA_CORR_VELO		
MD 32070: $MA_CORR_VELO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

COUPLE_POS_TOL_COARSE (A, M)		MD 37200: $MA_COUPLE_POS_TOL_COARSE		
MD 37200: $MA_COUPLE_POS_TOL_COARSE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

COUPLE_POS_TOL_FINE (A, M)		MD 37210: $MA_COUPLE_POS_TOL_FINE		
MD 37210: $MA_COUPLE_POS_TOL_FINE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

COUPLE_VELO_TOL_COARSE (A, M)		MD 37220: $MA_COUPLE_VELO_TOL_COARSE		
MD 37220: $MA_COUPLE_VELO_TOL_COARSE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

COUPLE_VELO_TOL_FINE (A, M)		MD 37230: $MA_COUPLE_VELO_TOL_FINE		
MD 37230: $MA_COUPLE_VELO_TOL_FINE				
Units and value ranges				
Physical unit:	Data type:			

COUPLE_VELO_TOL_FINE (A, M)		MD 37230: $MA_COUPLE_VELO_TOL_FINE		
-	TYPE_DOUBLE			
			Read access	

CTRLOUT_LIMIT (A, M)		MD 36210: $MA_CTRLOUT_LIMIT		
MD 36210: $MA_CTRLOUT_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CTRLOUT_LIMIT_TIME (A, M)		MD 36220: $MA_CTRLOUT_LIMIT_TIME		
MD 36220: $MA_CTRLOUT_LIMIT_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

CTRLOUT_NR (A, M)		MD 30120: $MA_CTRLOUT_NR		
MD 30120: $MA_CTRLOUT_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

CTRLOUT_SEGMENT_NR (A, M)		MD 30100: $MA_CTRLOUT_SEGMENT_NR		
MD 30100: $MA_CTRLOUT_SEGMENT_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

DES_VELO_LIMIT (A, M)		MD 36520: $MA_DES_VELO_LIMIT		
MD 36520: $MA_DES_VELO_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_P_MAX (A, M)		MD 37310: $MA_DIG_P_MAX		
MD 37310: $MA_DIG_P_MAX				
Units and value ranges				
Physical unit:	Data type:			

DIG_P_MAX (A, M)		MD 37310: $MA_DIG_P_MAX		
-	TYPE_DOUBLE			
			Read access	

DIG_P_MIN (A, M)		MD 37300: $MA_DIG_P_MIN		
MD 37300: $MA_DIG_P_MIN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_V_MAX (A, M)		MD 37320: $MA_DIG_V_MAX		
MD 37320: $MA_DIG_V_MAX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DISPLAY_IS_MODULO (A, M)		MD 30320: $MA_DISPLAY_IS_MODULO		
MD 30320: $MA_DISPLAY_IS_MODULO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

DRIFT_ENABLE (A, M)		MD 36700: $MA_DRIFT_ENABLE		
MD 36700: $MA_DRIFT_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

DRIFT_LIMIT (A, M)		MD 36710: $MA_DRIFT_LIMIT		
MD 36710: $MA_DRIFT_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DRIFT_VALUE (A, M)		MD 36720: $MA_DRIFT_VALUE		
MD 36720: $MA_DRIFT_VALUE				
Units and value ranges				
Physical unit:	Data type:			

DRIFT_VALUE (A, M)		MD 36720: $MA_DRIFT_VALUE		
-	TYPE_DOUBLE			
			Read access	

DRIVE_AX_RATIO_DENOM (A, M)		MD 31050: $MA_DRIVE_AX_RATIO_DENOM		
MD 31050: $MA_DRIVE_AX_RATIO_DENOM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

DRIVE_AX_RATIO_NUMERA (A, M)		MD 31060: $MA_DRIVE_AX_RATIO_NUMERA		
MD 31060: $MA_DRIVE_AX_RATIO_NUMERA				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

DRIVE_ENC_RATIO_DENOM (A, M)		MD 31070: $MA_DRIVE_ENC_RATIO_DENOM		
MD 31070: $MA_DRIVE_ENC_RATIO_DENOM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

DRIVE_ENC_RATIO_NUMERA (A, M)		MD 31080: $MA_DRIVE_ENC_RATIO_NUMERA		
MD 31080: $MA_DRIVE_ENC_RATIO_NUMERA				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	

DRIVE_ENC_RATIO_NUMERA (A, M)	MD 31080: $MA_DRIVE_ENC_RATIO_NUMERA	
	2	
	Read access	

DRIVE_SIGNAL_TRACKING (A, M)		MD 36730: $MA_DRIVE_SIGNAL_TRACKING		
MD 36730: $MA_DRIVE_SIGNAL_TRACKING				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

DYN_MATCH_ENABLE (A, M)		MD 32900: $MA_DYN_MATCH_ENABLE		
MD 32900: $MA_DYN_MATCH_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

DYN_MATCH_TIME (A, M)		MD 32910: $MA_DYN_MATCH_TIME		
MD 32910: $MA_DYN_MATCH_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_ABS_TURNS_MODULO (A, M)		MD 34220: $MA_ENC_ABS_TURNS_MODULO		
MD 34220: $MA_ENC_ABS_TURNS_MODULO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_CHANGE_TOL (A, M)		MD 36500: $MA_ENC_CHANGE_TOL		
MD 36500: $MA_ENC_CHANGE_TOL				
Units and value ranges				
Physical unit:	Data type:			

ENC_CHANGE_TOL (A, M)		MD 36500: $MA_ENC_CHANGE_TOL		
-	TYPE_DOUBLE			
			Read access	

ENC_COMP_ENABLE (A, M)		MD 32700: $MA_ENC_COMP_ENABLE		
MD 32700: $MA_ENC_COMP_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_DIFF_TOL (A, M)		MD 36510: $MA_ENC_DIFF_TOL		
MD 36510: $MA_ENC_DIFF_TOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

ENC_FEEDBACK_POL (A, M)		MD 32110: $MA_ENC_FEEDBACK_POL		
MD 32110: $MA_ENC_FEEDBACK_POL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_FREQ_LIMIT (A, M)		MD 36300: $MA_ENC_FREQ_LIMIT		
MD 36300: $MA_ENC_FREQ_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_FREQ_LIMIT_LOW (A, M)	MD 36302: $MA_ENC_FREQ_LIMIT_LOW	
MD 36302: $MA_ENC_FREQ_LIMIT_LOW		

ENC_FREQ_LIMIT_LOW (A, M)		MD 36302: $MA_ENC_FREQ_LIMIT_LOW		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_GRID_POINT_DIST (A, M)		MD 31010: $MA_ENC_GRID_POINT_DIST		
MD 31010: $MA_ENC_GRID_POINT_DIST				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_INPUT_NR (A, M)		MD 30230: $MA_ENC_INPUT_NR		
MD 30230: $MA_ENC_INPUT_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_INVERS (A, M)		MD 34320: $MA_ENC_INVERS		
MD 34320: $MA_ENC_INVERS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_IS_DIRECT (A, M)		MD 31040: $MA_ENC_IS_DIRECT		
MD 31040: $MA_ENC_IS_DIRECT				
Units and value ranges				
Physical unit:	Data type:			

ENC_IS_DIRECT (A, M)		MD 31040: $MA_ENC_IS_DIRECT		
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_IS_INDEPENDENT (A, M)		MD 30242: $MA_ENC_IS_INDEPENDENT		
MD 30242: $MA_ENC_IS_INDEPENDENT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_IS_LINEAR (A, M)		MD 31000: $MA_ENC_IS_LINEAR		
MD 31000: $MA_ENC_IS_LINEAR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_MARKER_INC (A, M)		MD 34310: $MA_ENC_MARKER_INC		
MD 34310: $MA_ENC_MARKER_INC				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_REFP_MARKER_DIST (A, M)		MD 34300: $MA_ENC_REFP_MARK-ER_DIST		
MD 34300: $MA_ENC_REFP_MARKER_DIST				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

ENC_REFP_MARKER_DIST (A, M)	MD 34300: $MA_ENC_REFP_MARKER_DIST	
Addressing		
Line index:	Max. line index:	
	2	
	Read access	

ENC_REFP_MODE (A, M)		MD 34200: $MA_ENC_REFP_MODE		
MD 34200: $MA_ENC_REFP_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_REFP_STATE (A, M)		MD 34210: $MA_ENC_REFP_STATE		
MD 34210: $MA_ENC_REFP_STATE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_RESOL (A, M)		MD 31020: $MA_ENC_RESOL		
MD 31020: $MA_ENC_RESOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_SEGMENT_NR (A, M)		MD 30210: $MA_ENC_SEGMENT_NR		
MD 30210: $MA_ENC_SEGMENT_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	

ENC_SEGMENT_NR (A, M)	MD 30210: $MA_ENC_SEGMENT_NR	
	2	
	Read access	

ENC_SSI_MESSAGE_FORMAT (A, M)		MD 34420: $MA_ENC_SSI_MES-SAGE_FORMAT		
MD 34420: $MA_ENC_SSI_MESSAGE_FORMAT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_SSI_MESSAGE_LENGTH (A, M)		MD 34410: $MA_ENC_SSI_MES-SAGE_LENGTH		
MD 34410: $MA_ENC_SSI_MESSAGE_LENGTH				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_SSI_STATUS (A, M)		MD 34400: $MA_ENC_SSI_STATUS		
MD 34400: $MA_ENC_SSI_STATUS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ENC_ZERO_MONITORING (A, M)		MD 36310: $MA_ENC_ZERO_MONITOR-ING		
MD 36310: $MA_ENC_ZERO_MONITORING				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	

ENC_ZERO_MONITORING (A, M)	MD 36310: $MA_ENC_ZERO_MONITOR-ING	
	2	
	Read access	

EPS_TLIFT_TANG_STEP (A, M)		MD 37400: $MA_EPS_TLIFT_TANG_STEP		
MD 37400: $MA_EPS_TLIFT_TANG_STEP				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

EQUIV_CURRCTRL_TIME (A, M)		MD 32800: $MA_EQUIV_CURRCTRL_TIME		
MD 32800: $MA_EQUIV_CURRCTRL_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

EQUIV_SPEEDCTRL_TIME (A, M)		MD 32810: $MA_EQUIV_SPEEDCTRL_TIME		
MD 32810: $MA_EQUIV_SPEEDCTRL_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

FFW_ACTIVATION_MODE (A, M)		MD 32630: $MA_FFW_ACTIVA-TION_MODE		
MD 32630: $MA_FFW_ACTIVATION_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FFW_MODE (A, M)	MD 32620: $MA_FFW_MODE	
MD 32620: $MA_FFW_MODE		

FFW_MODE (A, M)		MD 32620: $MA_FFW_MODE		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FIPO_TYPE (A, M)		MD 33000: $MA_FIPO_TYPE		
MD 33000: $MA_FIPO_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FIXED_STOP_ACKN_MASK (A, M)		MD 37060: $MA_FIXED_STOP_ACKN_MASK		
MD 37060: $MA_FIXED_STOP_ACKN_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FIXED_STOP_ALARM_MASK (A, M)		MD 37050: $MA_FIXED_STOP_ALARM_MASK		
MD 37050: $MA_FIXED_STOP_ALARM_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FIXED_STOP_ANA_TORQUE (A, M)		MD 37070: $MA_FIXED_STOP_ANA_TORQUE		
MD 37070: $MA_FIXED_STOP_ANA_TORQUE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FIXED_STOP_BY_SENSOR (A, M)		MD 37040: $MA_FIXED_STOP_BY_SEN-SOR		
MD 37040: $MA_FIXED_STOP_BY_SENSOR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FIXED_STOP_MODE (A, M)		MD 37000: $MA_FIXED_STOP_MODE		
MD 37000: $MA_FIXED_STOP_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FIXED_STOP_THRESHOLD (A, M)		MD 37030: $MA_FIXED_STOP_THRESH-OLD		
MD 37030: $MA_FIXED_STOP_THRESHOLD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FIXED_STOP_TORQUE_DEF (A, M)		MD 37010: $MA_FIXED_STOP_TOR-QUE_DEF		
MD 37010: $MA_FIXED_STOP_TORQUE_DEF				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FIXED_STOP_WINDOW_DEF (A, M)		MD 37020: $MA_FIXED_STOP_WIN-DOW_DEF		
MD 37020: $MA_FIXED_STOP_WINDOW_DEF				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FIX_POINT_POS (A, M)		MD 30600: $MA_FIX_POINT_POS		
MD 30600: $MA_FIX_POINT_POS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

FRAME_OR_CORRPOS_NOTALLOWED (A, M)		MD 32074: $MA_FRAME_OR_CORR-POS_NOTALLOWED		
MD 32074: $MA_FRAME_OR_CORRPOS_NOTALLOWED				

FRAME_OR_CORRPOS_NOTALLOWED (A, M)		MD 32074: $MA_FRAME_OR_CORR-POS_NOTALLOWED		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

FREQUENCY1 (A, M)		MD 31310: $MA_FREQUENCY1		
MD 31310: $MA_FREQUENCY1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FREQUENCY2 (A, M)		MD 31320: $MA_FREQUENCY2		
MD 31320: $MA_FREQUENCY2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FREQ_REDUCTION_POINT1 (A, M)		MD 31330: $MA_FREQ_REDUC-TION_POINT1		
MD 31330: $MA_FREQ_REDUCTION_POINT1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FREQ_REDUCTION_POINT2 (A, M)		MD 31340: $MA_FREQ_REDUC-TION_POINT2		
MD 31340: $MA_FREQ_REDUCTION_POINT2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FREQ_STEP_LIMIT (A, M)		MD 31350: $MA_FREQ_STEP_LIMIT		
MD 31350: $MA_FREQ_STEP_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FRICT_COMP_ACCEL1 (A, M)		MD 32550: $MA_FRICT_COMP_ACCEL1		
MD 32550: $MA_FRICT_COMP_ACCEL1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FRICT_COMP_ACCEL2 (A, M)		MD 32560: $MA_FRICT_COMP_ACCEL2		
MD 32560: $MA_FRICT_COMP_ACCEL2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FRICT_COMP_ACCEL3 (A, M)		MD 32570: $MA_FRICT_COMP_ACCEL3		
MD 32570: $MA_FRICT_COMP_ACCEL3				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FRICT_COMP_ADAPT_ENABLE (A, M)		MD 32510: $MA_FRICT_COMP_ADAPT_ENABLE		
MD 32510: $MA_FRICT_COMP_ADAPT_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

FRICT_COMP_CONST_MAX (A, M)		MD 32520: $MA_FRICT_COMP_CONST_MAX		
MD 32520: $MA_FRICT_COMP_CONST_MAX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FRICT_COMP_CONST_MIN (A, M)		MD 32530: $MA_FRICT_COMP_CONST_MIN		
MD 32530: $MA_FRICT_COMP_CONST_MIN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FRICT_COMP_ENABLE (A, M)		MD 32500: $MA_FRICT_COMP_ENABLE		
MD 32500: $MA_FRICT_COMP_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

FRICT_COMP_INC_FACTOR (A, M)		MD 32580: $MA_FRICT_COMP_INC_FACTOR		
MD 32580: $MA_FRICT_COMP_INC_FACTOR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FRICT_COMP_MODE (A, M)		MD 32490: $MA_FRICT_COMP_MODE		
MD 32490: $MA_FRICT_COMP_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FRICT_COMP_TIME (A, M)		MD 32540: $MA_FRICT_COMP_TIME		
MD 32540: $MA_FRICT_COMP_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

GANTRY_AXIS_TYPE (A, M)		MD 37100: $MA_GANTRY_AXIS_TYPE		
MD 37100: $MA_GANTRY_AXIS_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

GANTRY_BREAK_UP (A, M)		MD 37140: $MA_GANTRY_BREAK_UP		
MD 37140: $MA_GANTRY_BREAK_UP				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

GANTRY_POS_TOL_ERROR (A, M)		MD 37120: $MA_GANTRY_POS_TOL_ERROR		
MD 37120: $MA_GANTRY_POS_TOL_ERROR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

GANTRY_POS_TOL_REF (A, M)		MD 37130: $MA_GANTRY_POS_TOL_REF		
MD 37130: $MA_GANTRY_POS_TOL_REF				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

GANTRY_POS_TOL_WARNING (A, M)		MD 37110: $MA_GANTRY_POS_TOL_WARNING		
MD 37110: $MA_GANTRY_POS_TOL_WARNING				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

GEAR_STEP_CHANGE_ENABLE (A, M)	MD 35010: $MA_GEAR_STEP_CHANGE_ENABLE	

Parameterize gear stage change

Meaning of bit places:

o Bit 0 = 0 and bit 1 = 0:

There is an invariable transformation ratio between motor and load. The MD of the first gear stage is active. Gear stage change is not possible with M40 to M45.

o Bit 0 = 1:

Gear stage change at undefined change position. The gearbox can have up to 5 gear stages, which can be selected via M40, M41 to M45. To support the gear stage change, the motor can carry out oscillating motions, which must be enabled by the PLC program.

o Bit 1 = 1:

Same meaning as bit 0 = 1, but the gear stage change is carried out at a configured spindle position. The change position is configured in MD35012 $MA_GEAR_STEP_CHANGE_POSITION. The position is approached in the current gear stage before the gear stage change. If this bit is set, bit 0 is not taken into account!

o Bit 2: Reserved

o Bit 3 = 1:

The gear stage change dialog between NCK and PLC is simulated. The setpoint gear stage is output to the PLC. The system does not wait for a feedback signal from the PLC to the NCK. The acknowledgment is generated internally in the NCK.

o Bit 4: Reserved

o Bit 5 = 1:

The second gear stage data set is used when tapping with G331/G332. The bit must be set for the master spindle used when tapping. Bit 0 or bit 1 must be set.

Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

GEAR_STEP_MAX_VELO (A, M)	MD 35110: $MA_GEAR_STEP_MAX_VELO	

MD 35110: $MA_GEAR_STEP_MAX_VELO

Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

Addressing	
Line index:	Max. line index:
	2
	Read access

GEAR_STEP_MAX_VELO_LIMIT (A, M)	MD 35130: $MA_GEAR_STEP_MAX_VELO_LIMIT	

MD 35130: $MA_GEAR_STEP_MAX_VELO_LIMIT

Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

GEAR_STEP_MAX_VELO_LIMIT (A, M)	MD 35130: $MA_GEAR_STEP_MAX_VELO_LIMIT	
Addressing		
Line index:	Max. line index:	
	2	
	Read access	

GEAR_STEP_MIN_VELO (A, M)		MD 35120: $MA_GEAR_STEP_MIN_VELO		
MD 35120: $MA_GEAR_STEP_MIN_VELO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

GEAR_STEP_MIN_VELO_LIMIT (A, M)		MD 35140: $MA_GEAR_STEP_MIN_VELO_LIMIT		
MD 35140: $MA_GEAR_STEP_MIN_VELO_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

GEAR_STEP_POSCTRL_ACCEL (A, M)		MD 35210: $MA_GEAR_STEP_POSCTRL_ACCEL		
MD 35210: $MA_GEAR_STEP_POSCTRL_ACCEL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

GEAR_STEP_SPEEDCTRL_ACCEL (A, M)		MD 35200: $MA_GEAR_STEP_SPEEDCTRL_ACCEL		
MD 35200: $MA_GEAR_STEP_SPEEDCTRL_ACCEL				
Units and value ranges				
Physical unit:	Data type:			

GEAR_STEP_SPEEDCTRL_ACCEL (A, M)		MD 35200: $MA_GEAR_STEP_SPEEDCTRL_ACCEL		
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

HANDWH_MAX_INCR_SIZE (A, M)		MD 32080: $MA_HANDWH_MAX_INCR_SIZE		
MD 32080: $MA_HANDWH_MAX_INCR_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

HANDWH_MAX_INCR_VELO_SIZE (A, M)		MD 32082: $MA_HANDWH_MAX_INCR_VELO_SIZE		
MD 32082: $MA_HANDWH_MAX_INCR_VELO_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

HANDWH_STOP_COND (A, M)		MD 32084: $MA_HANDWH_STOP_COND		
MD 32084: $MA_HANDWH_STOP_COND				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

HANDWH_VELO_OVERLAY_FACTOR (A, M)		MD 32090: $MA_HANDWH_VE-LO_OVERLAY_FACTOR		
MD 32090: $MA_HANDWH_VELO_OVERLAY_FACTOR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

HIRTH_IS_ACTIVE (A, M)		MD 30505: $MA_HIRTH_IS_ACTIVE		
MD 30505: $MA_HIRTH_IS_ACTIVE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

INDEX_AX_ASSIGN_POS_TAB (A, M)		MD 30500: $MA_INDEX_AX_ASSIGN_POS_TAB		
Axis is an indexing axis The variable defines the axis as an indexing axis. Machine axes which only have to approach certain positions can be defined as indexing axes. The positions are shown in position tables 1 and 2: Multiple axes can also be assigned to an indexing position table. The requirement for this is that these indexing axes are of the same type (linear axis, rotary axis).				
Return value	o =0: The axis is not declared as an indexing axis. o =1: The axis is an indexing axis. The associated indexing positions are shown in table 1 (MD10910 $MN_INDEX_AX_POS_TAB_1). o =2: The axis is an indexing axis. The associated indexing positions are shown in table 2 (MD10930 $MN_INDEX_AX_POS_TAB_2). o =3: Equidistant indexing o >3: Alarm 17090 "Value greater than upper limit"			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

INDEX_AX_DENOMINATOR (A, M)		MD 30502: $MA_INDEX_AX_DENOMINATOR		
MD 30502: $MA_INDEX_AX_DENOMINATOR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

INDEX_AX_NUMERATOR (A, M)		MD 30501: $MA_INDEX_AX_NUMERATOR		
MD 30501: $MA_INDEX_AX_NUMERATOR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

INDEX_AX_OFFSET (A, M)		MD 30503: $MA_INDEX_AX_OFFSET		
MD 30503: $MA_INDEX_AX_OFFSET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

IS_CC_AX (A, M)		MD 30400: $MA_IS_CC_AX		
MD 30400: $MA_IS_CC_AX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

IS_CONCURRENT_POS_AX (A, M)		MD 30450: $MA_IS_CONCUR-RENT_POS_AX		
MD 30450: $MA_IS_CONCURRENT_POS_AX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

IS_ROT_AX (A, M)		MD 30300: $MA_IS_ROT_AX		
MD 30300: $MA_IS_ROT_AX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

IS_VIRTUAL_AX (A, M)		MD 30132: $MA_IS_VIRTUAL_AX		
Axis is a virtual axis The variable defines an axis as a virtual axis. A virtual axis is an axis which is interpolated in follow-up mode. The virtual axis is used with master value coupling. Examples are the electronic transfer CP, master value coupling with curve tables or flying scissors.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

JOG_AND_POS_JERK_ENABLE (A, M)		MD 32420: $MA_JOG_AND_POS_JERK_ENABLE		
MD 32420: $MA_JOG_AND_POS_JERK_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

JOG_AND_POS_MAX_JERK (A, M)		MD 32430: $MA_JOG_AND_POS_MAX_JERK		
MD 32430: $MA_JOG_AND_POS_MAX_JERK				

JOG_AND_POS_MAX_JERK (A, M)		MD 32430: $MA_JOG_AND_POS_MAX_JERK		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

JOG_INCR_WEIGHT (A, M)		MD 31090: $MA_JOG_INCR_WEIGHT		
MD 31090: $MA_JOG_INCR_WEIGHT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

JOG_REV_VELO (A, M)		MD 32050: $MA_JOG_REV_VELO		
MD 32050: $MA_JOG_REV_VELO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

JOG_REV_VELO_RAPID (A, M)		MD 32040: $MA_JOG_REV_VELO_RAP-ID		
MD 32040: $MA_JOG_REV_VELO_RAPID				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

JOG_VELO (A, M)		MD 32020: $MA_JOG_VELO		
Jog axis velocity The specified velocity applies to traversing in JOG mode with an axial feedrate override switch setting of 100%. The velocity is only used if the general SD41110 $SN_JOG_SET_VELO = 0 and the linear feedrate is selected (SD41100 $SN_JOG_REV_IS_ACTIVE = 0) for linear axes, or SD41130 $SN_JOG_ROT_AX_SET_VELO = 0 is selected for rotary axes.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

JOG_VELO_RAPID (A, M)		MD 32010: $MA_JOG_VELO_RAPID		
Rapid traverse in jog mode The specified axis velocity applies when the rapid traverse override key is actuated in JOG mode and when the axial feedrate override is set to 100%. The value entered must not exceed the maximum permissible axis velocity (MD32000 $MA_MAX_AX_VELO). This machine data is not used for the programmed rapid traverse G0.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

LEADSCREW_PITCH (A, M)		MD 31030: $MA_LEADSCREW_PITCH		
MD 31030: $MA_LEADSCREW_PITCH				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

LUBRICATION_DIST (A, M)		MD 33050: $MA_LUBRICATION_DIST		
MD 33050: $MA_LUBRICATION_DIST				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

MAX_ACCEL_OVL_FACTOR (A, M)		MD 32310: $MA_MAX_ACCEL_OVL_FACTOR		
MD 32310: $MA_MAX_ACCEL_OVL_FACTOR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

MAX_AX_ACCEL (A, M)		MD 32300: $MA_MAX_AX_ACCEL		
MD 32300: $MA_MAX_AX_ACCEL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

MAX_AX_VELO (A, M)		MD 32000: $MA_MAX_AX_VELO		
MD 32000: $MA_MAX_AX_VELO				
Units and value ranges				
Physical unit:	Data type:			

MAX_AX_VELO (A, M)		MD 32000: $MA_MAX_AX_VELO		
-	TYPE_DOUBLE			
			Read access	

MDCA_CTRLOUT_MODULE_NR (A, M)		MD 30110: CTRLOUT_MODULE_NR		
Alternative name:	CTRLOUT_MODULE_NR			
Setpoint assignment Setpoint assignment: Drive number / module number				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_CHAR		1	15
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDCA_CTRLOUT_TYPE (A, M)		MD 30130: CTRLOUT_TYPE		
Alternative name:	CTRLOUT_TYPE			
Type of setpoint output				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_CHAR		0	1
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDCA_ENC_MODULE_NR (A, M)		MD 30220: ENC_MODULE_NR[x] x = PlugplaceNo		
Alternative name:	ENC_MODULE_NR			
Actual value assignment Actual value assignment: Drive number / measuring circuit number				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_CHAR		1	15
Addressing				
Line index:			Max. line index:	
			2	
Validity:	From NCK version 3.6		Read access and write access	

MDCA_ENC_TYPE (A, M)		MD 30240: ENC_TYPE[x] x = Plugplace-No		
Alternative name:	ENC_TYPE			
Type of actual value sensing Type of actual value sensing (actual position value)				
Return value	Encoder type: 0: Simulation 1: Raw signal generator (high resolution) 2: Square wave generator - only with available onboard hardware 3: Encoder for semi-servo - only with available onboard hardware 4: Absolute encoder, gen. (e.g. with EnDat interface) 5: reserved			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_CHAR		0	5
Addressing				
Line index:			Max. line index:	
			2	
Validity:	From NCK version 3.6		Read access and write access	

MM_ENC_COMP_MAX_POINTS (A, M)		MD 38000: $MA_MM_ENC_COMP_MAX_POINTS		
MD 38000: $MA_MM_ENC_COMP_MAX_POINTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

MM_QEC_MAX_POINTS (A, M)		MD 38010: $MA_MM_QEC_MAX_POINTS		
MD 38010: $MA_MM_QEC_MAX_POINTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MODULO_RANGE (A, M)		MD 30330: $MA_MODULO_RANGE		
MD 30330: $MA_MODULO_RANGE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

NUM_ENCS (A, M)		MD 30200: $MA_NUM_ENCS		
MD 30200: $MA_NUM_ENCS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

PARAMSET_CHANGE_ENABLE (A, M)		MD 35590: $MA_PARA-MSET_CHANGE_ENABLE		
MD 35590: $MA_PARAMSET_CHANGE_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

PATH_TRANS_JERK_LIM (A, M)		MD 32432: $MA_PATH_TRANS_JERK_LIM		
MD 32432: $MA_PATH_TRANS_JERK_LIM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

POSCTRL_DAMPING (A, M)		MD 32950: $MA_POSCTRL_DAMPING		
MD 32950: $MA_POSCTRL_DAMPING				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

POSCTRL_GAIN (A, M)		MD 32200: $MA_POSCTRL_GAIN		
MD 32200: $MA_POSCTRL_GAIN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

POSITIONING_TIME (A, M)		MD 36020: $MA_POSITIONING_TIME		
MD 36020: $MA_POSITIONING_TIME				
Units and value ranges				
Physical unit:	Data type:			

POSITIONING_TIME (A, M)		MD 36020: $MA_POSITIONING_TIME		
s, user defined	TYPE_DOUBLE			
			Read access	

POS_AX_VELO (A, M)		MD 32060: $MA_POS_AX_VELO		
MD 32060: $MA_POS_AX_VELO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

POS_LIMIT_MINUS2 (A, M)		MD 36120: $MA_POS_LIMIT_MINUS2		
Software limit switch 2 minus The variable can define a 2nd SW limit switch in the negative direction in the machine axis system (machine position in the machine coordinate system). An interface signal from the PLC can be used to select whichever of the two SW limit switches 1 or 2 is to be active. E.g.: DB31, ... DBX12.2 = 0 (1st software limit switch minus) for axis 1 active DB31, ... DBX12.2 = 1 (2nd software limit switch minus) for axis 1 active. The variable is not relevant if the axis has not been referenced.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

POS_LIMIT_MINUS (A, M)		MD 36100: $MA_POS_LIMIT_MINUS		
MD 36100: $MA_POS_LIMIT_MINUS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

POS_LIMIT_PLUS2 (A, M)		MD 36130: $MA_POS_LIMIT_PLUS2		
Software limit switch 2 plus The variable can define a 2nd SW limit switch position in the positive direction in the machine axis system (machine position in the machine coordinate system). An interface signal from the PLC can be used to select whichever of the two SW limit switches 1 or 2 is to be active. E.g.: DB31, ... DBX12.3 = 0 (1st software limit switch plus) for axis 1 active DB31, ... DBX12.3 = 1 (2nd software limit switch plus) for axis 1 active. The variable is not relevant if the axis has not been referenced.				
Units and value ranges				
Physical unit:	Data type:			

POS_LIMIT_PLUS2 (A, M)		MD 36130: $MA_POS_LIMIT_PLUS2		
-	TYPE_DOUBLE			
			Read access	

POS_LIMIT_PLUS (A, M)		MD 36110: $MA_POS_LIMIT_PLUS		
Software limit switch 1 plus The variable specifies the 1st software limit switch position. The absolute position of the positive range limit of each axis can be entered in the machine axis system (machine position in the machine coordinate system). The variable is active after reference point approach if NC/PLC interface signal DB31, ... DBX12.3 (2nd software limit switch plus) has not been set. The variable is not relevant if the axis has not been referenced.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

RATED_OUTVAL (A, M)		MD 32250: $MA_RATED_OUTVAL		
MD 32250: $MA_RATED_OUTVAL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

RATED_VELO (A, M)		MD 32260: $MA_RATED_VELO		
MD 32260: $MA_RATED_VELO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

REFP_CAM_DIR_IS_MINUS (A, M)	MD 34010: $MA_REFP_CAM_DIR_IS_MINUS			
Direction of the reference point approach The variable specifies the direction of the reference point approach. For incremental measuring systems: If the machine axis is positioned in front of the reference cam, it accelerates according to the plus/minus traversing key pressed to the velocity specified in MD34020 $MA_REFP_VELO_SEARCH_CAM (reference point approach velocity) in the direction specified in MD34010 $MA_REFP_CAM_DIR_IS_MINUS. If the wrong traversing key is pressed, reference point approach is not started. If the machine axis is positioned on the reference cam, it accelerates to the velocity specified in MD34020 $MA_REFP_VELO_SEARCH_CAM and travels in the opposite direction to that specified in MD34010 $MA_REFP_CAM_DIR_IS_MINUS. For linear measuring systems with distance-coded reference marks: If the machine axis has a reference cam (linear measuring systems with distance-coded reference marks do not necessarily require a reference cam) and if the machine axis is positioned on the reference cam, it accelerates, irrespective of the plus/minus traversing key pressed, to the velocity specified in MD34040 $MA_REFP_VELO_SEARCH_MARKER (reference point creep velocity) in the opposite direction to that specified in MD34010 $MA_REFP_CAM_DIR_IS_MINUS.				
Return value	o 0: MD34010 $MA_REFP_CAM_DIR_IS_MINUS reference point approach in the plus direction o 1: MD34010 $MA_REFP_CAM_DIR_IS_MINUS reference point approach in the minus direction			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

REFP_CAM_IS_ACTIVE (A, M)	MD 34000: $MA_REFP_CAM_IS_ACTIVE			
MD 34000: $MA_REFP_CAM_IS_ACTIVE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

REFP_CAM_SHIFT (A, M)	MD 34092: $MA_REFP_CAM_SHIFT			
MD 34092: $MA_REFP_CAM_SHIFT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

REFP_CYCLE_NR (A, M)		MD 34110: $MA_REFP_CYCLE_NR		
MD 34110: $MA_REFP_CYCLE_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

REFP_MAX_CAM_DIST (A, M)		MD 34030: $MA_REFP_MAX_CAM_DIST		
MD 34030: $MA_REFP_MAX_CAM_DIST				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

REFP_MAX_MARKER_DIST (A, M)		MD 34060: $MA_REFP_MAX_MARK-ER_DIST		
MD 34060: $MA_REFP_MAX_MARKER_DIST				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

REFP_MOVE_DIST (A, M)		MD 34080: $MA_REFP_MOVE_DIST		
MD 34080: $MA_REFP_MOVE_DIST				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

REFP_MOVE_DIST_CORR (A, M)		MD 34090: $MA_REFP_MOVE_DIST_CORR		
MD 34090: $MA_REFP_MOVE_DIST_CORR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

REFP_MOVE_DIST_CORR (A, M)	MD 34090: $MA_REFP_MOVE_DIST_CORR	
	2	
	Read access	

REFP_SEARCH_MARKER_REVERSE (A, M)		MD 34050: $MA_REFP_SEARCH_MARKER_REVERSE		
MD 34050: $MA_REFP_SEARCH_MARKER_REVERSE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

REFP_SET_POS (A, M)		MD 34100: $MA_REFP_SET_POS		
MD 34100: $MA_REFP_SET_POS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

REFP_STOP_AT_ABS_MARKER (A, M)		MD 34330: $MA_REFP_STOP_AT_ABS_MARKER		
MD 34330: $MA_REFP_STOP_AT_ABS_MARKER				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

REFP_SYNC_ENCS (A, M)		MD 34102: $MA_REFP_SYNC_ENCS		
MD 34102: $MA_REFP_SYNC_ENCS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

REFP_VELO_POS (A, M)		MD 34070: $MA_REFP_VELO_POS		
MD 34070: $MA_REFP_VELO_POS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

REFP_VELO_SEARCH_CAM (A, M)		MD 34020: $MA_REFP_VELO_SEARCH_CAM		
MD 34020: $MA_REFP_VELO_SEARCH_CAM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

REFP_VELO_SEARCH_MARKER (A, M)		MD 34040: $MA_REFP_VELO_SEARCH_MARKER		
MD 34040: $MA_REFP_VELO_SEARCH_MARKER				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

ROT_IS_MODULO (A, M)		MD 30310: $MA_ROT_IS_MODULO		
Modulo conversion for rotary axis / spindle The variable defines whether an axis behaves like a linear axis during programming and positioning, or whether the modulo conversion is active for the rotary axis.				
Return value	o =1: There is a modulo conversion with the setpoint positions for the rotary axis. The software limit switch and the working area limitations are inactive, the traversing range is therefore infinite in both directions. o =0: No modulo conversion			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

SAFE_ACKN (A, M)		MD 36997: $MA_SAFE_ACKN		
MD 36997: $MA_SAFE_ACKN				
Units and value ranges				
Physical unit:	Data type:			

SAFE_ACKN (A, M)		MD 36997: $MA_SAFE_ACKN		
-	TYPE_DWORD			
			Read access	

SAFE_ACT_CHECKSUM (A, M)		MD 36998: $MA_SAFE_ACT_CHECK-SUM		
MD 36998: $MA_SAFE_ACT_CHECKSUM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_ACT_STOP_OUTPUT (A, M)		MD 36990: $MA_SAFE_ACT_STOP_OUT-PUT		
MD 36990: $MA_SAFE_ACT_STOP_OUTPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_CAM_MINUS_OUTPUT (A, M)		MD 36989: $MA_SAFE_CAM_MI-NUS_OUTPUT		
MD 36989: $MA_SAFE_CAM_MINUS_OUTPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_CAM_PLUS_OUTPUT (A, M)		MD 36988: $MA_SAFE_CAM_PLUS_OUTPUT		
MD 36988: $MA_SAFE_CAM_PLUS_OUTPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_CAM_POS_MINUS (A, M)		MD 36937: $MA_SAFE_CAM_POS_MINUS		
MD 36937: $MA_SAFE_CAM_POS_MINUS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_CAM_POS_PLUS (A, M)		MD 36936: $MA_SAFE_CAM_POS_PLUS		
MD 36936: $MA_SAFE_CAM_POS_PLUS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_CAM_TOL (A, M)		MD 36940: $MA_SAFE_CAM_TOL		
MD 36940: $MA_SAFE_CAM_TOL				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
			Read access	

SAFE_DES_CHECKSUM (A, M)		MD 36999: $MA_SAFE_DES_CHECKSUM		
MD 36999: $MA_SAFE_DES_CHECKSUM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_DES_VELO_LIMIT (A, M)		MD 36933: $MA_SAFE_DES_VELO_LIMIT		
MD 36933: $MA_SAFE_DES_VELO_LIMIT				
Units and value ranges				
Physical unit:	Data type:			

SAFE_DES_VELO_LIMIT (A, M)		MD 36933: $MA_SAFE_DES_VELO_LIMIT		
%	TYPE_DOUBLE			
			Read access	

SAFE_ENC_GEAR_DENOM (A, M)		MD 36921: $MA_SAFE_ENC_GEAR_DENOM		
MD 36921: $MA_SAFE_ENC_GEAR_DENOM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_ENC_GEAR_NUMERA (A, M)		MD 36922: $MA_SAFE_ENC_GEAR_NUMERA		
MD 36922: $MA_SAFE_ENC_GEAR_NUMERA				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_ENC_GEAR_PITCH (A, M)		MD 36920: $MA_SAFE_ENC_GEAR_PITCH		
MD 36920: $MA_SAFE_ENC_GEAR_PITCH				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
			Read access	

SAFE_ENC_GRID_POINT_DIST (A, M)		MD 36917: $MA_SAFE_ENC_GRID_POINT_DIST		
MD 36917: $MA_SAFE_ENC_GRID_POINT_DIST				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
			Read access	

SAFE_ENC_INPUT_NR (A, M)	MD 36912: $MA_SAFE_ENC_INPUT_NR	
MD 36912: $MA_SAFE_ENC_INPUT_NR		

SAFE_ENC_INPUT_NR (A, M)		MD 36912: $MA_SAFE_ENC_INPUT_NR		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SAFE_ENC_IS_LINEAR (A, M)		MD 36916: $MA_SAFE_ENC_IS_LINEAR		
MD 36916: $MA_SAFE_ENC_IS_LINEAR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

SAFE_ENC_MODULE_NR (A, M)		MD 36911: $MA_SAFE_ENC_MOD-ULE_NR		
MD 36911: $MA_SAFE_ENC_MODULE_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SAFE_ENC_POLARITY (A, M)		MD 36925: $MA_SAFE_ENC_POLARITY		
MD 36925: $MA_SAFE_ENC_POLARITY				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_ENC_RESOL (A, M)		MD 36918: $MA_SAFE_ENC_RESOL		
MD 36918: $MA_SAFE_ENC_RESOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_ENC_SEGMENT_NR (A, M)		MD 36910: $MA_SAFE_ENC_SEG-MENT_NR		
MD 36910: $MA_SAFE_ENC_SEGMENT_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SAFE_ENC_TYPE (A, M)		MD 36915: $MA_SAFE_ENC_TYPE		
MD 36915: $MA_SAFE_ENC_TYPE				

SAFE_ENC_TYPE (A, M)		MD 36915: $MA_SAFE_ENC_TYPE		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SAFE_EXT_STOP_INPUT (A, M)		MD 36977: $MA_SAFE_EXT_STOP_INPUT		
MD 36977: $MA_SAFE_EXT_STOP_INPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_FUNCTION_ENABLE (A, M)		MD 36901: $MA_SAFE_FUNCTION_ENABLE		
MD 36901: $MA_SAFE_FUNCTION_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_CAM_ENABLE (A, M)		MD 36903: $MA_SAFE_CAM_ENABLE		
MD 36903: $MA_SAFE_CAM_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_GEAR_SELECT_INPUT (A, M)		MD 36974: $MA_SAFE_GEAR_SELECT_INPUT		
MD 36974: $MA_SAFE_GEAR_SELECT_INPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_IS_ROT_AX (A, M)		MD 36902: $MA_SAFE_IS_ROT_AX		
MD 36902: $MA_SAFE_IS_ROT_AX				

SAFE_IS_ROT_AX (A, M)		MD 36902: $MA_SAFE_IS_ROT_AX		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

SAFE_MODE_SWITCH_TIME (A, M)		MD 36950: $MA_SAFE_MODE_SWITCH_TIME		
MD 36950: $MA_SAFE_MODE_SWITCH_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SAFE_MODULO_RANGE (A, M)		MD 36905: $MA_SAFE_MODU-LO_RANGE		
MD 36905: $MA_SAFE_MODULO_RANGE				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
			Read access	

SAFE_OVR_INPUT (A, M)		MD 36978: $MA_SAFE_OVR_INPUT		
MD 36978: $MA_SAFE_OVR_INPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_POS_LIMIT_MINUS (A, M)		MD 36935: $MA_SAFE_POS_LIMIT_MI-NUS		
MD 36935: $MA_SAFE_POS_LIMIT_MINUS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user de-fined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_POS_LIMIT_PLUS (A, M)		MD 36934: $MA_SAFE_POS_LIMIT_PLUS		
MD 36934: $MA_SAFE_POS_LIMIT_PLUS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_POS_SELECT_INPUT (A, M)		MD 36973: $MA_SAFE_POS_SELECT_INPUT		
MD 36973: $MA_SAFE_POS_SELECT_INPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_POS_STOP_MODE (A, M)		MD 36962: $MA_SAFE_POS_STOP_MODE		
MD 36962: $MA_SAFE_POS_STOP_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SAFE_POS_TOL (A, M)		MD 36942: $MA_SAFE_POS_TOL		
MD 36942: $MA_SAFE_POS_TOL				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
			Read access	

SAFE_PULSE_DISABLE_DELAY (A, M)		MD 36956: $MA_SAFE_PULSE_DISABLE_DELAY		
MD 36956: $MA_SAFE_PULSE_DISABLE_DELAY				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SAFE_PULSE_DIS_CHECK_TIME (A, M)		MD 36957: $MA_SAFE_PULSE_DIS_CHECK_TIME		
MD 36957: $MA_SAFE_PULSE_DIS_CHECK_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SAFE_PULSE_ENABLE_OUTPUT (A, M)		MD 36986: $MA_SAFE_PULSE_ENABLE_OUTPUT		
MD 36986: $MA_SAFE_PULSE_ENABLE_OUTPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_PULSE_STATUS_INPUT (A, M)		MD 36976: $MA_SAFE_PULSE_STATUS_INPUT		
MD 36976: $MA_SAFE_PULSE_STATUS_INPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_REFP_POS_TOL (A, M)		MD 36944: $MA_SAFE_REFP_POS_TOL		
MD 36944: $MA_SAFE_REFP_POS_TOL				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
			Read access	

SAFE_REFP_STATUS_OUTPUT (A, M)		MD 36987: $MA_SAFE_REFP_STATUS_OUTPUT		
MD 36987: $MA_SAFE_REFP_STATUS_OUTPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_SS_DISABLE_INPUT (A, M)		MD 36971: $MA_SAFE_SS_DISABLE_INPUT		
MD 36971: $MA_SAFE_SS_DISABLE_INPUT				
Units and value ranges				
Physical unit:	Data type:			

SAFE_SS_DISABLE_INPUT (A, M)		MD 36971: $MA_SAFE_SS_DISABLE_INPUT		
-	TYPE_DWORD			
			Read access	

SAFE_SS_STATUS_OUTPUT (A, M)		MD 36981: $MA_SAFE_SS_STATUS_OUTPUT		
MD 36981: $MA_SAFE_SS_STATUS_OUTPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_STANDSTILL_POS (A, M)		MD 36995: $MA_SAFE_STANDSTILL_POS		
MD 36995: $MA_SAFE_STANDSTILL_POS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_STANDSTILL_TOL (A, M)		MD 36930: $MA_SAFE_STANDSTILL_TOL		
MD 36930: $MA_SAFE_STANDSTILL_TOL				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
			Read access	

SAFE_STANDSTILL_VELO_TOL (A, M)		MD 36960: $MA_SAFE_STANDSTILL_VELO_TOL		
MD 36960: $MA_SAFE_STANDSTILL_VELO_TOL				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
			Read access	

SAFE_STOP_REQUEST_INPUT (A, M)		MD 36975: $MA_SAFE_STOP_REQUEST_INPUT		
MD 36975: $MA_SAFE_STOP_REQUEST_INPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_STOP_SWITCH_TIME_C (A, M)		MD 36952: $MA_SAFE_STOP_SWITCH_TIME_C		
MD 36952: $MA_SAFE_STOP_SWITCH_TIME_C				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SAFE_STOP_SWITCH_TIME_D (A, M)		MD 36953: $MA_SAFE_STOP_SWITCH_TIME_D		
MD 36953: $MA_SAFE_STOP_SWITCH_TIME_D				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SAFE_STOP_SWITCH_TIME_E (A, M)		MD 36954: $MA_SAFE_STOP_SWITCH_TIME_E		
MD 36954: $MA_SAFE_STOP_SWITCH_TIME_E				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SAFE_STOP_VELO_TOL (A, M)		MD 36948: $MA_SAFE_STOP_VE-LO_TOL		
MD 36948: $MA_SAFE_STOP_VELO_TOL				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user de-fined	TYPE_DOUBLE			
			Read access	

SAFE_SVSS_DISABLE_INPUT (A, M)		MD 36970: $MA_SAFE_SVSS_DISA-BLE_INPUT		
MD 36970: $MA_SAFE_SVSS_DISABLE_INPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_SVSS_STATUS_OUTPUT (A, M)		MD 36980: $MA_SAFE_SVSS_STA-TUS_OUTPUT		
MD 36980: $MA_SAFE_SVSS_STATUS_OUTPUT				

SAFE_SVSS_STATUS_OUTPUT (A, M)		MD 36980: $MA_SAFE_SVSS_STATUS_OUTPUT		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SAFE_VELO_LIMIT (A, M)		MD 36931: $MA_SAFE_VELO_LIMIT		
Limit value for safe velocity Specification of the limit values for the safe velocities 1, 2, 3 and 4. If SG1, SG2, SG3 or SG4 is selected and the current velocity exceeds this limit value, the control trips alarm 27011 with the stop response configured in MD36961 $MA_SAFE_VELO_STOP_MODE or MD36963 $MA_SAFE_VELO_STOP_REACTION. n = 0, 1, 2, 3 stand for the limit values of SG1, SG2, SG3, SG4				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_VELO_OVR_FACTOR (A, M)		MD 36932: $MA_SAFE_VELO_OVR_FACTOR		
MD 36932: $MA_SAFE_VELO_OVR_FACTOR				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_VELO_SELECT_INPUT (A, M)		MD 36972: $MA_SAFE_VELO_SELECT_INPUT		
MD 36972: $MA_SAFE_VELO_SELECT_INPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_VELO_STATUS_OUTPUT (A, M)		MD 36982: $MA_SAFE_VELO_STATUS_OUTPUT		
MD 36982: $MA_SAFE_VELO_STATUS_OUTPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_VELO_STOP_MODE (A, M)		MD 36961: $MA_SAFE_VELO_STOP_MODE		
MD 36961: $MA_SAFE_VELO_STOP_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SAFE_VELO_STOP_REACTION (A, M)		MD 36963: $MA_SAFE_VELO_STOP_REACTION		
MD 36963: $MA_SAFE_VELO_STOP_REACTION				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SAFE_VELO_SWITCH_DELAY (A, M)		MD 36951: $MA_SAFE_VELO_SWITCH_DELAY		
MD 36951: $MA_SAFE_VELO_SWITCH_DELAY				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SAFE_VELO_X (A, M)		MD 36946: $MA_SAFE_VELO_X		
MD 36946: $MA_SAFE_VELO_X				
Units and value ranges				
Physical unit:	Data type:			

SAFE_VELO_X (A, M)		MD 36946: $MA_SAFE_VELO_X		
mm/min, inch/min, user defined	TYPE_DOUBLE			
			Read access	

SAFE_VELO_X_STATUS_OUTPUT (A, M)		MD 36985: $MA_SAFE_VELO_X_STATUS_OUTPUT		
MD 36985: $MA_SAFE_VELO_X_STATUS_OUTPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SCALING_FACTOR_G70_G71 (A, M)		MD 31200: $MA_SCALING_FACTOR_G70_G71		
MD 31200: $MA_SCALING_FACTOR_G70_G71				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

SERVO_DISABLE_DELAY_TIME (A, M)		MD 36620: $MA_SERVO_DISABLE_DELAY_TIME		
MD 36620: $MA_SERVO_DISABLE_DELAY_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SIMU_AX_VDI_OUTPUT (A, M)		MD 30350: $MA_SIMU_AX_VDI_OUTPUT		
MD 30350: $MA_SIMU_AX_VDI_OUTPUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

SPIND_ACTIVE_AFTER_RESET (A, M)		MD 35040: $MA_SPIND_ACTIVE_AFTER_RESET		
MD 35040: $MA_SPIND_ACTIVE_AFTER_RESET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

SPIND_ASSIGN_TO_MACHAX (A, M)		MD 35000: $MA_SPIND_ASSIGN_TO_MACHAX		
MD 35000: $MA_SPIND_ASSIGN_TO_MACHAX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SPIND_DEFAULT_ACT_MASK (A, M)		MD 35030: $MA_SPIND_DEFAULT_ACT_MASK		
MD 35030: $MA_SPIND_DEFAULT_ACT_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SPIND_DEFAULT_MODE (A, M)		MD 35020: $MA_SPIND_DEFAULT_MODE		
MD 35020: $MA_SPIND_DEFAULT_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SPIND_DES_VELO_TOL (A, M)		MD 35150: $MA_SPIND_DES_VELO_TOL		
MD 35150: $MA_SPIND_DES_VELO_TOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

SPIND_EXTERN_VELO_LIMIT (A, M)		MD 35160: $MA_SPIND_EXTERN_VELO_LIMIT		
MD 35160: $MA_SPIND_EXTERN_VELO_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

SPIND_FUNC_RESET_MODE (A, M)		MD 35032: $MA_SPIND_FUNC_RESET_MODE		
MD 35032: $MA_SPIND_FUNC_RESET_MODE				
Units and value ranges				
Physical unit:	Data type:			

SPIND_FUNC_RESET_MODE (A, M)		MD 35032: $MA_SPIND_FUNC_RESET_MODE		
-	TYPE_DWORD			
			Read access	

SPIND_ON_SPEED_AT_IPO_START (A, M)		MD 35500: $MA_SPIND_ON_SPEED_AT_IPO_START		
MD 35500: $MA_SPIND_ON_SPEED_AT_IPO_START				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

SPIND_OSCILL_ACCEL (A, M)		MD 35410: $MA_SPIND_OSCILL_ACCEL		
MD 35410: $MA_SPIND_OSCILL_ACCEL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

SPIND_OSCILL_DES_VELO (A, M)		MD 35400: $MA_SPIND_OSCILL_DES_VELO		
MD 35400: $MA_SPIND_OSCILL_DES_VELO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

SPIND_OSCILL_START_DIR (A, M)		MD 35430: $MA_SPIND_OSCILL_START_DIR		
MD 35430: $MA_SPIND_OSCILL_START_DIR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SPIND_OSCILL_TIME_CCW (A, M)		MD 35450: $MA_SPIND_OSCILL_TIME_CCW		
MD 35450: $MA_SPIND_OSCILL_TIME_CCW				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SPIND_OSCILL_TIME_CW (A, M)		MD 35440: $MA_SPIND_OSCILL_TIME_CW		
MD 35440: $MA_SPIND_OSCILL_TIME_CW				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SPIND_POSCTRL_VELO (A, M)		MD 35300: $MA_SPIND_POSCTRL_VELO		
MD 35300: $MA_SPIND_POSCTRL_VELO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

SPIND_POSITIONING_DIR (A, M)		MD 35350: $MA_SPIND_POSITIONING_DIR		
MD 35350: $MA_SPIND_POSITIONING_DIR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SPIND_STOPPED_AT_IPO_START (A, M)		MD 35510: $MA_SPIND_STOPPED_AT_IPO_START		
MD 35510: $MA_SPIND_STOPPED_AT_IPO_START				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

SPIND_VELO_LIMIT (A, M)		MD 35100: $MA_SPIND_VELO_LIMIT		
MD 35100: $MA_SPIND_VELO_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

STANDSTILL_DELAY_TIME (A, M)		MD 36040: $MA_STANDSTILL_DELAY_TIME		
MD 36040: $MA_STANDSTILL_DELAY_TIME				
Units and value ranges				
Physical unit:	Data type:			

STANDSTILL_DELAY_TIME (A, M)		MD 36040: $MA_STANDSTILL_DELAY_TIME		
s, user defined	TYPE_DOUBLE			
			Read access	

STANDSTILL_POS_TOL (A, M)		MD 36030: $MA_STANDSTILL_POS_TOL		
MD 36030: $MA_STANDSTILL_POS_TOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

STANDSTILL_VELO_TOL (A, M)		MD 36060: $MA_STANDSTILL_VELO_TOL		
MD 36060: $MA_STANDSTILL_VELO_TOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

START_STOP_FREQUENCY (A, M)		MD 31300: $MA_START_STOP_FREQUENCY		
MD 31300: $MA_START_STOP_FREQUENCY				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

STIFFNESS_CONTROL_ENABLE (A, M)		MD 32640: $MA_STIFFNESS_CONTROL_ENABLE		
MD 32640: $MA_STIFFNESS_CONTROL_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

STOP_LIMIT_COARSE (A, M)		MD 36000: $MA_STOP_LIMIT_COARSE		
MD 36000: $MA_STOP_LIMIT_COARSE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

STOP_LIMIT_FINE (A, M)		MD 36010: $MA_STOP_LIMIT_FINE		
MD 36010: $MA_STOP_LIMIT_FINE				

STOP_LIMIT_FINE (A, M)		MD 36010: $MA_STOP_LIMIT_FINE		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TANG_OFFSET (A, M)		MD 37402: $MA_TANG_OFFSET		
MD 37402: $MA_TANG_OFFSET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TEMP_COMP_TYPE (A, M)		MD 32750: $MA_TEMP_COMP_TYPE		
MD 32750: $MA_TEMP_COMP_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

TORQUE_OFFSET (A, M)		MD 32460: $MA_TORQUE_OFFSET		
MD 32460: $MA_TORQUE_OFFSET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

VCO100_WEIGHT (A, M)		MD 31160: $MA_VCO100_WEIGHT		
MD 31160: $MA_VCO100_WEIGHT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

VCO25_WEIGHT (A, M)		MD 31130: $MA_VCO25_WEIGHT		
MD 31130: $MA_VCO25_WEIGHT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

VCO50_WEIGHT (A, M)		MD 31140: $MA_VCO50_WEIGHT		
MD 31140: $MA_VCO50_WEIGHT				

VCO50_WEIGHT (A, M)		MD 31140: $MA_VCO50_WEIGHT		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

VCO75_WEIGHT (A, M)		MD 31150: $MA_VCO75_WEIGHT		
MD 31150: $MA_VCO75_WEIGHT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

VELO_FFW_WEIGHT (A, M)		MD 32610: $MA_VELO_FFW_WEIGHT		
MD 32610: $MA_VELO_FFW_WEIGHT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

3.2.1.2 SE: Setting data

This block contains the axis-specific setting data

AA_OFF_LIMIT (A, SE)		SD 43350: $SA_AA_OFF_LIMIT		
Compensation value upper limit $AA_OFF with clear Upper limit of compensation value which can be preset by means of synchronized actions via the system variable $AA_OFF. This limit value acts on the absolutely effective compensation value via $AA_OFF. It is possible to interrogate the compensation value for limit-range violation via the system variable $AA_OFF_LIMIT.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

ASSIGN_FEED_PER_REV_SOURCE (A, SE)		SD 43300: $SA_AS-SIGN_FEED_PER_REV_SOURCE		
SD 43300: $SA_ASSIGN_FEED_PER_REV_SOURCE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

FIXED_STOP_SWITCH (A, SE)		SD 43500: $SA_FIXED_STOP_SWITCH		
SD 43500: $SA_FIXED_STOP_SWITCH				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FIXED_STOP_TORQUE (A, SE)		SD 43510: $SA_FIXED_STOP_TORQUE		
SD 43510: $SA_FIXED_STOP_TORQUE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

FIXED_STOP_WINDOW (A, SE)		SD 43520: $SA_FIXED_STOP_WINDOW		
SD 43520: $SA_FIXED_STOP_WINDOW				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

LEAD_OFFSET_IN_POS (A, SE)		SD 43102: $SA_LEAD_OFFSET_IN_POS		
SD 43102: $SA_LEAD_OFFSET_IN_POS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

LEAD_OFFSET_OUT_POS (A, SE)		SD 43106: $SA_LEAD_OFF-SET_OUT_POS		
SD 43106: $SA_LEAD_OFFSET_OUT_POS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

LEAD_SCALE_IN_POS (A, SE)		SD 43104: $SA_LEAD_SCALE_IN_POS		
SD 43104: $SA_LEAD_SCALE_IN_POS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

LEAD_SCALE_OUT_POS (A, SE)		SD 43108: $SA_LEAD_SCALE_OUT_POS		
SD 43108: $SA_LEAD_SCALE_OUT_POS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

LEAD_TYPE (A, SE)		SD 43100: $SA_LEAD_TYPE		
SD 43100: $SA_LEAD_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MDB_WORKAREA_MINUS_ENABLE (A, SE)		SD 43410: $SA_MDB_WORKAREA_MINUS_ENABLE		
Alternative name:	WORKAREA_MINUS_ENABLE			
Working area limitation active in the negative dir Working area limitation active in the negative direction				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDB_WORKAREA_PLUS_ENABLE (A, SE)		SD 43400: $SA_MDB_WORKAREA_PLUS_ENABLE		
Alternative name:	WORKAREA_PLUS_ENABLE			
Working area limitation active in the positive dir Working area limitation active in the positive direction				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDD_SPIND_MAX_VELO_G26 (A, SE)		SD 43220: $SA_MDD_SPIND_MAX_VELO_G26		
Alternative name:	SPIND_MAX_VELO_G26			
Maximum spindle speed at G26 Maximum spindle speed at G26 (master spindle)				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDD_SPIND_MAX_VELO_LIMS (A, SE)		SD 43230: $SA_MDD_SPIND_MAX_VELO_LIMS		
Alternative name:	SPIND_MAX_VELO_LIMS			
Spindle speed limitation Spindle speed limitation (master spindle)				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDD_SPIND_MIN_VELO_G25 (A, SE)		SD 43210: $SA_MDD_SPIND_MIN_VELO_G25		
Alternative name:	SPIND_MIN_VELO_G25			
Minimum spindle speed at G25 Minimum spindle speed at G25 (master spindle)				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDD_WORKAREA_LIMIT_MINUS (A, SE)		SD 43430: $SA_MDD_WORKAREA_LIMIT_MINUS		
Alternative name:	WORKAREA_LIMIT_MINUS			
Working area limitation in the negative direction				

MDD_WORKAREA_LIMIT_MINUS (A, SE)		SD 43430: $SA_MDD_WORKAREA_LIMIT_MINUS		
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDD_WORKAREA_LIMIT_PLUS (A, SE)		SD 43420: $SA_MDD_WORKAREA_LIMIT_PLUS		
Alternative name:	WORKAREA_LIMIT_PLUS			
Working area limitation in the positive direction				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

OSCILL_CTRL_MASK (A, SE)		SD 43770: $SA_OSCILL_CTRL_MASK		
SD 43770: $SA_OSCILL_CTRL_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

OSCILL_DWELL_TIME1 (A, SE)		SD 43720: $SA_OSCILL_DWELL_TIME1		
SD 43720: $SA_OSCILL_DWELL_TIME1				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

OSCILL_DWELL_TIME2 (A, SE)		SD 43730: $SA_OSCILL_DWELL_TIME2		
SD 43730: $SA_OSCILL_DWELL_TIME2				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

OSCILL_END_POS (A, SE)		SD 43760: $SA_OSCILL_END_POS		
SD 43760: $SA_OSCILL_END_POS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

OSCILL_IS_ACTIVE (A, SE)		SD 43780: $SA_OSCILL_IS_ACTIVE		
SD 43780: $SA_OSCILL_IS_ACTIVE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

OSCILL_NUM_SPARK_CYCLES (A, SE)		SD 43750: $SA_OS-CILL_NUM_SPARK_CYCLES		
SD 43750: $SA_OSCILL_NUM_SPARK_CYCLES				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

OSCILL_REVERSE_POS1 (A, SE)		SD 43700: $SA_OSCILL_RE-VERSE_POS1		
SD 43700: $SA_OSCILL_REVERSE_POS1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

OSCILL_REVERSE_POS2 (A, SE)		SD 43710: $SA_OSCILL_RE-VERSE_POS2		
SD 43710: $SA_OSCILL_REVERSE_POS2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

OSCILL_VELO (A, SE)		SD 43740: $SA_OSCILL_VELO		
SD 43740: $SA_OSCILL_VELO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TEMP_COMP_ABS_VALUE (A, SE)		SD 43900: $SA_TEMP_COMP_ABS_VALUE		
Positn.-independent temperature compensation value The variable specifies the position-independent temperature compensation value. This value must be specified according to the current temperature of the PLC (user program). As soon as the position-independent temperature compensation becomes active (MD32750 $MA_TEMP_COMP_TYPE = 1 or 3), the machine axis also traverses this compensation value. The calculation principles must be defined by the machine manufacturer (measurement of temperature, offset tables etc.).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TEMP_COMP_REF_POSITION (A, SE)		SD 43920: $SA_TEMP_COMP_REF_POSITION		
SD 43920: $SA_TEMP_COMP_REF_POSITION				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TEMP_COMP_SLOPE (A, SE)		SD 43910: $SA_TEMP_COMP_SLOPE		
SD 43910: $SA_TEMP_COMP_SLOPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

3.2.2 Bag: Mode Group Data

3.2.2.1 S: State data

During NC operation different internal states occur and system-specific data may change during operation. To distinguish those from system variables, they are classified as state data.

A distinction is made between:
- NCK-specific state data
- Mode-group-specific state data
- Channel-specific state data
- Drive-specific state data (FDD)
- Drive-specific state data (MSD)

ncAutoCounter (B, S)				
Counter for Auto key Counter which is incremented with each 0->edge of the Auto key				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numBAGs	
			Read access	

ncJogCounter (B, S)				
Counter for Jog key Counter which is incremented with each 0->edge of the Jog key				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numBAGs	
			Read access	

ncMDACounter (B, S)				
Counter for MDI key Counter which is incremented with each 0->edge of the MDI key				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numBAGs	
			Read access	

opMode (B, S)		DB11, DBXn6.0-n6.2 (n=0,2,4,6, ...)		
Active mode				
Return value	0 = JOG 1 = MDI 2 = AUTO			
Units and value ranges				
Physical unit:	Data type:			

opMode (B, S)		DB11, DBXn6.0-n6.2 (n=0,2,4,6, ...)		
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

readyActive (B, S)		DB11, DBX6.3, DBX26.3, DBX46.3, ...		
Mode group ready Code whether mode group is ready				
Return value	0 = not ready 1 = ready			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

resetActive (B, S)		DB11, DBX6.7, DBX26.7, DBX46.7, ...		
All channels of mode group in Reset Code whether all channels in mode group are in Reset				
Return value	0 = not all channels in reset 1 = all channels in reset			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

autoJogState (B, S)		$AC_AUTO_JOG_STATE		
Status of Automatic+JOG mode				
Return value	1: Automatic is selected, $MN_JOG_MODE_MASK is set, and the mode group (BAG) is in "Reset". A JOG motion can then be initiated by pressing the +/- buttons or turning the hand wheel in Auto. 2: This mode group has been switched internally to JOG on account of a JOG motion. VDI and OPI still show Automatic. 0: In all other cases			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$numBAGs	
			Read access	

3.2.3 Channel: Channel-assigned data

3.2.3.1 AUXFU: Active auxiliary functions

The module includes the active auxiliary functions for each group.
In the line, the auxiliary function group (64 groups) and the
desired view are addressed:
Line 1001-1064: Active auxiliary function from the point of view of the NCK
Line 2001-2064: Collected auxiliary function (after search run) from the point of view of the NCK
Line 3001-3064: Active auxiliary function from the point of view of the PLC
Line 1-64: Summary of the above views

Only the values of lines 3001-3064 can be written.
When writing individual values, it must be taken care
that the status variable is written last.
The entire data block of an auxiliary function will not be
accepted before this variable is written.

extension (C, AUXFU)		$AC_AUXFU_EXT[groupIndex]		
Extension of the auxiliary function				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			3128	
			Read access and write access	

status (C, AUXFU)				
Status of the auxiliary function				
Return value	Bit0 = 1: Auxiliary function has been collected (NCK view) Bit1 = 1: Auxiliary function has been output to PLC (NCK view) Bit2 = 1: Auxiliary function has been acknowledged by PLC (NCK view) Bit3 = 1: Auxiliary function has been acknowledged by PLC (PLC view) Bit4 = 1: Auxiliary function has been functionally completed (PLC view) Bit14 = 1: Value type is LONG Bit15 = 1: Value type is DOUBLE			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	

status (C, AUXFU)		
	3128	
	Read access and write access	

type (C, AUXFU)		$AC_AUXFU_TYPE[groupIndex]		
Type of the auxiliary function Type of the auxiliary function, e.g. "M", "S", "T", "D", "F", "H", "L".				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			3128	
			Read access and write access	

valueDo (C, AUXFU)		$AC_AUXFU_VALUE[groupIndex]		
Double value of the auxiliary function Value of the auxiliary function. This value will be supplied, if "status" Bit15 = 1				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			3128	
			Read access and write access	

valueLo (C, AUXFU)		$AC_AUXFU_M_VALUE[groupIndex]		
Long value of the auxiliary function Value of the auxiliary function. This value will be supplied, if "status" Bit14 = 1				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			3128	
			Read access and write access	

acAuxfuMTick (C, AUXFU)		$AC_AUXFU_M_TICK[groupIndex]		
Time stamp of the active auxiliary M function The variable is used to read the time stamp of the last auxiliary function group collected (search function) or output for an auxiliary function group. If no auxiliary function has been output for a specified group yet, the variable provides value -1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	-1	INT_MIN	INT_MAX
Addressing				
Line index:			Max. line index:	
			3128	
			Read access and write access	

acAuxfuPredefIndex (C, AUXFU)		$AC_AUXFU_PREDEF_INDEX[groupIndex]		
Pre-defined index of the active auxiliary function The variable is used to read the pre-defined index of the last auxiliary function group collected (search function) or output for an auxiliary function group. If no auxiliary function has been output for a specified group yet, the variable provides value -1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	-1	-1	INT_MAX
Addressing				
Line index:			Max. line index:	
			3064	
			Read access and write access	

acAuxfuSpec (C, AUXFU)	$AC_AUXFU_SPEC[groupIndex]	

Output specif. of the active auxiliary function

The variable is used to read the output specification of the last auxiliary function group collected (search function) or output for an auxiliary function group.
If no auxiliary function has been output for a specified group yet, the variable provides value -1.

The output specification is bit-coded:
Bit 0 = 1 Acknowledgment "normal" after an OB1 cycle
Bit 1 = 1 Acknowledgment "quick" with OB40
Bit 2 = 1 No pre-defined auxiliary function
Bit 3 = 1 No output to the PLC
Bit 4 = 1 Spindle reaction after acknowledgment by the PLC
Bit 5 = 1 Output before movement
Bit 6 = 1 Output during movement
Bit 7 = 1 Output at block end
Bit 8 = 1 No output after block search type 1,2,4
Bit 9 = 1 Collection during block search type 5 (SERUPRO)
Bit 10 = 1 No output during block search type 5 (SERUPRO)
Bit 11 = 1 Cross-channel auxiliary function (SERUPRO)
Bit 12 = 1 Output made through synchronized action
Bit 13 = 1 Implicit auxiliary function
Bit 14 = 1 Active M01
Bit 15 = 1 No output during travel-in test run
Bit 16 = 1 Nibbling OFF
Bit 17 = 1 Nibbling ON
Bit 18 = 1 Nibbling

Units and value ranges

Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	-1	INT_MIN	INT_MAX

Addressing

Line index:	Max. line index:
	3064
	Read access and write access

acAuxfuTickSeq (C, AUXFU)	$AC_AUXFU_TICK[groupIndex,0]	

Sequence counter of the active auxiliary function

The variable is used to read the output sequence counter (all outputs within an IPO cycle) of the last auxiliary function group collected (search function) or output for an auxiliary function group.

Units and value ranges

Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	INT_MIN	INT_MAX

acAuxfuTickSeq (C, AUXFU)	$AC_AUXFU_TICK[groupIndex,0]	
Addressing		
Line index:	Max. line index:	
	3064	
	Read access and write access	

acAuxfuTickPack (C, AUXFU)		$AC_AUXFU_TICK[groupIndex,1]		
Package counter of the active auxiliary function The variable is used to read the package counter per sequence of the last auxiliary function group collected (search function) or output for an auxiliary function group.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	INT_MIN	INT_MAX
Addressing				
Line index:			Max. line index:	
			3064	
			Read access and write access	

acAuxfuTickHifu (C, AUXFU)		$AC_AUXFU_TICK[groupIndex,2]		
Auxil. funct. counter of the act. auxil. function The variable is used to read the auxiliary function counter per package of the last auxiliary function group collected (search function) or output for an auxiliary function group.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	INT_MIN	INT_MAX
Addressing				
Line index:			Max. line index:	
			3064	
			Read access and write access	

3.2.3.2 CP: Generic coupling

This block contains the data of the generic coupling.

aaCpNumDefLa (C, CP)		$AA_CPNDEFLA[a]		
Number of defined leading axes The number of leading axes which have been defined for the specified following axis				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	

aaCpNumDefLa (C, CP)	$AA_CPNDEFLA[a]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

aaCpDefLa (C, CP)		$AA_CPDEFLA[ax,n]		
Defined leading axis of the following axis The axis index of the nth leading axis which has been defined for the specified following axis				
Return value	-1 = the specified coupling is not defined or n == 0 or n > $$aaCpNumDefLa (= number of defined leading axes of the following axis)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	-1	-1	
Addressing				
Line index:			Max. line index:	
			Read access	

aaCpNumActLa (C, CP)		$AA_CPNACTLA[a]		
Number of active leading axes The number of leading axes which are active for the specified following axis				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpActLa (C, CP)		$AA_CPACTLA[ax,n]		
Active leading axis of the following axis The axis index of the nth leading axis which is active for the specified following axis				
Return value	-1 = the specified coupling is not active or n == 0 or n > $$aaCpNumActLa (= number of active leading axes of the following axis)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	-1	-1	
Addressing				
Line index:			Max. line index:	
			Read access	

aaCpNumActFa (C, CP)		$AA_CPNACTFA[ax]		
Number of active following axes Number of couplings (following axes), in which the specified axis LAx is active as leading axis				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpActFa (C, CP)		$AA_CPACTFA[ax,n]		
Active following axis of the leading axis The axis index of the following axis of the nth coupling, in which the specified axis LAx is active as leading axis				
Return value	-1 = the following axis found is unknown in the channel or n == 0 or n > $$aaCpNumActFa (= number of active couplings of the axis as leading axis)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	-1	-1	
Addressing				
Line index:			Max. line index:	
			Read access	

aaCpBlockChg (C, CP)		$AA_CPBC[a]		
Block change criterion for the following axis The block change criterion indicates the condition that has to be fulfilled before one can continue with the next block of the NC program if a coupling has been activated for the stated following axis, FAx				
Return value	NONE - Block change is performed immediately FINE - Block change is performed with "Synchronism fine" COARSE - Block change is performed with "Synchronism coarse" IPOSTOP - Block change is performed with "Setpoint synchronism"			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpfModeOff (C, CP)		$AA_CPFMOF[a]		
Coupling deactivation mode Identifies the behavior of the following axis if the coupling is deactivated				
Return value	STOP - Stop following axis/spindle CON - Continue movement with the current velocity ADD			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpfActive (C, CP)		$AA_CPFACT[a]		
Types of active coupling Bit-coded for identifying all types of coupling which are active for the stated following axis, FAx				
Return value	0 = NONE - No active coupling to the following axis Bit 0 (0x0001) - TRAIL - Uses a coupling factor Bit 1 (0x0002) - LEAD - Uses a curve table Bit 2 (0x0004) - ELG - An electronic gearbox link Bit 3 (0x0008) - Reserved Bit 4 (0x0010) - COUP - Spindle/partial spindle coupling Bit 5 (0x0020) - GANTRY - Coupling of the split axes (axes mechanically bound) Bit 6 (0x0040) - TANG - Tangential coupling using a curve table Bit 7 (0x0080) - GEN_CP - Generic coupling			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpfRS (C, CP)		$AA_CPFRS[a]		
Reference system for the following axis The reference system indicates the point at which the coupling process is applied				
Return value	BCS - Basic coordinate system MCS - Machine coordinate system			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

aaCpfRS (C, CP)	$AA_CPFRS[a]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

aaCpfCmdPosTotal (C, CP)		$AA_CPFCMDPT[a]		
Sum of all position commands of all leading axes Proportion of the axis position command due to the coupled axes. The sum of the dependent proportion of the position command for all leading axes for the stated following axis, FAx.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpfCmdVelTotal (C, CP)		$AA_CPFCMDVT[a]		
Sum of all velocity commands of all leading axes Proportion of the axis position command due to the coupled axes. The sum of the dependent proportion of the velocity command for all leading axes for the specified following axis.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpfReqVelocity (C, CP)		$AA_CPFREQV[a]		
Required coupling velocity Returns the speed requested by the active following axes/spindles.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCplType (C, CP)		$AA_CPLTYPE[a,b]		
Type of coupling Indicates the process that is used with the coupling of the stated following axis with the stated leading axis.				
Return value	0 = NONE - No defined coupling with these axes Bit 0 (0x0001) - TRAIL - Uses a coupling factor Bit 1 (0x0002) - LEAD - Uses a curve table Bit 2 (0x0004) - ELG - An electronic gearbox link Bit 3 (0x0008) - Reserved Bit 4 (0x0010) - COUP - Spindle/partial spindle coupling Bit 5 (0x0020) - GANTRY - Coupling of the split axes (axes mechanically bound) Bit 6 (0x0040) - TANG - Tangential coupling with the aid of a curve table Bit 7 (0x0080) - GEN_CP - Generic coupling			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaCplState (C, CP)		$AA_CPLSTATE[a,b]		
The status of the coupling A string which describes the actual status of the coupling				
Return value	DEF = Defined (but not yet activated) ON = Active OF = Deactivated			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaCplCTabId (C, CP)		$AA_CPLCTID[a,b]		
The ID of the active curve table ID number of the curve table which is used with the coupling of the stated axes.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

aaCplCTabId (C, CP)	$AA_CPLCTID[a,b]	
	$$numMachAxes * $$numMachAxes	
	Read access	

aaCplNumerator (C, CP)		$AA_CPLNUM[a,b]		
Counter of coupling factor				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaCplDenominator (C, CP)		$AA_CPLDEN[a,b]		
Denominator of coupling factor				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaCplCmdPos (C, CP)		$AA_CPLCMDP[a,b]		
Position command of the stated coupling The proportion of the axis position command assigned to the stated coupling.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaCplCmdVel (C, CP)		$AA_CPLCMDV[a,b]		
Velocity command of the stated coupling The proportion of the axis acceration command assigned to the stated coupling.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

aaCplCmdVel (C, CP)	$AA_CPLCMDV[a,b]	
	$$numMachAxes * $$numMachAxes	
	Read access	

aaCplRS (C, CP)	$AA_CPLRS[a,b]			
Reference system for the stated coupling Reference system for the specified coupling				
Return value	Reference system for the specified coupling DescriptionValue range: BCS - Basic coordinate system MCS - Machine coordinate system			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaCpfAccelTotal (C, CP)	$AA_CPFACCT[a]			
Proportion of axis acceleration of all couplings Proportion of axis acceleration due to the coupled axes. The sum of the dependent proportion of the acceleration of all leading axes for the stated following axis, FAx.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCplAccel (C, CP)	$AA_CPLACC[a,b]			
Acceleration proportion of the leading axis Acceleration proportion of the following axis caused by an active coupling to the specified leading axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaCplSetVal (C, CP)		$AA_CPLSETVAL[a,b]		
Type of defined value for the coupling Indicates the type of defined value used for the coupling				
Return value	ACTPOS = Actual position CMDPOS = Setpoint position CMDVEL = Setpoint velocity			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaCpfModeOn (C, CP)		$AA_CPFMON[a]		
Coupling activation mode Indicates the behavior of the following axis, FAx, when the coupling is activated.				
Return value	STOP - Stop following axis/spindle CON - Continue motion of the FAx with the current velocity ADD			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpfMSOn (C, CP)		$AA_CPFMSON[a]		
Following axis mode On strategy Indicates the activation strategy of the following axis				
Return value	CNONE CFAST COARSE NTG ACN ACP DCT NTGP DCP			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

aaCpfMSOn (C, CP)	$AA_CPFMSON[a]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

aaCpMReset (C, CP)		$AA_CPMRESET[a]		
Coupling mode through RESET				
Return value	NONE ON OFF DEL			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpMStart (C, CP)		$AA_CPMSTART[a]		
Coupling mode through program start				
Return value	NONE ON OFF DEL			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpMStartPrt (C, CP)		$AA_CPMSTARTPRT[a]		
Coupling mode through SERUPRO start				
Return value	NONE ON OFF DEL			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

aaCpMStartPrt (C, CP)	$AA_CPMSTARTPRT[a]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

aaCpSetType (C, CP)		$AA_CPSETTYPE[a]		
Coupling set coupling type				
Return value	NONE TRAIL LEAD EG COUP			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:		Max. line index:		
		$$numMachAxes		
		Read access		

aaCplInTrans (C, CP)		$AA_CPLINTR[a,b]		
Input transmission correction of coupling factor				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		$$numMachAxes * $$numMachAxes		
		Read access		

aaCplInScale (C, CP)		$AA_CPLINSC[a,b]		
Input scaling factor of coupling factor				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		$$numMachAxes * $$numMachAxes		
		Read access		

aaCplOutTrans (C, CP)	$AA_CPLOUTTR[a,b]	
Output transmission correction of coupling factor		

aaCplOutTrans (C, CP)		$AA_CPLOUTTR[a,b]		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaCplOutScale (C, CP)		$AA_CPLOUTSC[a,b]		
Output scaling factor of coupling factor				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaCpSynCoPos (C, CP)		$AA_CPSYNCOP[a]		
Synchronization coarse positioning tolerance Coarse positioning tolerance for coupling synchronization				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpSynFiPos (C, CP)		$AA_CPSYNFIP[a]		
Synchronization fine positioning tolerance Fine positioning tolerance for coupling synchronization				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpSynCoVel (C, CP)	$AA_CPSYNCOV[a]	
Synchronization coarse velocity tolerance Coarse velocity tolerance for coupling synchronization		

aaCpSynCoVel (C, CP)		$AA_CPSYNCOV[a]		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpSynFiVel (C, CP)		$AA_CPSYNFIV[a]		
Synchronization fine velocity tolerance Fine velocity tolerance for coupling synchronization				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpMVdi (C, CP)		$AA_CPMVDI[a]		
Behavior of the coupl. module regard. VDI signals Behavior of the coupling module regarding VDI signals				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpSynCoPos2 (C, CP)		$AA_CPSYNCOP2[a]		
Threshold value second synchr. monitoring coarse Second synchronism monitoring of the following axis / spindle: threshold value coarse				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpSynFiPos2 (C, CP)		$AA_CPSYNFIP2[a]		
Threshold value second synchr. monitoring fine Second synchronism monitoring of the following axis / spindle: threshold value fine				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCpMAlarm (C, CP)		$AA_CPMALARM[a]		
Behavior of coupling module re handling of alarms Behavior of coupling module regarding suppression of alarms				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

3.2.3.3 DIAG: Diagnostics data, available only for internal development purposes

This block contains channel-specific diagnostic data.

actCycleTimeNet (C, DIAG)				
actCycleTimeNet				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
			Read access	

minCycleTimeNet (C, DIAG)				
minCycleTimeNet				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
			Read access and write access	

maxCycleTimeNet (C, DIAG)				
maxCycleTimeNet				

maxCycleTimeNet (C, DIAG)				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
			Read access and write access	

actCycleTimeBrut (C, DIAG)				
actCycleTimeBrut				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

minCycleTimeBrut (C, DIAG)				
minCycleTimeBrut				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
			Read access and write access	

maxCycleTimeBrut (C, DIAG)				
maxCycleTimeBrut				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
			Read access and write access	

availableBodies (C, DIAG)				
availableBodies				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

numConfigBodies (C, DIAG)				
numConfigBodies				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

maxAllocBodies (C, DIAG)				
maxAllocBodies				

maxAllocBodies (C, DIAG)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

bodySize (C, DIAG)				
bodySize				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

ipoBufLevel (C, DIAG)				
ipoBufLevel				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

handleDebugFlags (C, DIAG)				
handleDebugFlags				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

internalBlockNumber (C, DIAG)				
internalBlockNumber				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

resetCounter (C, DIAG)				
resetCounter				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

reposCounter (C, DIAG)				
reposCounter				

reposCounter (C, DIAG)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

fifoFileActNetData (C, DIAG)				
fifoFileActNetData				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

countOfFreeByteBrut (C, DIAG)				
countOfFreeByteBrut				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

ncscTraceCounter (C, DIAG)				
ncscTraceCounter				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

varAccessError (C, DIAG)				
varAccessError				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

theErrorObjOK (C, DIAG)				
theErrorObjOK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

testReqVarServerSuppr (C, DIAG)				
testReqVarServerSuppr				

testReqVarServerSuppr (C, DIAG)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

testInchMetr (C, DIAG)				
testInchMetr				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

stackSizeUsed (C, DIAG)				
stackSizeUsed				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

nextSLSubtype (C, DIAG)				
nextSLSubtype				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access and write access	

COTraceFlags (C, DIAG)				
COTraceFlags				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access and write access	

coCliStiCounterMax (C, DIAG)				
coCliStiCounterMax				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

coCliTimestampMax (C, DIAG)				
coCliTimestampMax				

coCliTimestampMax (C, DIAG)				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DWORD			
			Read access	

coCliTimestampMaxStartup (C, DIAG)				
coCliTimestampMaxStartup				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DWORD			
			Read access	

protocVLTime (C, DIAG)				
protocVLTime				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

interferenceCheckOn (C, DIAG)				
interferenceCheckOn				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access and write access	

ipoCounter (C, DIAG)				
ipoCounter				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

stopCondSet (C, DIAG)				
stopCondSet				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

taskRuntimeInProg (C, DIAG)				
taskRuntimeInProg				

taskRuntimeInProg (C, DIAG)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

asyncOrderForWaitTime (C, DIAG)				
Asynchronous order in the EES subtask Asynchronous order for a wait time that is executed in the EES subtask. Use: Simulation of slow external drives.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access and write access	

3.2.3.4 DIAGN: Diagnostic module

This module contains information about channel-specific NC diagnostic data.
Net times: Time without interrupts by higher priority time levels.
Gross times: Time with interrupts by higher priority time levels.
Time levels in order of their priority: position controller, interpolator, block preparation.

aclpoBuf (C, DIAGN)		$AC_IPO_BUF		
Level of IPO buffer Level of IPO buffer (number of blocks)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actCycleTimeBrut (C, DIAGN)				
Current gross cycle time Current gross run time in ms.				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

actCycleTimeNet (C, DIAGN)				
Current net cycle time Current net run time in ms.				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

cuttingTime (C, DIAGN)		$AC_CUTTING_TIME		
Tool operating time (in seconds). Tool operating time (in seconds): The operating time of the path axes excluding active rapid traverse is measured in all NC programs between NC Start and Program End/NC Reset. The measurement is also interrupted during an active dwell time. The timer is automatically set to zero every time the control boots on default values.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

cycleTime (C, DIAGN)		$AC_CYCLE_TIME		
Runtime of selected NC program Runtime of selected NC program (in seconds): The runtime between NC Start and Program End / NC Reset is measured in the selected NC program. The timer is cleared when a new NC program is started.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

ipoBufLevel (C, DIAGN)				
Fill level of the IPO buffer (integer value in %)				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
%	TYPE_UWORD		0	100
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

maxCycleTimeBrut (C, DIAGN)				
Maximum gross cycle time Maximum gross run time in ms.				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			100	
			Read access and write access	

maxCycleTimeBrutPo (C, DIAGN)				
Maximum gross run time Maximum gross run time since cold restart in ms.				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			100	
			Read access and write access	

maxCycleTimeNet (C, DIAGN)				
Maximum net cycle time Maximum net run time in ms.				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			100	
			Read access and write access	

maxCycleTimeNetPo (C, DIAGN)				
Maximum net run time Maximum net run time since cold restart in ms.				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			100	
			Read access and write access	

minCycleTimeBrut (C, DIAGN)				
minimum gross cycle time Minimum gross run time in ms.				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			100	
			Read access and write access	

minCycleTimeBrutPo (C, DIAGN)				
Minimum gross run time Minimum gross run time since cold restart in ms.				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			100	
			Read access and write access	

minCycleTimeNet (C, DIAGN)				
Minimum net cycle time Minimum net run time in ms.				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			100	
			Read access and write access	

minCycleTimeNetPo (C, DIAGN)				
Minimum net run time Minimum net run time since cold restart in ms.				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			100	
			Read access and write access	

operatingTime (C, DIAGN)		$AC_OPERATING_TIME		
Total runtime NC programs Total runtime of NC programs in Automatic mode (in seconds): The runtimes of all programs are summed between NC Start and Program End/NC Reset. The timer is set to zero on every control boot.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

aveCycleTimeNet (C, DIAGN)				
Average net run time Average net run time in s				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

sumCycleTimeNet (C, DIAGN)				
Sum of net run times Sum of net run times in ms				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			

sumCycleTimeNet (C, DIAGN)				
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

taskName (C, DIAGN)				
Name of task				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

taskAvailable (C, DIAGN)				
Task is present				
Return value	The reasons are bit-coded. 0: Task not present 1: Task present			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_UWORD		0	
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

3.2.3.5 EPSP: Reserved for internal use only (ePS).

Parameters reserved for internal use only (ePS).

There is one EPSP module in each active channel. Each EPSP module holds six modifiable parameter arrays of different data types.

The dimension of each parameter array is given by MD 18840: $MN_MM_EPSPARAM_DIMENSION. There is only one EPSPARAM_DIMENSION value for the system. Every EPSP array in every EPSP module has this dimension. The value can range from 0 (= no parameters) to 100. The default dimension is 10.

Each EPSP parameter array occupies one column of an EPSP module. In each column, array elements [0], [1], [2], etc. are in module rows 1, 2, 3, etc. Module row 0 is not present in EPSP.

The parameter names, module columns, and types of the modifiable data are:
- $EPS_R[n] - Column 2 - Real
- $EPS_I[n] - Column 3 - Integer
- $EPS_B[n] - Column 4 - Boolean
- $EPS_A[n] - Column 5 - Axis
- $EPS_C[n] - Column 6 - Character
- $EPS_S[n] - Column 7 - String[32] (32 characters including terminator)

In addition, column 1 row 1 of an EPSP module contains a copy of the value of $MN_MM_EPSPARAM_DIMENSION. This value is read-only. When it is greater than zero, it is the largest row number that is present for the other columns. When it is equal to zero, none of the other columns is present, but this column 1 row 1 value is always present.

mmEpsParamDimension (C, EPSP)		$MM_EPSPARAM_DIMENSION		EPS
The machine data $MN_MM_EPSPARAM_DIMENSION Reserved for internal use only (ePS).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_WORD			
			Read access	

epsR (C, EPSP)		$EPS_R[x] x = ParameterNo		EPS
ePS parameter of type REAL Reserved for internal use only (ePS).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

epsR (C, EPSP)	$EPS_R[x] x = ParameterNo	EPS
	$MM_EPSPARAM_DIMENSION + 1	
	Read access and write access	

epsI (C, EPSP)		$EPS_I[x] x = ParameterNo		EPS
ePS parameter of type INT Reserved for internal use only (ePS).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$MM_EPSPARAM_DIMENSION + 1	
			Read access and write access	

epsB (C, EPSP)		$EPS_B[x] x = ParameterNo		EPS
ePS parameter of type BOOL Reserved for internal use only (ePS).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			$MM_EPSPARAM_DIMENSION + 1	
			Read access and write access	

epsA (C, EPSP)		$EPS_A[x] x = ParameterNo		EPS
ePS parameter of type AXIS Reserved for internal use only (ePS).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$MM_EPSPARAM_DIMENSION + 1	
			Read access and write access	

epsC (C, EPSP)		$EPS_C[x] x = ParameterNo		EPS
ePS parameter of type CHAR Reserved for internal use only (ePS).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			

epsC (C, EPSP)	$EPS_C[x] x = ParameterNo	EPS
Addressing		
Line index:	Max. line index:	
	$MM_EPSPARAM_DIMENSION + 1	
	Read access and write access	

epsS (C, EPSP)		$EPS_S[x] x = ParameterNo		EPS
ePS parameter of type STRING Reserved for internal use only (ePS).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MM_EPSPARAM_DIMENSION + 1	
			Read access and write access	

3.2.3.6 ETP: Event types

Description of logging event types.
It is permissible to access this module via several lines and columns.

The line index identifies a specific event.
Standard events: line index <= 10000:
OEM events: line index > 10000:
User index: is determined by the 1000s digit of the line index
Event type: is determined by the last three digits of the line index

Examples of the line index:
00001: Standard event of user 0 with the number 1 (IPO)
00006: Standard event of user 0 with the number 6 (NC start)
03006: Standard event of user 3 with the number 6 (NC start)
06006: Standard event of user 6 with the number 6 (NC start)
10001: OEM event of user 0 with the number 1
13002: OEM event of user 3 with the number 2

Standard event types:

Cyclic events:
1 = IPO and IPO cycle
15 = IPO2
47 = IPO3 (from SW $[[SW510400]])
48 = IPO4 (from SW $[[SW510400]])

Acyclic events related to axis motions:
2 = GEO_AXIS_START and Geo axis starts or changes the direction
18 = GEO_AXIS_STARTa see VDI interface NCK->PLC channel specific
DBB40 Bit6 and Bit7 (Bit6 = motion command+, Bit7 = motion command-)
Event occurs when a bit is reset.
3 = GEO_AXIS_STOP and Geo axis stops
19 = GEO_AXIS_STOPa, see VDI interface NCK->PLC channel specific
DBB40 Bit6 and Bit7 (Bit6 = motion command-, Bit7 = motion command+)
Event occurs when both bits are set to 0 and one of them was previously active.
4 = MA_AXIS_START, One machine axis of the channel starts or changes the direction
see VDI interface NCK->PLC axis-specific
DBB64 Bit6 and Bit7 (Bit6 = motion command-, Bit7 = motion command+)
Event occurs when a bit is reset.
5 = MA_AXIS_STOP, One machine axis stops
see VDI interface NCK->PLC axis-specific

DBB64 Bit6 and Bit7 (Bit6 = motion command-, Bit7 = motion command+)
Event occurs if both bits are set to 0 and one of them was previously active.

Acyclic events related to channel influence:

6 = NC_START NC start (if detected in NC)
7 = NC_STOP NC stop (if detected in NC, axes may still be traversed)
60 = CHAN_STATE The channel state changes (from SW $[[SW991806]])
61 = PROG_STATE The program state changes (from SW $[[SW991806]])

Acyclic events related to part program processing:

8 = BLOCK_BEG_1 and Block start (first IPO cycle of a block) without intermediate blocks, all program levels
52 = BLOCK_BEG_1a

9 = BLOCK_BEG_2 and Block start (first IPO cycle of a block) with intermediate blocks, all program levels
20 = BLOCK_BEG_2a
56 = BLOCK_BEG_2b
57 = BLOCK_BEG_2c
10 = BLOCK_BEG_3 Block start (first IPO cycle of a block) without intermediate blocks, only main program level and MDI level

16 = BLOCK_BEG_S1 and Block start (search run with computation) with intermediate blocks, all program levels
22 = BLOCK_BEG_S1a
11 = BLOCK_END_1 Block end (first IPO cycle of a block) without intermediate blocks, all program levels

12 = BLOCK_END_2 and Block end (first IPO cycle of a block) with intermediate blocks, all program levels
21 = BLOCK_END_2a
13 = BLOCK_END_3 Block end (first IPO cycle of a block) without intermediate blocks, only main program level and MDI level

17 = BLOCK_END_S1 Block end (search run with computation) with intermediate blocks, all program levels

31 = BLOCK_END_P1 Block end (run in) (from SW $[[SW530000]])
32 = BLOCK_END_P1a Block end (run in) (from SW $[[SW530000]])

44 = BLOCK_END_I1 Block end (interpreter) (from SW $[[SW510300]])

43 = NC_LEVEL_CHG Level change during part program processing (from SW $[[SW510300]])

Acyclic events triggered by part program command WRTPR

23 = PROT_TXT_REQ Logging a WRTPR text
24 = PROT_TXT_REQ_S1 Logging a WRTPR text (search run with computation)
33 = PROT_TXT_REQ_P1 Logging a WRTPR text (run in) (from SW $[[SW510300]])

Acyclic events triggered by the logging process itself

14 = PROT_FILE_BEG Start logging related to a log file.
29 = PROT_START_TRIG Start trigger has triggered (from SW $[[SW510300]])
30 = PROT_STOP_TRIG Stop trigger has triggered (from SW $[[SW510300]])
46 = PROT_START Start logging (from SW $[[SW510300]])
45 = PROT_STOP Stop logging (from SW $[[SW510300]])

Acyclic events triggered by buttons

42 = CANCEL_BUTTON The Cancel button was pressed (from SW $[[SW510300]])

Acyclic events triggered by alarms

41 = ALARM_REPORTED An alarm has occurred (from SW $[[SW510300]])

Acyclic events triggered by synchronized action

36 = SYNC_ACT_ACTIV Activating synchronized action (from SW $[[SW510300]])
37 = SYNC_ACT_DEACT Deactivating synchronized action (from SW $[[SW510300]])
38 = SYNC_ACT_FIRE Synchronized action triggers (from SW $[[SW510300]])

Acyclic events triggered by tool

25 = TOOL_CHANGE tool change (from SW $[[SW420000]])
27 = TOOL_CHANGE_S1 tool change (search run with computation) (from SW $[[SW440000]])
34 = TOOL_CHANGE_P1 tool change (run in) (from SW $[[SW510300]])
26 = CUTTEDGE_CHANGE cutting edge change (from SW $[[SW420000]])
28 = CUTTEDGE_CHANGE_S1 cutting edge change (search run with computation) (from SW $[[SW440000]])
35 = CUTTEDGE_CHANGE_P1 cutting edge change (run in) (from SW $[[SW510300]])

Acyclic events triggered by PLC

39 = PLC_OB_1 PLC OB1 started (from SW $[[SW510300]])
40 = PLC_OB40 PLC OB40 started (from SW $[[SW510300]])¶

asciiMode (C, ETP)				
Data logging format				
Return value	0: Data recorded in binary format with fixed alignment to 8 bytes 1: Data recorded in ASCII format 2: Data recorded in binary format with variable alignment 3: Data recorded in binary format with variable alignment and optimization of two consecutive data records of the same event. In this case, only the header is logged, not the actual data.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access and write access	

countActivated (C, ETP)				
Number of occurrences of event Number of times the event has occurred				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
			Read access	

countActivatedL (C, ETP)				
Number of times the event has occurred.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

dataListIndex (C, ETP)				
Index of data list Index of data list to be used				
Return value	All valid columns in module ETPD - 1)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	

dataListIndex (C, ETP)				
			siehe Bausteinkopf	
			Read access and write access	

dataProtok (C, ETP)				
Bytes written to the Fifo file Number of bytes entered in the Fifo file				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
			Read access	

dataUploaded (C, ETP)				
Bytes read to the Fifo file Number of bytes already uploaded from the Fifo file				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
			Read access	

eventActive (C, ETP)				
Event state				
Return value	0: Not active 1: Active 2: Deactivate and release data set			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access and write access	

eventActiveStatus (C, ETP)				
State For diagnosis: Event state				
Return value	0: Activated 1: Not activated 2: Cannot be activated because the sum of the variable lengths is too large 3: Cannot be activated because the internal resources are not sufficient 4: Cannot be activated because the protocol file cannot be created 100-...- cannot be activated because the variable specification with the index (value - 100) is wrong			

eventActiveStatus (C, ETP)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

headerType (C, ETP)				
Type of header in the data record				

Return value	0: No header
	1: Short header with the following structure:
	UDword dataStamp; // Data record identified by a consecutive number
	UWord event; // Enter type of event that occurred
	UWord protCount; // The number of times the event has been logged
	2: Long header with the following structure:
	UDword dataStamp; // Data record identified by a consecutive number
	UWord event; // Enter type of event that occurred
	UByte chan; // Channel in which the event occurred
	UByte dummy1; // Still free
	UDword protCount; // The number of times the event has been logged
	UDword dummy2; // Still free
	3: Mid-length header, non-aligned with the following structure:
	UDword dataStamp; // Data record identified by a consecutive number
	UWord event; // Enter type of event that occurred
	UByte chan; // Channel in which the event occurred
	UByte dummy1; // Still free
	UDword protCount; // The number of times the event has been logged			

Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	0	3
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access and write access	

maxElementsFastFifoUsed (C, ETP)				
Maximum Fifo buffer fill level reached For diagnosis: Maximum number of entries in the FIFO buffer				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	

maxElementsFastFifoUsed (C, ETP)				
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

maxFileLength (C, ETP)				
Maximum length of log file Maximum length of the log file. Values less than 1024 are interpreted as KB, larger values as bytes. The minimum permissible length is 3000.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access and write access	

maxGrossFileLengthUsed (C, ETP)				
Maximum log file size For diagnosis: Maximum gross size of log file				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

maxNetFileLengthTooSmall (C, ETP)				
Number of bytes log file undersized For diagnosis: Number of (net) bytes by which log file is undersized				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

numElementsFastFifoTooSmall (C, ETP)				
Number of entries Fifo buffer undersized For diagnosis: Number of entries by which the Fifo buffer is undersized				

numElementsFastFifoTooSmall (C, ETP)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

protocolFilename (C, ETP)				
Log file name Name of the log file including the path				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access and write access	

resultPar1 (C, ETP)				
General result value General result value, the significance is a function of the event. SYNC_ACT_ACTIVATE, SYNC_ACT_DEACTIVATE, and SYNC_ACT_FIRE: ID of the synchronous action. All non-stated events do not supply this result value.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

skip (C, ETP)				
Number of events to be skipped				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access and write access	

startTriggerLock (C, ETP)				
Processing block for the start trigger Setting, whether the start trigger is not to be processed during this event.				
Return value	0: Trigger is processed 1: Trigger is not processed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
			Read access and write access	

stopTriggerLock (C, ETP)				
Processing block for the stop trigger Setting, whether the stop trigger is not to be processed during this event.				
Return value	0: Trigger is processed 1: Trigger is not processed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
			Read access and write access	

suppressProtLock (C, ETP)				
$$traceProtocolLock is inactive Clears the effect of $$traceProtocolLock				
Return value	0: The disable is active 1: The disable is canceled for this event			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

timePeriod (C, ETP)				
Time base Time base for cyclic event only				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
ms	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

3.2.3.7 FA: Active Work Offset

The following frame indices are available:

0: $P_ACTFRAME = current resulting work offset
1: $P_IFRAME = current settable work offset
2: $P_PFRAME = current programmable work offset
3: EXTFRAME = current external work offset
4: TOTFRAME = current total work offset = sum of ACTFRAME and EXTFRAME
5: $P_ACTBFRAME = current total base frame
6: $P_SETFRAME = current 1st system frame (set actual value, scratching)
7: $P_EXTSFRAME = current 2nd system frame (set actual value, scratching)
8: $P_PARTFRAME = current 3rd system frame (TCARR and PAROT with orientable tool carrier)
9: $P_TOOLFRAME = current 4th system frame (TOROT and TOFRAME)
10: $AC_MEASFRAME = result frame for workpiece and tool measurement
11: $P_WPFRAME = current 5th system frame (workpiece reference points) as from SW $[[SW440000]]
12: $P_CYCFRAME = current 6th system frame (cycles) as from SW $[[SW440000]]
13: $P_TRAFRAME = current 7th system frame (transformation) as from SW $[[SW520000]]
14: $P_ISO1FRAME = current ISO system frame for G51.1 mirroring as from SW $[[SW660000]]
15: $P_ISO2FRAME = current ISO system frame for G68 2DROT as from SW $[[SW660000]]
16: $P_ISO3FRAME = current ISO system frame for G68 3DROT as from SW $[[SW660000]]
17: $P_ISO4FRAME = current ISO system frame for G51 scale as from SW $[[SW660000]]
18: $P_ACSFRAME = current resulting frame for the SZS (ACS) as from SW $[[SW660000]]
19: $P_RELFRAME = current 12th system frame for relative coordinate systems as from SW $[[SW700000]]
20: $P_TRAFRAME_P = current frame of the workpiece component of an active kinematic (orientation) transformation as from SW $[[SW900000]]
21: $P_TRAFRAME_T = current frame of the tool component of an active kinematic (orientation) transformation as from SW $[[SW900000]]

The maximum frame index is 21.

linShift (C, FA)		diverse, siehe Bausteinbescheibung		PA
Translation of an active frame Translation of an active work offset (the physical unit is defined in basicLengthUnit in module Y in area N).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			20 * $$numMachAxes	
			Read access	

linShiftFine (C, FA)		diverse, siehe Bausteinbescheibung		
Fine offset of an active frame				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			20 * $$numMachAxes	
			Read access and write access	

mirrorImgActive (C, FA)		diverse, siehe Bausteinbescheibung		PA
Mirroring enabled in an active frame Mirroring enabled in an active work offset				
Return value	0 = mirroring not active 1 = mirroring active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			20 * $$numMachAxes	
			Read access	

rotation (C, FA)		diverse, siehe Bausteinbescheibung		PA
Rotation of an active frame Rotation of an active work offset				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			20 * $$numMachAxes	
			Read access	

rotationCoordinate (C, FA)				
Coordinate rotation of active frames Rotation around a coordinate of an active work offset 1: Rotation around the first non-existing geometry axis.				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

rotationCoordinate (C, FA)		
	20 * $$numMachAxes	
	Read access	

scaleFact (C, FA)		diverse, siehe Bausteinbescheibung		PA
Scaling factor of an active frame Scaling factor of an active work offset				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			20 * $$numMachAxes	
			Read access	

3.2.3.8 FB: Base Frame

This only applies if $MC_MM_NUM_BASE_FRAMES > 0.

The maximum frame index is: $MC_MM_NUM_BASE_FRAMES - 1

linShift (C, FB)		$P_CHBFR[x,y,TR] x=FrameNo, y=Axis		PA
Translation of a settable frame Translation of settable work offset (the physical unit is defined in basicLengthUnit in module Y in area N).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_BASE_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
			Read access and write access	

linShiftFine (C, FB)		$P_CHBFR[x,y,SI] x=FrameNo, y=Axis		
Fine offset for frames Fine offset with frames, expansion of basic frames and settable frames				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

linShiftFine (C, FB)	$P_CHBFR[x,y,SI] x=FrameNo, y=Axis	
		$MC_MM_NUM_BASE_FRAMES * ($$num-GeoAxes + $$numAuxAxes)
		Read access and write access

mirrorImgActive (C, FB)		$P_CHBFR[x,y,MI] x=FrameNo, y=Axis		PA
Mirroring enabled in a settable frame Mirroring enabled in a settable work offset				
Return value	0: Mirroring not active 1: Mirroring active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_BASE_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
			Read access and write access	

rotation (C, FB)		$P_CHBFR[x,y,RT] x=FrameNo, y=Axis		PA
Rotation of a settable frame Rotation of a settable work offset				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_BASE_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
			Read access and write access	

rotationCoordinate (C, FB)				
Coordinate rotation of settable frames Rotation around coordinate of a channel base frame 1: Rotation around first non-existing geometry axis.				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_BASE_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
			Read access and write access	

scaleFact (C, FB)		$P_CHBFR[x,y,SC] x=FrameNo, y=Axis		PA
Scaling factor of a settable frame Scaling factor of a settable work offset				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_BASE_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
			Read access and write access	

3.2.3.9 FE: External Work Offset

There is exactly one external frame defined by the PLC.

The maximum frame index is: 0

linShift (C, FE)		$AA_ETRANS[x] x = FrameNo		PA
Translation of an external frame Translation of external work offset (the physical unit is defined in basicLengthUnit in module Y in area N).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numGeoAxes	
			Read access and write access	

linShiftFine (C, FE)		diverse, siehe Bausteinbescheibung		
Fine offset of external frame Fine offset of external work offset				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numGeoAxes	
			Read access and write access	

mirrorImgActive (C, FE)		diverse, siehe Bausteinbescheibung		PA
Mirroring of an external frame Mirroring of an external work offset				
Return value	0 = mirroring not active 1 = mirroring active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			20 * $$numMachAxes	
			Read access	

rotation (C, FE)		diverse, siehe Bausteinbescheibung		PA
Rotation of an external frame Rotation of an external work offset				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			20 * $$numMachAxes	
			Read access	

rotationCoordinate (C, FE)				
Coordinate rotation external work offset Rotation around coordinate of an external work offset 1: Rotation around first non-existing geometry axis.				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			20 * $$numMachAxes	
			Read access	

scaleFact (C, FE)		diverse, siehe Bausteinbescheibung		PA
Scaling factor of external frame Scaling factor of an external work offset				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

scaleFact (C, FE)	diverse, siehe Bausteinbescheibung	PA
Addressing		
Line index:	Max. line index:	
	20 * $$numMachAxes	
	Read access	

3.2.3.10 FG: Frames for grinding applications

These occur only if $MC_MM_NUM_G_FRAMES > 0 and $MN_MM_NUM_GLOBAL_G_FRAMES = 0, otherwise all grinding frames are configured globally for the NCU.

The following frame indices are possible:
0: GRAME1
1: GRAME2
2: GRAME3
3: GRAME4
...
n: GRAMEn
...
99: GRAME100

The maximum frame index is: $MC_MM_NUM_G_FRAMES - 1

The PI service SETUFR has to be called to activate the grinding frames.

linShift (C, FG)		$P_GFR[x,y,TR] x=FrameNo,y=Axis		PA
Translation of grinding frame Translation of grinding frame (the physical unit is defined in basicLengthUnit in block Y in area N).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_G_FRAMES * ($$numGeoAxes + $$numAuxAxes)	
			Read access and write access	

linShiftFine (C, FG)		$P_GFR[x,y,SI] x=FrameNo,y=Axis		
Fine offset for frames Fine offset with frames, expansion of basic frames and settable frames				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

linShiftFine (C, FG)	$P_GFR[x,y,SI] x=FrameNo,y=Axis	
Addressing		
Line index:	Max. line index:	
	$MC_MM_NUM_G_FRAMES * ($$numGeoAxes + $$numAuxAxes)	
	Read access and write access	

mirrorImgActive (C, FG)		$P_GFR[x,y,MI] x = FrameNo,y=Axis		PA
Mirroring of grinding frame Mirroring				
Return value	0 = mirroring not active 1 = mirroring active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:		Max. line index:		
		$MC_MM_NUM_G_FRAMES * ($$numGeoAxes + $$numAuxAxes)		
		Read access and write access		

rotation (C, FG)		$P_GFR[x,y,RT] x = FrameNo,y=Axis		PA
Rotation of grinding frame Rotation				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		$MC_MM_NUM_G_FRAMES * ($$numGeoAxes + $$numAuxAxes)		
		Read access and write access		

rotationCoordinate (C, FG)				
Rotation of grinding frame coordinate Rotation around a coordinate. 1: Rotation around the first non-existent geometry axis.				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		

rotationCoordinate (C, FG)				
			$MC_MM_NUM_G_FRAMES * ($$numGeoAxes + $$numAuxAxes)	
			Read access and write access	

scaleFact (C, FG)		$P_GFR[x,y,SC] x = FrameNo,y=Axis		PA
Scaling factor of grinding frame Scaling factor				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_G_FRAMES * ($$numGeoAxes + $$numAuxAxes)	
			Read access and write access	

3.2.3.11 FS: System Frame

Those that exist are set by the bits in $MC_MM_SYSTEM_FRAME_MASK.
Consequently, there may be gaps between the active system frames.

The maximum frame index is:
3 up to but excluding SW $[[SW440000]].
5 from and including SW $[[SW440000]].
11 as from and including SW $[[SW660000]].
12 as from and including SW $[[SW700000]].

linShift (C, FS)		$P_SETFR[Achse, TR]		
Translation				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			12 * ($$numGeoAxes+$$numAuxAxes)	
			Read access and write access	

linShiftFine (C, FS)	$P_SETFR[Achse, SI]	
Fine offset of a system frame Fine offset		

linShiftFine (C, FS)		$P_SETFR[Achse, SI]		
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			12 * ($$numGeoAxes+$$numAuxAxes)	
			Read access and write access	

mirrorImgActive (C, FS)		$P_SETFR[Achse, MI]		
Mirroring a system frame Mirroring				
Return value	0: Mirroring not active 1: Mirroring active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			12 * ($$numGeoAxes+$$numAuxAxes)	
			Read access and write access	

rotation (C, FS)		$P_SETFR[Achse, RT]		
Rotation of a system frame Rotation				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			12 * ($$numGeoAxes+$$numAuxAxes)	
			Read access and write access	

rotationCoordinate (C, FS)				
Coordinate rotation of a system frame Rotation around a coordinate of a system frame 1: Rotation around first non-existing geometry axis.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			12 * ($$numGeoAxes+$$numAuxAxes)	
			Read access and write access	

scaleFact (C, FS)		$P_SETFR[Achse, SC]		
Scaling factor of a system frame Scaling factor				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			12 * ($$numGeoAxes+$$numAuxAxes)	
			Read access and write access	

3.2.3.12 FU: Settable Work Offset

This only applies if $MC_MM_NUM_USER_FRAMES > 0 and $MN_MM_NUM_GLOBAL_USER_FRAMES = 0, otherwise all settable frames have an NCU-global configuration.

The following frame indices are possible:
0 = G500
1 = G54
2 = G55
3 = G56
4 = G57
5 = G505
6 = G506
:
n = G5n
:
99 = G599

The maximum frame index is: $MC_MM_NUM_USER_FRAMES - 1

The PI service SETUFR has to be called in order to activate the settable frames.

linShift (C, FU)		$P_UIFR[x,y,TR] x=FrameNo,y=Axis		PA
Translation of a settable frame Translation of settable work offset (the physical unit is defined in basicLengthUnit in module Y in area N).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

linShift (C, FU)	$P_UIFR[x,y,TR] x=FrameNo,y=Axis	PA
	$MC_MM_NUM_USER_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
	Read access and write access	

linShiftFine (C, FU)		$P_UIFR[x,y,SI] x=FrameNo,y=Axis		
Fine offset for frames Fine offset with frames, expansion of basic frames and settable frames				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_USER_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
			Read access and write access	

mirrorImgActive (C, FU)		$P_UIFR[x,y,MI] x = FrameNo,y=Axis		PA
Mirroring enabled in a settable frame Mirroring enabled in a settable work offset				
Return value	0 = mirroring not active 1 = mirroring active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_USER_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
			Read access and write access	

rotation (C, FU)		$P_UIFR[x,y,RT] x = FrameNo,y=Axis		PA
Rotation of a settable frame Rotation of a settable work offset				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_USER_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
			Read access and write access	

rotationCoordinate (C, FU)				
Coordinate rotation of settable frames Rotation around a coordinate of a settable work offset 1: Rotation around first non-existing geometry axis.				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_USER_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
			Read access and write access	

scaleFact (C, FU)		$P_UIFR[x,y,SC] x = FrameNo,y=Axis		PA
Scaling factor of a settable frame Scaling factor of a settable work offset				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_USER_FRAMES * ($$num-GeoAxes + $$numAuxAxes)	
			Read access and write access	

3.2.3.13 M: Machine data

Channel-specific machine data

ADD_MOVE_ACCEL_RESERVE (C, M)		MD 20610: $MC_ADD_MOVE_AC-CEL_RESERVE		
MD 20610: $MC_ADD_MOVE_ACCEL_RESERVE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

ALLOW_G0_IN_G96 (C, M)		MD 20750: $MC_ALLOW_G0_IN_G96		
MD 20750: $MC_ALLOW_G0_IN_G96				
Units and value ranges				
Physical unit:	Data type:			

ALLOW_G0_IN_G96 (C, M)		MD 20750: $MC_ALLOW_G0_IN_G96		
-	TYPE_BOOLEAN			
			Read access	

AUXFU_ASSIGN_EXTENSION (C, M)		MD 22020: $MC_AUXFU_ASSIGN_EXTENSION		
MD 22020: $MC_AUXFU_ASSIGN_EXTENSION				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

AUXFU_ASSIGN_GROUP (C, M)		MD 22000: $MC_AUXFU_ASSIGN_GROUP		
MD 22000: $MC_AUXFU_ASSIGN_GROUP				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

AUXFU_ASSIGN_TYPE (C, M)		MD 22010: $MC_AUXFU_ASSIGN_TYPE		
MD 22010: $MC_AUXFU_ASSIGN_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

AUXFU_ASSIGN_VALUE (C, M)		MD 22030: $MC_AUXFU_ASSIGN_VALUE		
MD 22030: $MC_AUXFU_ASSIGN_VALUE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	

AUXFU_ASSIGN_VALUE (C, M)	MD 22030: $MC_AUXFU_ASSIGN_VALUE	
	2	
	Read access	

AUXFU_AT_BLOCK_SEARCH_END (C, M)		MD 22300: $MC_AUXFU_AT_BLOCK_SEARCH_END		
MD 22300: $MC_AUXFU_AT_BLOCK_SEARCH_END				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

AUXFU_D_SYNC_TYPE (C, M)		MD 22250: $MC_AUXFU_D_SYNC_TYPE		
MD 22250: $MC_AUXFU_D_SYNC_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

AUXFU_E_SYNC_TYPE (C, M)		MD 22260: $MC_AUXFU_E_SYNC_TYPE		
MD 22260: $MC_AUXFU_E_SYNC_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

AUXFU_F_SYNC_TYPE (C, M)		MD 22240: $MC_AUXFU_F_SYNC_TYPE		
MD 22240: $MC_AUXFU_F_SYNC_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

AUXFU_H_SYNC_TYPE (C, M)		MD 22230: $MC_AUXFU_H_SYNC_TYPE		
MD 22230: $MC_AUXFU_H_SYNC_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

AUXFU_M_SYNC_TYPE (C, M)		MD 22200: $MC_AUXFU_M_SYNC_TYPE		
MD 22200: $MC_AUXFU_M_SYNC_TYPE				
Units and value ranges				
Physical unit:	Data type:			

AUXFU_M_SYNC_TYPE (C, M)		MD 22200: $MC_AUXFU_M_SYNC_TYPE		
-	TYPE_CHAR			
			Read access	

AUXFU_S_SYNC_TYPE (C, M)		MD 22210: $MC_AUXFU_S_SYNC_TYPE		
MD 22210: $MC_AUXFU_S_SYNC_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

AUXFU_T_SYNC_TYPE (C, M)		MD 22220: $MC_AUXFU_T_SYNC_TYPE		
MD 22220: $MC_AUXFU_T_SYNC_TYPE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

AXCONF_CHANAX_NAME_TAB (C, M)		MD 20080: $MC_AXCONF_CHANAX_NAME_TAB		
MD 20080: $MC_AXCONF_CHANAX_NAME_TAB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

AXCONF_GEOAX_ASSIGN_TAB (C, M)		MD 20050: $MC_AXCONF_GEOAX_ASSIGN_TAB		
MD 20050: $MC_AXCONF_GEOAX_ASSIGN_TAB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

AXCONF_GEOAX_NAME_TAB (C, M)		MD 20060: $MC_AXCONF_GEOAX_NAME_TAB		
MD 20060: $MC_AXCONF_GEOAX_NAME_TAB				
Units and value ranges				
Physical unit:	Data type:			

AXCONF_GEOAX_NAME_TAB (C, M)		MD 20060: $MC_AXCONF_GEO-AX_NAME_TAB		
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

AXCONF_MACHAX_USED (C, M)		MD 20070: $MC_AXCONF_MA-CHAX_USED		
MD 20070: $MC_AXCONF_MACHAX_USED				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

CIRCLE_ERROR_CONST (C, M)		MD 21000: $MC_CIRCLE_ER-ROR_CONST		
MD 21000: $MC_CIRCLE_ERROR_CONST				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CIRCLE_ERROR_FACTOR (C, M)		MD 21010: $MC_CIRCLE_ERROR_FAC-TOR		
MD 21010: $MC_CIRCLE_ERROR_FACTOR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

COLLECT_TOOL_CHANGE (C, M)		MD 20128: $MC_COL-LECT_TOOL_CHANGE		
MD 20128: $MC_COLLECT_TOOL_CHANGE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

COMPRESS_BLOCK_PATH_LIMIT (C, M)		MD 20170: $MC_COMPRESS_BLOCK_PATH_LIMIT		
MD 20170: $MC_COMPRESS_BLOCK_PATH_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

COMPRESS_VELO_TOL (C, M)		MD 20172: $MC_COMPRESS_VELO_TOL		
MD 20172: $MC_COMPRESS_VELO_TOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CONST_VELO_MIN_TIME (C, M)		MD 20500: $MC_CONST_VELO_MIN_TIME		
MD 20500: $MC_CONST_VELO_MIN_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

CONTOUR_ASSIGN_FASTOUT (C, M)		MD 21070: $MC_CONTOUR_ASSIGN_FASTOUT		
MD 21070: $MC_CONTOUR_ASSIGN_FASTOUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

CONTOUR_TUNNEL_REACTION (C, M)		MD 21060: $MC_CONTOUR_TUNNEL_REACTION		
MD 21060: $MC_CONTOUR_TUNNEL_REACTION				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

CONTOUR_TUNNEL_TOL (C, M)		MD 21050: $MC_CONTOUR_TUNNEL_TOL		
MD 21050: $MC_CONTOUR_TUNNEL_TOL				
Units and value ranges				
Physical unit:	Data type:			

CONTOUR_TUNNEL_TOL (C, M)		MD 21050: $MC_CONTOUR_TUNNEL_TOL		
-	TYPE_DOUBLE			
			Read access	

COUPLE_AXIS_1 (C, M)		MD 21300: $MC_COUPLE_AXIS_1		
MD 21300: $MC_COUPLE_AXIS_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

COUPLE_BLOCK_CHANGE_CTRL_1 (C, M)		MD 21320: $MC_COUPLE_BLOCK_CHANGE_CTRL_1		
MD 21320: $MC_COUPLE_BLOCK_CHANGE_CTRL_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

COUPLE_IS_WRITE_PROT_1 (C, M)		MD 21340: $MC_COUPLE_IS_WRITE_PROT_1		
MD 21340: $MC_COUPLE_IS_WRITE_PROT_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

COUPLE_RESET_MODE_1 (C, M)		MD 21330: $MC_COUPLE_RESET_MODE_1		
MD 21330: $MC_COUPLE_RESET_MODE_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

COUPLING_MODE_1 (C, M)		MD 21310: $MC_COUPLING_MODE_1		
MD 21310: $MC_COUPLING_MODE_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

CPREC_WITH_FFW (C, M)		MD 20470: $MC_CPREC_WITH_FFW		
MD 20470: $MC_CPREC_WITH_FFW				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

CUBIC_SPLINE_BLOCKS (C, M)		MD 20160: $MC_CUBIC_SPLINE_BLOCKS		
MD 20160: $MC_CUBIC_SPLINE_BLOCKS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

CUTCOM_CORNER_LIMIT (C, M)		MD 20210: $MC_CUTCOM_CORNER_LIMIT		
MD 20210: $MC_CUTCOM_CORNER_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CUTCOM_CURVE_INSERT_LIMIT (C, M)		MD 20230: $MC_CUTCOM_CURVE_INSERT_LIMIT		
MD 20230: $MC_CUTCOM_CURVE_INSERT_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CUTCOM_INTERS_POLY_ENABLE (C, M)		MD 20256: $MC_CUTCOM_INTERS_POLY_ENABLE		
MD 20256: $MC_CUTCOM_INTERS_POLY_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

CUTCOM_MAXNUM_CHECK_BLOCKS (C, M)		MD 20240: $MC_CUTCOM_MAXNUM_CHECK_BLOCKS		
MD 20240: $MC_CUTCOM_MAXNUM_CHECK_BLOCKS				
Units and value ranges				
Physical unit:	Data type:			

CUTCOM_MAXNUM_CHECK_BLOCKS (C, M)		MD 20240: $MC_CUTCOM_MAX-NUM_CHECK_BLOCKS		
-	TYPE_CHAR			
			Read access	

CUTCOM_MAXNUM_SUPPR_BLOCKS (C, M)		MD 20252: $MC_CUTCOM_MAX-NUM_SUPPR_BLOCKS		
MD 20252: $MC_CUTCOM_MAXNUM_SUPPR_BLOCKS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

CUTCOM_MAX_DISC (C, M)		MD 20220: $MC_CUTCOM_MAX_DISC		
MD 20220: $MC_CUTCOM_MAX_DISC				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CUTCOM_PARALLEL_ORI_LIMIT (C, M)		MD 21080: $MC_CUTCOM_PARAL-LEL_ORI_LIMIT		
MD 21080: $MC_CUTCOM_PARALLEL_ORI_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CUTCOM_PLANE_ORI_LIMIT (C, M)		MD 21082: $MC_CUT-COM_PLANE_ORI_LIMIT		
MD 21082: $MC_CUTCOM_PLANE_ORI_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CUTCOM_PLANE_PATH_LIMIT (C, M)		MD 21084: $MC_CUT-COM_PLANE_PATH_LIMIT		
MD 21084: $MC_CUTCOM_PLANE_PATH_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CUTTING_EDGE_DEFAULT (C, M)		MD 20270: $MC_CUTTING_EDGE_DEFAULT		
MD 20270: $MC_CUTTING_EDGE_DEFAULT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

CUTTING_EDGE_RESET_VALUE (C, M)		MD 20130: $MC_CUTTING_EDGE_RESET_VALUE		
Cutting edge length offset during run-up Specifies the selection of the cutting edge (D number) for the tool length offset. The following dependencies exist: o With run-up, reset or part program end via the setting of MD20110 $MC_RESET_MODE_MASK o With part program start via MD20112 $MC_START_MODE_MASK The last tool offset, usually the tool on the spindle, remains active if bits 0 and 6 are set in MD20110 $MC_RESET_MODE_MASK.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

DIAMETER_AX_DEF (C, M)		MD 20100: $MC_DIAMETER_AX_DEF		
MD 20100: $MC_DIAMETER_AX_DEF				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

DIG_A_MAX (C, M)		MD 21460: $MC_DIG_A_MAX		
MD 21460: $MC_DIG_A_MAX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_A_MAX_MOVE (C, M)		MD 21462: $MC_DIG_A_MAX_MOVE		
MD 21462: $MC_DIG_A_MAX_MOVE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_A_MAX_SCAN (C, M)		MD 21464: $MC_DIG_A_MAX_SCAN		
MD 21464: $MC_DIG_A_MAX_SCAN				

DIG_A_MAX_SCAN (C, M)		MD 21464: $MC_DIG_A_MAX_SCAN		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_DELTAPOS (C, M)		MD 21440: $MC_DIG_DELTAPOS		
MD 21440: $MC_DIG_DELTAPOS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_L_INKR (C, M)		MD 21424: $MC_DIG_L_INKR		
MD 21424: $MC_DIG_L_INKR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

DIG_L_MIN (C, M)		MD 21430: $MC_DIG_L_MIN		
MD 21430: $MC_DIG_L_MIN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_L_NORMAL (C, M)		MD 21432: $MC_DIG_L_NORMAL		
MD 21432: $MC_DIG_L_NORMAL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_L_NOTAUS (C, M)		MD 21434: $MC_DIG_L_NOTAUS		
MD 21434: $MC_DIG_L_NOTAUS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_L_NOTAUS_EXT (C, M)		MD 21436: $MC_DIG_L_NOTAUS_EXT		
MD 21436: $MC_DIG_L_NOTAUS_EXT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_L_OFF_Z (C, M)		MD 21422: $MC_DIG_L_OFF_Z		
MD 21422: $MC_DIG_L_OFF_Z				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_L_ORDER (C, M)		MD 21420: $MC_DIG_L_ORDER		
MD 21420: $MC_DIG_L_ORDER				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

DIG_PROT_VERSION (C, M)		MD 21400: $MC_DIG_PROT_VERSION		
MD 21400: $MC_DIG_PROT_VERSION				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

DIG_P_HAND (C, M)		MD 21470: $MC_DIG_P_HAND		
MD 21470: $MC_DIG_P_HAND				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_P_SCAN (C, M)		MD 21472: $MC_DIG_P_SCAN		
MD 21472: $MC_DIG_P_SCAN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_T_SCAN (C, M)		MD 21474: $MC_DIG_T_SCAN		
MD 21474: $MC_DIG_T_SCAN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

DIG_V_EILGANG (C, M)		MD 21450: $MC_DIG_V_EILGANG		
MD 21450: $MC_DIG_V_EILGANG				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

F_VALUES_ACTIVE_AFTER_RESET (C, M)		MD 22410: $MC_F_VALUES_ACTIVE_AFTER_RESET		
MD 22410: $MC_F_VALUES_ACTIVE_AFTER_RESET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

GCODE_GROUPS_TO_PLC (C, M)		MD 22510: $MC_GCODE_GROUPS_TO_PLC		
MD 22510: $MC_GCODE_GROUPS_TO_PLC				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

GCODE_RESET_VALUES (C, M)		MD 20150: $MC_GCODE_RESET_VALUES		
MD 20150: $MC_GCODE_RESET_VALUES				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

GEOAX_CHANGE_M_CODE (C, M)		MD 22532: $MC_GEOAX_CHANGE_M_CODE		
MD 22532: $MC_GEOAX_CHANGE_M_CODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

GEOAX_CHANGE_RESET (C, M)		MD 20118: $MC_GEOAX_CHANGE_RESET		
MD 20118: $MC_GEOAX_CHANGE_RESET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

HANDWH_CHAN_STOP_COND (C, M)		MD 20624: $MC_HANDWH_CHAN_STOP_COND		
MD 20624: $MC_HANDWH_CHAN_STOP_COND				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

HANDWH_GEOAX_MAX_INCR_SIZE (C, M)		MD 20620: $MC_HANDWH_GEOAX_MAX_INCR_SIZE		
MD 20620: $MC_HANDWH_GEOAX_MAX_INCR_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

HANDWH_GEOAX_MAX_INCR_VSIZE (C, M)		MD 20622: $MC_HANDWH_GEOAX_MAX_INCR_VSIZE		
MD 20622: $MC_HANDWH_GEOAX_MAX_INCR_VSIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

IGNORE_INHIBIT_ASUP (C, M)		MD 20116: $MC_IGNORE_INHIBIT_ASUP		
MD 20116: $MC_IGNORE_INHIBIT_ASUP				
Units and value ranges				
Physical unit:	Data type:			

IGNORE_INHIBIT_ASUP (C, M)		MD 20116: $MC_IGNORE_INHIBIT_ASUP		
-	TYPE_DWORD			
			Read access	

IGNORE_SINGLEBLOCK_ASUP (C, M)		MD 20117: $MC_IGNORE_SINGLE-BLOCK_ASUP		
MD 20117: $MC_IGNORE_SINGLEBLOCK_ASUP				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

LEN_AC_FIFO (C, M)		MD 28264: $MC_LEN_AC_FIFO		
MD 28264: $MC_LEN_AC_FIFO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

LIFTFAST_DIST (C, M)		MD 21200: $MC_LIFTFAST_DIST		
MD 21200: $MC_LIFTFAST_DIST				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

LOOKAH_NUM_OVR_POINTS (C, M)		MD 20430: $MC_LOO-KAH_NUM_OVR_POINTS		
MD 20430: $MC_LOOKAH_NUM_OVR_POINTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

LOOKAH_OVR_POINTS (C, M)		MD 20440: $MC_LOOKAH_OVR_POINTS		
MD 20440: $MC_LOOKAH_OVR_POINTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

LOOKAH_RELIEVE_BLOCK_CYCLE (C, M)		MD 20450: $MC_LOOKAH_RELIEVE_BLOCK_CYCLE		
MD 20450: $MC_LOOKAH_RELIEVE_BLOCK_CYCLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

LOOKAH_USE_VELO_NEXT_BLOCK (C, M)		MD 20400: $MC_LOOKAH_USE_VELO_NEXT_BLOCK		
MD 20400: $MC_LOOKAH_USE_VELO_NEXT_BLOCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

MAX_LEAD_ANGLE (C, M)		MD 21090: $MC_MAX_LEAD_ANGLE		
MD 21090: $MC_MAX_LEAD_ANGLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

MAX_PATH_JERK (C, M)		MD 20600: $MC_MAX_PATH_JERK		
MD 20600: $MC_MAX_PATH_JERK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

MAX_TILT_ANGLE (C, M)		MD 21092: $MC_MAX_TILT_ANGLE		
MD 21092: $MC_MAX_TILT_ANGLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

MDC_TECHNOLOGY_MODE (C, M)		MD 27800: $MC_MDC_TECHNOLOGY_MODE		
Alternative name:	TECHNOLOGY_MODE			
MD 27800: $MC_MDC_TECHNOLOGY_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			

MDC_TECHNOLOGY_MODE (C, M)	MD 27800: $MC_MDC_TECHNOLOGY_MODE	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

MDS_CHAN_NAME (C, M)		MD 20000: $MC_MDS_CHAN_NAME		
Alternative name:	CHAN_NAME			
MD 20000: $MC_MDS_CHAN_NAME				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

MM_IPO_BUFFER_SIZE (C, M)		MD 28060: $MC_MM_IPO_BUFFER_SIZE		
MD 28060: $MC_MM_IPO_BUFFER_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_IPO_TASK_STACK_SIZE (C, M)		MD 28510: $MC_MM_IPO_TASK_STACK_SIZE		
MD 28510: $MC_MM_IPO_TASK_STACK_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_LINK_TOA_UNIT (C, M)		MD 28085: $MC_MM_LINK_TOA_UNIT		
MD 28085: $MC_MM_LINK_TOA_UNIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_LUD_VALUES_MEM (C, M)		MD 28040: $MC_MM_LUD_VALUES_MEM		
MD 28040: $MC_MM_LUD_VALUES_MEM				
Units and value ranges				
Physical unit:	Data type:			

MM_LUD_VALUES_MEM (C, M)		MD 28040: $MC_MM_LUD_VALUES_MEM		
-	TYPE_DWORD			
			Read access	

MM_MAX_TRACE_DATAPOINTS (C, M)		MD 28180: $MC_MM_MAX_TRACE_DATAPOINTS		
MD 28180: $MC_MM_MAX_TRACE_DATAPOINTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_AC_MARKER (C, M)		MD 28256: $MC_MM_NUM_AC_MARKER		
MD 28256: $MC_MM_NUM_AC_MARKER				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_AC_PARAM (C, M)		MD 28254: $MC_MM_NUM_AC_PARAM		
MD 28254: $MC_MM_NUM_AC_PARAM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_AC_TIMER (C, M)		MD 28258: $MC_MM_NUM_AC_TIMER		
MD 28258: $MC_MM_NUM_AC_TIMER				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_BLOCKS_IN_PREP (C, M)		MD 28070: $MC_MM_NUM_BLOCKS_IN_PREP		
MD 28070: $MC_MM_NUM_BLOCKS_IN_PREP				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_CC_BLOCK_ELEMENTS (C, M)		MD 28090: $MC_MM_NUM_CC_BLOCK_ELEMENTS		
MD 28090: $MC_MM_NUM_CC_BLOCK_ELEMENTS				

MM_NUM_CC_BLOCK_ELEMENTS (C, M)		MD 28090: $MC_MM_NUM_CC_BLOCK_ELEMENTS		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_CC_BLOCK_USER_MEM (C, M)		MD 28100: $MC_MM_NUM_CC_BLOCK_USER_MEM		
MD 28100: $MC_MM_NUM_CC_BLOCK_USER_MEM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_FCTDEF_ELEMENTS (C, M)		MD 28252: $MC_MM_NUM_FCTDEF_EL-EMENTS		
MD 28252: $MC_MM_NUM_FCTDEF_ELEMENTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_LUD_NAMES_TOTAL (C, M)		MD 28020: $MC_MM_NUM_LUD_NAMES_TOTAL		
MD 28020: $MC_MM_NUM_LUD_NAMES_TOTAL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_PROTECT_AREA_ACTIVE (C, M)		MD 28210: $MC_MM_NUM_PRO-TECT_AREA_ACTIVE		
MD 28210: $MC_MM_NUM_PROTECT_AREA_ACTIVE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_PROTECT_AREA_CHAN (C, M)		MD 28200: $MC_MM_NUM_PRO-TECT_AREA_CHAN		
MD 28200: $MC_MM_NUM_PROTECT_AREA_CHAN				
Units and value ranges				
Physical unit:	Data type:			

MM_NUM_PROTECT_AREA_CHAN (C, M)		MD 28200: $MC_MM_NUM_PROTECT_AREA_CHAN		
-	TYPE_DWORD			
			Read access	

MM_NUM_REORG_LUD_MODULES (C, M)		MD 28010: $MC_MM_NUM_REORG_LUD_MODULES		
MD 28010: $MC_MM_NUM_REORG_LUD_MODULES				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_R_PARAM (C, M)		MD 28050: $MC_MM_NUM_R_PARAM		
MD 28050: $MC_MM_NUM_R_PARAM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_SYNC_ELEMENTS (C, M)		MD 28250: $MC_MM_NUM_SYNC_ELEMENTS		
MD 28250: $MC_MM_NUM_SYNC_ELEMENTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_USER_FRAMES (C, M)		MD 28080: $MC_MM_NUM_USER_FRAMES		
MD 28080: $MC_MM_NUM_USER_FRAMES				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_VDIVAR_ELEMENTS (C, M)		MD 28150: $MC_MM_NUM_VDIVAR_ELEMENTS		
MD 28150: $MC_MM_NUM_VDIVAR_ELEMENTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_PREP_TASK_STACK_SIZE (C, M)		MD 28500: $MC_MM_PREP_TASK_STACK_SIZE		
MD 28500: $MC_MM_PREP_TASK_STACK_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_REORG_LOG_FILE_MEM (C, M)		MD 28000: $MC_MM_RE-ORG_LOG_FILE_MEM		
MD 28000: $MC_MM_REORG_LOG_FILE_MEM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_TRACE_DATA_FUNCTION (C, M)		MD 22714: $MC_MM_TRACE_DA-TA_FUNCTION		
MD 22714: $MC_MM_TRACE_DATA_FUNCTION				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MODESWITCH_MASK (C, M)		MD 20114: $MC_MODESWITCH_MASK		
MD 20114: $MC_MODESWITCH_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MODE_AC_FIFO (C, M)		MD 28266: $MC_MODE_AC_FIFO		
MD 28266: $MC_MODE_AC_FIFO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

MULTFEED_ASSIGN_FASTIN (C, M)		MD 21220: $MC_MULTFEED_AS-SIGN_FASTIN		
MD 21220: $MC_MULTFEED_ASSIGN_FASTIN				
Units and value ranges				
Physical unit:	Data type:			

MULTFEED_ASSIGN_FASTIN (C, M)		MD 21220: $MC_MULTFEED_ASSIGN_FASTIN		
-	TYPE_DWORD			
			Read access	

MULTFEED_STORE_MASK (C, M)		MD 21230: $MC_MULTFEED_STORE_MASK		
MD 21230: $MC_MULTFEED_STORE_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

NIBBLE_PRE_START_TIME (C, M)		MD 26018: $MC_NIBBLE_PRE_START_TIME		
MD 26018: $MC_NIBBLE_PRE_START_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

NIBBLE_PUNCH_CODE (C, M)		MD 26008: $MC_NIBBLE_PUNCH_CODE		
MD 26008: $MC_NIBBLE_PUNCH_CODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

NIBBLE_PUNCH_INMASK (C, M)		MD 26006: $MC_NIBBLE_PUNCH_INMASK		
MD 26006: $MC_NIBBLE_PUNCH_INMASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

NIBBLE_PUNCH_OUTMASK (C, M)		MD 26004: $MC_NIBBLE_PUNCH_OUT-MASK		
Mask for fast output bits The variable defines a signal that triggers the stroke of the stamp in the punching process. o Bit[0] defines the output bit for the "Trigger stroke" signal				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

NIBBLE_SIGNAL_CHECK (C, M)		MD 26020: $MC_NIBBLE_SIG-NAL_CHECK		
MD 26020: $MC_NIBBLE_SIGNAL_CHECK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

NUM_AC_FIFO (C, M)		MD 28260: $MC_NUM_AC_FIFO		
MD 28260: $MC_NUM_AC_FIFO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

ONLINE_CUTCOM_ENABLE (C, M)		MD 20254: $MC_ONLINE_CUTCOM_EN-ABLE		
MD 20254: $MC_ONLINE_CUTCOM_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

ORIENTATION_IS_EULER (C, M)		MD 21100: $MC_ORIENTATION_IS_EU-LER		
MD 21100: $MC_ORIENTATION_IS_EULER				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

PATH_IPO_IS_ON_TCP (C, M)		MD 20260: $MC_PATH_IPO_IS_ON_TCP		
MD 20260: $MC_PATH_IPO_IS_ON_TCP				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

PUNCHNIB_ACTIVATION (C, M)		MD 26012: $MC_PUNCHNIB_ACTIVA-TION		
MD 26012: $MC_PUNCHNIB_ACTIVATION				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

PUNCHNIB_ASSIGN_FASTIN (C, M)		MD 26000: $MC_PUNCHNIB_AS-SIGN_FASTIN		
MD 26000: $MC_PUNCHNIB_ASSIGN_FASTIN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

PUNCHNIB_ASSIGN_FASTOUT (C, M)		MD 26002: $MC_PUNCHNIB_AS-SIGN_FASTOUT		
MD 26002: $MC_PUNCHNIB_ASSIGN_FASTOUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

PUNCHNIB_AXIS_MASK (C, M)		MD 26010: $MC_PUNCHNIB_AX-IS_MASK		
MD 26010: $MC_PUNCHNIB_AXIS_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

PUNCH_PARTITION_TYPE (C, M)		MD 26016: $MC_PUNCH_PARTI-TION_TYPE		
MD 26016: $MC_PUNCH_PARTITION_TYPE				
Units and value ranges				
Physical unit:	Data type:			

PUNCH_PARTITION_TYPE (C, M)		MD 26016: $MC_PUNCH_PARTI-TION_TYPE		
-	TYPE_DWORD			
			Read access	

PUNCH_PATH_SPLITTING (C, M)		MD 26014: $MC_PUNCH_PATH_SPLIT-TING		
MD 26014: $MC_PUNCH_PATH_SPLITTING				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

REFP_NC_START_LOCK (C, M)		MD 20700: $MC_REFP_NC_START_LOCK		
MD 20700: $MC_REFP_NC_START_LOCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

REORG_LOG_LIMIT (C, M)		MD 27900: $MC_REORG_LOG_LIMIT		
MD 27900: $MC_REORG_LOG_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

RESET_MODE_MASK (C, M)		MD 20110: $MC_RESET_MODE_MASK		
Spec. init. setting of control after Reset/PP end Specifies the initial setting of the control after run-up and reset/part program end with regard to the G codes (in particular current plane and settable work offset), tool length offset and transformation by setting the following bits.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SETINT_ASSIGN_FASTIN (C, M)		MD 21210: $MC_SETINT_ASSIGN_FAS-TIN		
MD 21210: $MC_SETINT_ASSIGN_FASTIN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SPF_END_TO_VDI (C, M)		MD 20800: $MC_SPF_END_TO_VDI		
MD 20800: $MC_SPF_END_TO_VDI				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

SPIND_ASSIGN_TAB_ENABLE (C, M)		MD 20092: $MC_SPIND_AS-SIGN_TAB_ENABLE		
MD 20092: $MC_SPIND_ASSIGN_TAB_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SPIND_DEF_MASTER_SPIND (C, M)		MD 20090: $MC_SPIND_DEF_MAS-TER_SPIND		
MD 20090: $MC_SPIND_DEF_MASTER_SPIND				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SPLINE_FEED_PRECISION (C, M)		MD 20262: $MC_SPLINE_FEED_PRECI-SION		
MD 20262: $MC_SPLINE_FEED_PRECISION				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

START_AC_FIFO (C, M)		MD 28262: $MC_START_AC_FIFO		
MD 28262: $MC_START_AC_FIFO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

START_MODE_MASK (C, M)		MD 20112: $MC_START_MODE_MASK		
MD 20112: $MC_START_MODE_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

S_VALUES_ACTIVE_AFTER_RESET (C, M)		MD 22400: $MC_S_VALUES_ACTIVE_AFTER_RESET		
MD 22400: $MC_S_VALUES_ACTIVE_AFTER_RESET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

THREAD_AUTO_LIFTFASTANGLE (C, M)		MD 20660: $MC_THREAD_AUTO_LIFTFASTANGLE		
MD 20660: $MC_THREAD_AUTO_LIFTFASTANGLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

THREAD_START_IS_HARD (C, M)		MD 20650: $MC_THREAD_START_IS_HARD		
MD 20650: $MC_THREAD_START_IS_HARD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

TOCARR_CHANGE_M_CODE (C, M)		MD 22530: $MC_TOCARR_CHANGE_M_CODE		
MD 22530: $MC_TOCARR_CHANGE_M_CODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TOOL_CARRIER_RESET_VALUE (C, M)		MD 20126: $MC_TOOL_CARRIER_RESET_VALUE		
MD 20126: $MC_TOOL_CARRIER_RESET_VALUE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TOOL_CHANGE_MODE (C, M)		MD 22550: $MC_TOOL_CHANGE_MODE		
MD 22550: $MC_TOOL_CHANGE_MODE				
Units and value ranges				
Physical unit:	Data type:			

TOOL_CHANGE_MODE (C, M)		MD 22550: $MC_TOOL_CHANGE_MODE		
-	TYPE_CHAR			
			Read access	

TOOL_CHANGE_M_CODE (C, M)		MD 22560: $MC_TOOL_CHANGE_M_CODE		
MD 22560: $MC_TOOL_CHANGE_M_CODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TOOL_GRIND_AUTO_TMON (C, M)		MD 20350: $MC_TOOL_GRIND_AUTO_TMON		
MD 20350: $MC_TOOL_GRIND_AUTO_TMON				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

TOOL_MANAGEMENT_MASK (C, M)		MD 20310: $MC_TOOL_MANAGEMENT_MASK		
MD 20310: $MC_TOOL_MANAGEMENT_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TOOL_MANAGEMENT_TOOLHOLDER (C, M)		MD 20124: $MC_TOOL_MANAGEMENT_TOOLHOLDER		
MD 20124: $MC_TOOL_MANAGEMENT_TOOLHOLDER				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TOOL_PARAMETER_DEF_MASK (C, M)		MD 20360: $MC_TOOL_PARAMETER_DEF_MASK		
MD 20360: $MC_TOOL_PARAMETER_DEF_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TOOL_PRESEL_RESET_VALUE (C, M)		MD 20121: $MC_TOOL_PRESEL_RESET_VALUE		
MD 20121: $MC_TOOL_PRESEL_RESET_VALUE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TOOL_RESET_NAME (C, M)		MD 20122: $MC_TOOL_RESET_NAME		
MD 20122: $MC_TOOL_RESET_NAME				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

TOOL_RESET_VALUE (C, M)		MD 20120: $MC_TOOL_RESET_VALUE		
MD 20120: $MC_TOOL_RESET_VALUE Specifies the selection of the tool (T number) for the tool length offset. The following dependencies exist: o With run-up, reset or part program end via the setting of MD20110 $MC_RESET_MODE_MASK o With part program start via MD20112 $MC_START_MODE_MASK.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TOOL_TIME_MONITOR_MASK (C, M)		MD 20320: $MC_TOOL_TIME_MONITOR_MASK		
MD 20320: $MC_TOOL_TIME_MONITOR_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TRAANG_ANGLE_1 (C, M)		MD 24700: $MC_TRAANG_ANGLE_1		
MD 24700: $MC_TRAANG_ANGLE_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAANG_ANGLE_2 (C, M)		MD 24750: $MC_TRAANG_ANGLE_2		
MD 24750: $MC_TRAANG_ANGLE_2				

TRAANG_ANGLE_2 (C, M)		MD 24750: $MC_TRAANG_ANGLE_2		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAANG_BASE_TOOL_1 (C, M)		MD 24710: $MC_TRAANG_BASE_TOOL_1		
MD 24710: $MC_TRAANG_BASE_TOOL_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAANG_BASE_TOOL_2 (C, M)		MD 24760: $MC_TRAANG_BASE_TOOL_2		
MD 24760: $MC_TRAANG_BASE_TOOL_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAANG_PARALLEL_ACCEL_RES_1 (C, M)		MD 24721: $MC_TRAANG_PARALLEL_ACCEL_RES_1		
MD 24721: $MC_TRAANG_PARALLEL_ACCEL_RES_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAANG_PARALLEL_ACCEL_RES_2 (C, M)		MD 24771: $MC_TRAANG_PARALLEL_ACCEL_RES_2		
MD 24771: $MC_TRAANG_PARALLEL_ACCEL_RES_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAANG_PARALLEL_VELO_RES_1 (C, M)		MD 24720: $MC_TRAANG_PARALLEL_VELO_RES_1		
MD 24720: $MC_TRAANG_PARALLEL_VELO_RES_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAANG_PARALLEL_VELO_RES_2 (C, M)		MD 24770: $MC_TRAANG_PARALLEL_VELO_RES_2		
MD 24770: $MC_TRAANG_PARALLEL_VELO_RES_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACE_SCOPE_MASK (C, M)		MD 22708: $MC_TRACE_SCOPE_MASK		
MD 22708: $MC_TRACE_SCOPE_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

TRACE_STARTTRACE_EVENT (C, M)		MD 22700: $MC_TRACE_STARTTRACE_EVENT		
MD 22700: $MC_TRACE_STARTTRACE_EVENT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

TRACE_STARTTRACE_STEP (C, M)		MD 22702: $MC_TRACE_STARTTRACE_STEP		
MD 22702: $MC_TRACE_STARTTRACE_STEP				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRACE_STOPTRACE_EVENT (C, M)		MD 22704: $MC_TRACE_STOP-TRACE_EVENT		
MD 22704: $MC_TRACE_STOPTRACE_EVENT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

TRACE_STOPTRACE_STEP (C, M)		MD 22706: $MC_TRACE_STOP-TRACE_STEP		
MD 22706: $MC_TRACE_STOPTRACE_STEP				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRACE_VARIABLE_INDEX (C, M)		MD 22712: $MC_TRACE_VARIABLE_IN-DEX		
MD 22712: $MC_TRACE_VARIABLE_INDEX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRACE_VARIABLE_NAME (C, M)		MD 22710: $MC_TRACE_VARIA-BLE_NAME		
MD 22710: $MC_TRACE_VARIABLE_NAME				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRACLG_CONTACT_LOWER_LIMIT (C, M)		MD 21520: $MC_TRACLG_CON-TACT_LOWER_LIMIT		
MD 21520: $MC_TRACLG_CONTACT_LOWER_LIMIT				

TRACLG_CONTACT_LOWER_LIMIT (C, M)		MD 21520: $MC_TRACLG_CONTACT_LOWER_LIMIT		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_CONTACT_UPPER_LIMIT (C, M)		MD 21518: $MC_TRACLG_CONTACT_UPPER_LIMIT		
MD 21518: $MC_TRACLG_CONTACT_UPPER_LIMIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_CTRLSPI_NR (C, M)		MD 21524: $MC_TRACLG_CTRLSPI_NR		
MD 21524: $MC_TRACLG_CTRLSPI_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

TRACLG_CTRLSPI_VERT_OFFSET (C, M)		MD 21502: $MC_TRACLG_CTRLSPI_VERT_OFFSET		
MD 21502: $MC_TRACLG_CTRLSPI_VERT_OFFSET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_G0_IS_SPECIAL (C, M)		MD 21526: $MC_TRACLG_G0_IS_SPECIAL		
MD 21526: $MC_TRACLG_G0_IS_SPECIAL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

TRACLG_GRINDSPI_HOR_OFFSET (C, M)		MD 21501: $MC_TRACLG_GRINDSPI_HOR_OFFSET		
MD 21501: $MC_TRACLG_GRINDSPI_HOR_OFFSET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_GRINDSPI_NR (C, M)		MD 21522: $MC_TRACLG_GRIND-SPI_NR		
MD 21522: $MC_TRACLG_GRINDSPI_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

TRACLG_GRINDSPI_VERT_OFFSET (C, M)		MD 21500: $MC_TRACLG_GRIND-SPI_VERT_OFFSET		
MD 21500: $MC_TRACLG_GRINDSPI_VERT_OFFSET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_HOR_DIR_SUPPORTAX_1 (C, M)		MD 21510: $MC_TRACLG_HOR_DIR_SUPPOR-TAX_1		
MD 21510: $MC_TRACLG_HOR_DIR_SUPPORTAX_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_HOR_DIR_SUPPORTAX_2 (C, M)		MD 21514: $MC_TRACLG_HOR_DIR_SUPPOR-TAX_2		
MD 21514: $MC_TRACLG_HOR_DIR_SUPPORTAX_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_SUPPORT_HOR_OFFSET (C, M)		MD 21506: $MC_TRACLG_SUP-PORT_HOR_OFFSET		
MD 21506: $MC_TRACLG_SUPPORT_HOR_OFFSET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_SUPPORT_LEAD_ANGLE (C, M)	MD 21516: $MC_TRACLG_SUP-PORT_LEAD_ANGLE	
MD 21516: $MC_TRACLG_SUPPORT_LEAD_ANGLE		

TRACLG_SUPPORT_LEAD_ANGLE (C, M)		MD 21516: $MC_TRACLG_SUPPORT_LEAD_ANGLE		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_SUPPORT_VERT_OFFSET (C, M)		MD 21504: $MC_TRACLG_SUPPORT_VERT_OFFSET		
MD 21504: $MC_TRACLG_SUPPORT_VERT_OFFSET				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_VERT_DIR_SUPPORTAX_1 (C, M)		MD 21508: $MC_TRACLG_VERT_DIR_SUPPORTAX_1		
MD 21508: $MC_TRACLG_VERT_DIR_SUPPORTAX_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACLG_VERT_DIR_SUPPORTAX_2 (C, M)		MD 21512: $MC_TRACLG_VERT_DIR_SUPPORTAX_2		
MD 21512: $MC_TRACLG_VERT_DIR_SUPPORTAX_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACYL_BASE_TOOL_1 (C, M)		MD 24820: $MC_TRACYL_BASE_TOOL_1		
MD 24820: $MC_TRACYL_BASE_TOOL_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRACYL_BASE_TOOL_2 (C, M)		MD 24870: $MC_TRACYL_BASE_TOOL_2		
MD 24870: $MC_TRACYL_BASE_TOOL_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRACYL_ROT_AX_OFFSET_1 (C, M)		MD 24800: $MC_TRACYL_ROT_AX_OFFSET_1		
MD 24800: $MC_TRACYL_ROT_AX_OFFSET_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACYL_ROT_AX_OFFSET_2 (C, M)		MD 24850: $MC_TRACYL_ROT_AX_OFFSET_2		
MD 24850: $MC_TRACYL_ROT_AX_OFFSET_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACYL_ROT_SIGN_IS_PLUS_1 (C, M)		MD 24810: $MC_TRACYL_ROT_SIGN_IS_PLUS_1		
MD 24810: $MC_TRACYL_ROT_SIGN_IS_PLUS_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

TRACYL_ROT_SIGN_IS_PLUS_2 (C, M)		MD 24860: $MC_TRACYL_ROT_SIGN_IS_PLUS_2		
MD 24860: $MC_TRACYL_ROT_SIGN_IS_PLUS_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

TRAFO5_BASE_TOOL_1 (C, M)		MD 24550: $MC_TRAFO5_BASE_TOOL_1		
MD 24550: $MC_TRAFO5_BASE_TOOL_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO5_BASE_TOOL_2 (C, M)		MD 24650: $MC_TRAFO5_BASE_TOOL_2		
MD 24650: $MC_TRAFO5_BASE_TOOL_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO5_JOINT_OFFSET_1 (C, M)		MD 24560: $MC_TRAFO5_JOINT_OFFSET_1		
MD 24560: $MC_TRAFO5_JOINT_OFFSET_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO5_JOINT_OFFSET_2 (C, M)		MD 24660: $MC_TRAFO5_JOINT_OFFSET_2		
MD 24660: $MC_TRAFO5_JOINT_OFFSET_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO5_NON_POLE_LIMIT_1 (C, M)		MD 24530: $MC_TRAFO5_NON_POLE_LIMIT_1		
MD 24530: $MC_TRAFO5_NON_POLE_LIMIT_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAFO5_NON_POLE_LIMIT_2 (C, M)		MD 24630: $MC_TRAFO5_NON_POLE_LIMIT_2		
MD 24630: $MC_TRAFO5_NON_POLE_LIMIT_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAFO5_NUTATOR_AX_ANGLE_1 (C, M)		MD 24564: $MC_TRAFO5_NUTATOR_AX_ANGLE_1		
MD 24564: $MC_TRAFO5_NUTATOR_AX_ANGLE_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAFO5_NUTATOR_AX_ANGLE_2 (C, M)		MD 24664: $MC_TRAFO5_NUTATOR_AX_ANGLE_2		
MD 24664: $MC_TRAFO5_NUTATOR_AX_ANGLE_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAFO5_PART_OFFSET_1 (C, M)		MD 24500: $MC_TRAFO5_PART_OFFSET_1		
MD 24500: $MC_TRAFO5_PART_OFFSET_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO5_PART_OFFSET_2 (C, M)		MD 24600: $MC_TRAFO5_PART_OFF-SET_2		
MD 24600: $MC_TRAFO5_PART_OFFSET_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO5_POLE_LIMIT_1 (C, M)		MD 24540: $MC_TRAFO5_POLE_LIMIT_1		
MD 24540: $MC_TRAFO5_POLE_LIMIT_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAFO5_POLE_LIMIT_2 (C, M)		MD 24640: $MC_TRAFO5_POLE_LIMIT_2		
MD 24640: $MC_TRAFO5_POLE_LIMIT_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRAFO5_ROT_AX_OFFSET_1 (C, M)		MD 24510: $MC_TRAFO5_ROT_AX_OFF-SET_1		
MD 24510: $MC_TRAFO5_ROT_AX_OFFSET_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO5_ROT_AX_OFFSET_2 (C, M)		MD 24610: $MC_TRAFO5_ROT_AX_OFF-SET_2		
MD 24610: $MC_TRAFO5_ROT_AX_OFFSET_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

TRAFO5_ROT_AX_OFFSET_2 (C, M)	MD 24610: $MC_TRAFO5_ROT_AX_OFF-SET_2	
	2	
	Read access	

TRAFO5_ROT_SIGN_IS_PLUS_1 (C, M)		MD 24520: $MC_TRA-FO5_ROT_SIGN_IS_PLUS_1		
MD 24520: $MC_TRAFO5_ROT_SIGN_IS_PLUS_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO5_ROT_SIGN_IS_PLUS_2 (C, M)		MD 24620: $MC_TRA-FO5_ROT_SIGN_IS_PLUS_2		
MD 24620: $MC_TRAFO5_ROT_SIGN_IS_PLUS_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO5_TOOL_ROT_AX_OFFSET_1 (C, M)		MD 24562: $MC_TRA-FO5_TOOL_ROT_AX_OFFSET_1		
MD 24562: $MC_TRAFO5_TOOL_ROT_AX_OFFSET_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO5_TOOL_ROT_AX_OFFSET_2 (C, M)		MD 24662: $MC_TRA-FO5_TOOL_ROT_AX_OFFSET_2		
MD 24662: $MC_TRAFO5_TOOL_ROT_AX_OFFSET_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

TRAFO5_TOOL_ROT_AX_OFFSET_2 (C, M)	MD 24662: $MC_TRA-FO5_TOOL_ROT_AX_OFFSET_2	
Addressing		
Line index:	Max. line index:	
	2	
	Read access	

TRAFO_AXES_IN_1 (C, M)		MD 24110: $MC_TRAFO_AXES_IN_1		
MD 24110: $MC_TRAFO_AXES_IN_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_AXES_IN_2 (C, M)		MD 24210: $MC_TRAFO_AXES_IN_2		
MD 24210: $MC_TRAFO_AXES_IN_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_AXES_IN_3 (C, M)		MD 24310: $MC_TRAFO_AXES_IN_3		
MD 24310: $MC_TRAFO_AXES_IN_3				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_AXES_IN_4 (C, M)		MD 24410: $MC_TRAFO_AXES_IN_4		
MD 24410: $MC_TRAFO_AXES_IN_4				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	

TRAFO_AXES_IN_4 (C, M)	MD 24410: $MC_TRAFO_AXES_IN_4	
	2	
	Read access	

TRAFO_AXES_IN_5 (C, M)		MD 24432: $MC_TRAFO_AXES_IN_5		
MD 24432: $MC_TRAFO_AXES_IN_5				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_AXES_IN_6 (C, M)		MD 24442: $MC_TRAFO_AXES_IN_6		
MD 24442: $MC_TRAFO_AXES_IN_6				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_AXES_IN_7 (C, M)		MD 24452: $MC_TRAFO_AXES_IN_7		
MD 24452: $MC_TRAFO_AXES_IN_7				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_AXES_IN_8 (C, M)		MD 24462: $MC_TRAFO_AXES_IN_8		
MD 24462: $MC_TRAFO_AXES_IN_8				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_CHANGE_M_CODE (C, M)		MD 22534: $MC_TRAFO_CHANGE_M_CODE		
MD 22534: $MC_TRAFO_CHANGE_M_CODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TRAFO_GEOAX_ASSIGN_TAB_1 (C, M)		MD 24120: $MC_TRAFO_GEOAX_ASSIGN_TAB_1		
MD 24120: $MC_TRAFO_GEOAX_ASSIGN_TAB_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_GEOAX_ASSIGN_TAB_2 (C, M)		MD 24220: $MC_TRAFO_GEOAX_ASSIGN_TAB_2		
MD 24220: $MC_TRAFO_GEOAX_ASSIGN_TAB_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_GEOAX_ASSIGN_TAB_3 (C, M)		MD 24320: $MC_TRAFO_GEOAX_ASSIGN_TAB_3		
MD 24320: $MC_TRAFO_GEOAX_ASSIGN_TAB_3				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_GEOAX_ASSIGN_TAB_4 (C, M)		MD 24420: $MC_TRAFO_GEOAX_ASSIGN_TAB_4		
MD 24420: $MC_TRAFO_GEOAX_ASSIGN_TAB_4				

TRAFO_GEOAX_ASSIGN_TAB_4 (C, M)		MD 24420: $MC_TRAFO_GEOAX_ASSIGN_TAB_4		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_GEOAX_ASSIGN_TAB_5 (C, M)		MD 24434: $MC_TRAFO_GEOAX_ASSIGN_TAB_5		
MD 24434: $MC_TRAFO_GEOAX_ASSIGN_TAB_5				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_GEOAX_ASSIGN_TAB_6 (C, M)		MD 24444: $MC_TRAFO_GEOAX_ASSIGN_TAB_6		
MD 24444: $MC_TRAFO_GEOAX_ASSIGN_TAB_6				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_GEOAX_ASSIGN_TAB_7 (C, M)		MD 24454: $MC_TRAFO_GEOAX_ASSIGN_TAB_7		
MD 24454: $MC_TRAFO_GEOAX_ASSIGN_TAB_7				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_GEOAX_ASSIGN_TAB_8 (C, M)		MD 24464: $MC_TRAFO_GEOAX_ASSIGN_TAB_8		
MD 24464: $MC_TRAFO_GEOAX_ASSIGN_TAB_8				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRAFO_RESET_VALUE (C, M)		MD 20140: $MC_TRAFO_RESET_VALUE		
MD 20140: $MC_TRAFO_RESET_VALUE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

TRAFO_TYPE_1 (C, M)		MD 24100: $MC_TRAFO_TYPE_1		
MD 24100: $MC_TRAFO_TYPE_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TRAFO_TYPE_2 (C, M)		MD 24200: $MC_TRAFO_TYPE_2		
MD 24200: $MC_TRAFO_TYPE_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TRAFO_TYPE_3 (C, M)		MD 24300: $MC_TRAFO_TYPE_3		
MD 24300: $MC_TRAFO_TYPE_3				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TRAFO_TYPE_4 (C, M)		MD 24400: $MC_TRAFO_TYPE_4		
MD 24400: $MC_TRAFO_TYPE_4				
Units and value ranges				
Physical unit:	Data type:			

TRAFO_TYPE_4 (C, M)		MD 24400: $MC_TRAFO_TYPE_4		
-	TYPE_DWORD			
			Read access	

TRAFO_TYPE_5 (C, M)		MD 24430: $MC_TRAFO_TYPE_5		
MD 24430: $MC_TRAFO_TYPE_5				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TRAFO_TYPE_6 (C, M)		MD 24440: $MC_TRAFO_TYPE_6		
MD 24440: $MC_TRAFO_TYPE_6				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TRAFO_TYPE_7 (C, M)		MD 24450: $MC_TRAFO_TYPE_7		
MD 24450: $MC_TRAFO_TYPE_7				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TRAFO_TYPE_8 (C, M)		MD 24460: $MC_TRAFO_TYPE_8		
MD 24460: $MC_TRAFO_TYPE_8				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TRANSMIT_BASE_TOOL_1 (C, M)		MD 24920: $MC_TRANS-MIT_BASE_TOOL_1		
MD 24920: $MC_TRANSMIT_BASE_TOOL_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRANSMIT_BASE_TOOL_2 (C, M)		MD 24970: $MC_TRANSMIT_BASE_TOOL_2		
MD 24970: $MC_TRANSMIT_BASE_TOOL_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

TRANSMIT_POLE_SIDE_FIX_1 (C, M)		MD 24911: $MC_TRANSMIT_POLE_SIDE_FIX_1		
MD 24911: $MC_TRANSMIT_POLE_SIDE_FIX_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

TRANSMIT_POLE_SIDE_FIX_2 (C, M)		MD 24961: $MC_TRANSMIT_POLE_SIDE_FIX_2		
MD 24961: $MC_TRANSMIT_POLE_SIDE_FIX_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

TRANSMIT_ROT_AX_OFFSET_1 (C, M)		MD 24900: $MC_TRANSMIT_ROT_AX_OFFSET_1		
MD 24900: $MC_TRANSMIT_ROT_AX_OFFSET_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRANSMIT_ROT_AX_OFFSET_2 (C, M)		MD 24950: $MC_TRANSMIT_ROT_AX_OFFSET_2		
MD 24950: $MC_TRANSMIT_ROT_AX_OFFSET_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRANSMIT_ROT_SIGN_IS_PLUS_1 (C, M)		MD 24910: $MC_TRANSMIT_ROT_SIGN_IS_PLUS_1		
MD 24910: $MC_TRANSMIT_ROT_SIGN_IS_PLUS_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

TRANSMIT_ROT_SIGN_IS_PLUS_2 (C, M)		MD 24960: $MC_TRANSMIT_ROT_SIGN_IS_PLUS_2		
MD 24960: $MC_TRANSMIT_ROT_SIGN_IS_PLUS_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

WORKAREA_WITH_TOOL_RADIUS (C, M)		MD 21020: $MC_WORKAREA_WITH_TOOL_RADIUS		
MD 21020: $MC_WORKAREA_WITH_TOOL_RADIUS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

X_AXIS_IN_OLD_X_Z_PLANE (C, M)		MD 21110: $MC_X_AXIS_IN_OLD_X_Z_PLANE		
MD 21110: $MC_X_AXIS_IN_OLD_X_Z_PLANE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

MM_ABSBLOCK (C, M)		MD 28400: $MC_MM_ABSBLOCK		
MD 28400: $MC_MM_ABSBLOCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

AUXFU_ASSOC_M1_VALUE (C, M)		MD 22256: $MC_AUXFU_ASSOC_M1_VALUE		
MD 22256: $MC_AUXFU_ASSOC_M1_VALUE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

MM_SYSTEM_DATAFRAME_MASK (C, M)		MD 28083: $MC_MM_SYSTEM_DATAFRAME_MASK		
MD 28083: $MC_MM_SYSTEM_DATAFRAME_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

MM_PROTOC_NUM_ETP_STD_TYP (C, M)		MD 28302: $MC_MM_PROTOC_NUM_ETP_STD_TYP		
MD 28302: $MC_MM_PROTOC_NUM_ETP_STD_TYP				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

3.2.3.14 NIB: Nibbling data

The module NIB contains technology-specific data for nibbling.

actPunchRate (C, NIB)				N4
Strokes per minute				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

actPunchRate (C, NIB)				N4
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

automCutSegment (C, NIB)				N4
Automatic block division Identifier that indicates which type of automatic block division is active. The division is specified by the commands 'SPP' and 'SPN' in the part program.				
Return value	0 = no block division 1 = number of segments per block ('SNP') 2 = segments of fixed length ('SPP')			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

numStrokes (C, NIB)				N4
Number of strokes Number of strokes when the instruction 'SPN' divides the block into segments (variable 'automCutSegment' = 1).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

partDistance (C, NIB)				N4
Length of the path between punches If the block has been divided in segments with the instruction 'SPP' (variable 'automCutSegment' = 2) the variable specifies the length of the path between the punches.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

punchActive (C, NIB)				N4
Punching or nibbling active Identification of punching or nibbling active. The part program turns off/on punching and nibbling with 'SPOF', 'SON' and 'PON'. Rapid punching and nibbling are turned on/off with 'SONS' and 'PONS'. The variable 'punchActive' specified the present state.				
Return value	0 = inactive 1 = punching active 2 = nibbling active 3 = rapid punching active 4 = rapid nibbling active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

punchDelayActive (C, NIB)				N4
Punching with delay active Identifier that indicates whether punching with delay is active. The part program can turn on/off the delay with the instructions 'PDELAYON' and 'PDELAYOF'. The variable 'PunchDelayActive' indicates the present state.				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

punchDelayTime (C, NIB)		SD 42400: PUNCH_DWELL_TIME		N4
Punching delay time				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

strokeNr (C, NIB)				
Current stroke number				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

3.2.3.15 O: Internal

C-O

LOOKAH_NUM_CHECKED_BLOCKS (C, O)		MD 29000: $OC_LOO-KAH_NUM_CHECKED_BLOCKS		
MD 29000: $OC_LOOKAH_NUM_CHECKED_BLOCKS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

3.2.3.16 PA: Protection zones

Up to 10 protection zones can be defined. Each protection zone is described by a polygon function consisting of up to 10 elements. The maximum permissible number of protection zones is specified in "$$numProtArea" in the module Y in area C. The maximum permissible number of polygon definition elements is specified in "$ $numContourInProtArea" in module Y in area C. Module PA contains the individual coordinates of the polygon elements. The protection zones are addressed via the variable indices.

The classification as NCK or channel-specific protection zone does not affect the protection zone monitoring function but simply indicates the area in which the protection zone is registered.

The physical unit actually used for length quantities is defined in "$$/C/SGA/extUnit" in module SGA in area C.

MDD_PA_CENT_ABS_0 (C, PA)		$SC_PA_CENT_ABS[x,0] x = Number protection zone		A3
Abscissa arc centre of 1st contour element Absolute abscissa value of arc centre of 1st contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CENT_ABS_0 (C, PA)		$SC_PA_CENT_ABS[x,0] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_1 (C, PA)		$SC_PA_CENT_ABS[x,1] x = Number protection zone		A3
Abscissa arc centre of 2nd contour element Absolute abscissa value of arc centre of 2nd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_2 (C, PA)		$SC_PA_CENT_ABS[x,2] x = Number protection zone		A3
Abscissa arc centre of 3rd contour element Absolute abscissa value of arc centre of 3rd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_3 (C, PA)		$SC_PA_CENT_ABS[x,3] x = Number protection zone		A3
Abscissa arc centre of 4th contour element Absolute abscissa value of arc centre of 4th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_4 (C, PA)		$SC_PA_CENT_ABS[x,4] x = Number protection zone		A3
Abscissa arc centre of 5th contour element Absolute abscissa value of arc centre of 5th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_5 (C, PA)		$SC_PA_CENT_ABS[x,5] x = Number protection zone		A3
Abscissa arc centre of 6th contour element Absolute abscissa value of arc centre of 6th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_6 (C, PA)		$SC_PA_CENT_ABS[x,6] x = Number protection zone		A3
Abscissa arc centre of 7th contour element Absolute abscissa value of arc centre of 7th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_7 (C, PA)		$SC_PA_CENT_ABS[x,7] x = Number protection zone		A3
Abscissa arc centre of 8th contour element Absolute abscissa value of arc centre of 8th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CENT_ABS_7 (C, PA)		$SC_PA_CENT_ABS[x,7] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_8 (C, PA)		$SC_PA_CENT_ABS[x,8] x = Number protection zone		A3
Abscissa arc centre of 9th contour element Absolute abscissa value of arc centre of 9th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_9 (C, PA)		$SC_PA_CENT_ABS[x,9] x = Number protection zone		A3
Abscissa arc centre of 10th contour element Absolute abscissa value of arc centre of 10th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_0 (C, PA)		$SC_PA_CENT_ORD[x,0] x = Number protection zone		A3
Ordinate arc centre of 1st contour element Absolute ordinate value of arc centre of 1st contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_1 (C, PA)		$SC_PA_CENT_ORD[x,1] x = Number protection zone		A3
Ordinate arc centre of 2nd contour element Absolute ordinate value of arc centre of 2nd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_2 (C, PA)		$SC_PA_CENT_ORD[x,2] x = Number protection zone		A3
Odinate arc centre of 3rd contour element Absolute ordinate value of arc centre of 3rd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_3 (C, PA)		$SC_PA_CENT_ORD[x,3] x = Number protection zone		A3
Ordinate arc centre of 4th contour element Absolute ordinate value of arc centre of 4th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_4 (C, PA)		$SC_PA_CENT_ORD[x,4] x = Number protection zone		A3
Ordinate arc centre of 5th contour element Absolute ordinate value of arc centre of 5th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CENT_ORD_4 (C, PA)		$SC_PA_CENT_ORD[x,4] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_5 (C, PA)		$SC_PA_CENT_ORD[x,5] x = Number protection zone		A3
Ordinate arc centre of 6th contour element Absolute ordinate value of arc centre of 6th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_6 (C, PA)		$SC_PA_CENT_ORD[x,6] x = Number protection zone		A3
Ordinate arc centre of 7th contour element Absolute ordinate value of arc centre of 7th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_7 (C, PA)		$SC_PA_CENT_ORD[x,7] x = Number protection zone		A3
Ordinate arc centre of 8th contour element Absolute ordinate value of arc centre of 8th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_8 (C, PA)		$SC_PA_CENT_ORD[x,8] x = Number protection zone		A3
Ordinate arc centre of 9th contour element Absolute ordinate value of arc centre of 9th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_9 (C, PA)		$SC_PA_CENT_ORD[x,9] x = Number protection zone		A3
Ordinate arc centre of 10th contour element Absolute ordinate value of arc centre of 10th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_0 (C, PA)		$SC_PA_CONT_ABS[x,0] x = Number protection zone		A3
Abscissa end point of 1st contour element Absolute abscissa value of end point of 1st contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_1 (C, PA)		$SC_PA_CONT_ABS[x,1] x = Number protection zone		A3
Abscissa end point of 2nd contour element Absolute abscissa value of end point of 2nd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CONT_ABS_1 (C, PA)		$SC_PA_CONT_ABS[x,1] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_2 (C, PA)		$SC_PA_CONT_ABS[x,2] x = Number protection zone		A3
Abscissa end point of 3rd contour element Absolute abscissa value of end point of 3rd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_3 (C, PA)		$SC_PA_CONT_ABS[x,3] x = Number protection zone		A3
Abscissa end point of 4th contour element Absolute abscissa value of end point of 4th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_4 (C, PA)		$SC_PA_CONT_ABS[x,4] x = Number protection zone		A3
Abscissa end point of 5th contour element Absolute abscissa value of end point of 5th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_5 (C, PA)		$SC_PA_CONT_ABS[x,5] x = Number protection zone		A3
Abscissa end point of 6th contour element Absolute abscissa value of end point of 6th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_6 (C, PA)		$SC_PA_CONT_ABS[x,6] x = Number protection zone		A3
Abscissa end point of 7th contour element Absolute abscissa value of end point of 7th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_7 (C, PA)		$SC_PA_CONT_ABS[x,7] x = Number protection zone		A3
Abscissa end point of 8th contour element Absolute abscissa value of end point of 8th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_8 (C, PA)		$SC_PA_CONT_ABS[x,8] x = Number protection zone		A3
Abscissa end point of 9th contour element Absolute abscissa value of end point of 9th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CONT_ABS_8 (C, PA)		$SC_PA_CONT_ABS[x,8] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_9 (C, PA)		$SC_PA_CONT_ABS[x,9] x = Number protection zone		A3
Abscissa end point of 10th contour element Absolute abscissa value of end point of 10th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_0 (C, PA)		$SC_PA_CONT_ORD[x,0] x = Number protection zone		A3
Ordinate end point of 1st contour element Absolute ordinate value of end point of 1st contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_1 (C, PA)		$SC_PA_CONT_ORD[x,1] x = Number protection zone		A3
Ordinate end point of 2nd contour element Absolute ordinate value of end point of 2nd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_2 (C, PA)		$SC_PA_CONT_ORD[x,2] x = Number protection zone		A3
Odinate end point of 3rd contour element Absolute ordinate value of end point of 3rd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_3 (C, PA)		$SC_PA_CONT_ORD[x,3] x = Number protection zone		A3
Ordinate end point of 4th contour element Absolute ordinate value of end point of 4th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_4 (C, PA)		$SC_PA_CONT_ORD[x,4] x = Number protection zone		A3
Ordinate end point of 5th contour element Absolute ordinate value of end point of 5th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_5 (C, PA)		$SC_PA_CONT_ORD[x,5] x = Number protection zone		A3
Ordinate end point of 6th contour element Absolute ordinate value of end point of 6th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CONT_ORD_5 (C, PA)		$SC_PA_CONT_ORD[x,5] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_6 (C, PA)		$SC_PA_CONT_ORD[x,6] x = Number protection zone		A3
Ordinate end point of 7th contour element Absolute ordinate value of end point of 7th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_7 (C, PA)		$SC_PA_CONT_ORD[x,7] x = Number protection zone		A3
Ordinate end point of 8th contour element Absolute ordinate value of end point of 8th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_8 (C, PA)		$SC_PA_CONT_ORD[x,8] x = Number protection zone		A3
Ordinate end point of 9th contour element Absolute ordinate value of end point of 9th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_9 (C, PA)		$SC_PA_CONT_ORD[x,9] x = Number protection zone		A3
Ordinate end point of 10th contour element Absolute ordinate value of end point of 10th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_MINUS_LIM (C, PA)		$SC_PA_MINUS_LIM[x] x = Number protection zone		A3
Lower limit of protection zone, applicate Limitation in the minus direction of the protection zone in the axis that is perpendicular to the polygon definition (applicate)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_PLUS_LIM (C, PA)		$SC_PA_PLUS_LIM[x] x = Number protection zone		A3
Upper boundary of protection zone, applicate Limitation of the protection zone in the plus direction of the axis that is perpendicular to the polygon definition (applicate)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_ACTIV_IMMED (C, PA)	$SC_PA_ACTIV_IMMED[x] x = Number protection zone	A3
Active immediately after referencing Code for "active immediately after referencing", i.e. the protection zone is active as soon as the control has been started up and the axes have been referenced		
Return value	0 = protection zone is not active immediately 1 = protection zone is active immediately	

MDU_PA_ACTIV_IMMED (C, PA)		$SC_PA_ACTIV_IMMED[x] x = Number protection zone		A3
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_NUM (C, PA)		$SC_PA_CONT_NUM[x] x = Number protection zone		A3
Number of valid contour elements				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	$$numContourInProtArea
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_0 (C, PA)		$SC_PA_CONT_TYP[x,0] x = Number protection zone		A3
Contour type of 1st contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_1 (C, PA)		$SC_PA_CONT_TYP[x,1] x = Number protection zone		A3
Contour type of 2nd contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

MDU_PA_CONT_TYP_1 (C, PA)		$SC_PA_CONT_TYP[x,1] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_2 (C, PA)		$SC_PA_CONT_TYP[x,2] x = Number protection zone		A3
Contour type of 3rd contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_3 (C, PA)		$SC_PA_CONT_TYP[x,3] x = Number protection zone		A3
Contour type of 4th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_4 (C, PA)		$SC_PA_CONT_TYP[x,4] x = Number protection zone		A3
Contour type of 5th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

MDU_PA_CONT_TYP_4 (C, PA)		$SC_PA_CONT_TYP[x,4] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_5 (C, PA)		$SC_PA_CONT_TYP[x,5] x = Number protection zone		A3
Contour type of 6th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_6 (C, PA)		$SC_PA_CONT_TYP[x,6] x = Number protection zone		A3
Contour type of 7th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_7 (C, PA)		$SC_PA_CONT_TYP[x,7] x = Number protection zone		A3
Contour type of 8th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

MDU_PA_CONT_TYP_7 (C, PA)		$SC_PA_CONT_TYP[x,7] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_8 (C, PA)		$SC_PA_CONT_TYP[x,8] x = Number protection zone		A3
Contour type of 9th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_9 (C, PA)		$SC_PA_CONT_TYP[x,9] x = Number protection zone		A3
Contour type of 10th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_LIM_3DIM (C, PA)		$SC_PA_LIM_3DIM[x] x = Number protection zone		A3
Limitation of protection zone applicate Code for limitation of protection zone in the axis that is perpendicular to polygon definition (applicate)				
Return value	0 = no limitation 1 = limitation in positive direction 2 = limitation in negative direction 3 = limitation in both directions			
Units and value ranges				
Physical unit:	Data type:			

MDU_PA_LIM_3DIM (C, PA)		$SC_PA_LIM_3DIM[x] x = Number protection zone		A3
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_ORI (C, PA)		$SC_PA_ORI[x] x = Number protection zone		A3
Plane assignment of protection zone Code for plane assignment of protection zone				
Return value	0 = G17 1 = G18 2 = G19			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_TW (C, PA)		$SC_PA_T_W[x] x = Number protection zone		A3
Workpiece or tool-related protection zone Code for workpiece or tool-oriented protection zone				
Return value	0 = workpiece-related 1 = reserved 2 = reserved 3 = tool-related			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

collisionAlarm (C, PA)		OD19830 $ON_COLLISION_MASK		
Collision of two protection zones				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

collisionAlarm (C, PA)	OD19830 $ON_COLLISION_MASK	
Addressing		
Line index:	Max. line index:	
	3	
	Read access	

acCollPos (C, PA)				
Collision position in world coordinates Contact point between two collision bodies during a collision alarm. Corresponds with system variable $AC_COLLPOS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

declarProtObject (C, PA)				
Declaration of a variable protection zone				
Return value	0=No object 1=WORKPIECE 2=FIXTURE			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

declarProtObjectString (C, PA)				
String parameter(s) of a protection zone String parameter(s) of the declaration of a variable protection zone				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			4	
			Read access	

declarProtObjectReal (C, PA)				
Real parameter(s) of a protection zone Real parameters of the declaration of a variable protection zone				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

workpieceStatus (C, PA)				
Error status after calling the procedure WORKPIECE Error status after calling the procedure WORKPIECE Corresponds to the system variable $P_WP_STAT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_WORD			
			Read access	

fixtureStatus (C, PA)				
Error status after calling the procedure FIXTURE Error status after calling the procedure FIXTURE Corresponds to the system variable $P_FIX_STAT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_WORD			
			Read access	

3.2.3.17 RP: Arithmetic parameter

Arithmetic parameters are special predefined variables which are addressed with the letter R followed by a number. The contents and meaning of an arithmetic parameter are defined by the programmer of a part program. 100 R variables are defined by default. The number of R variables can be set via machine data 28050 (MM_NUM_R_PARAM).

rpa (C, RP)		$R[x] x = ParameterNo		PA
Alternative name:	R			
R variables				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

rpa (C, RP)		$R[x] x = ParameterNo		PA
Addressing				
Line index:			Max. line index:	
			MM_NUM_R_PARAM + 1	
Validity:	From NCK version 3.6		Read access and write access	

3.2.3.18 S: State data

During NC operation different internal states occur and system-specific data may change during operation. To distinguish those from system variables, they are classified as state data.

A distinction is made between:
- NCK-specific state data
- Mode-group-specific state data
- Channel-specific state data
- Drive-specific state data (FDD)
- Drive-specific state data (MSD)

G0Mode (C, S)		$AC_G0MODE		
G00: Path axes traverse as pos. axes G00 is active and $MC_G0_LINEAR_MODE is FALSE (Siemens mode) or $MC_EXTERN_G0_LINEAR_MODE is FALSE (ISO mode) and therefore non-linear interpolation is active with G0, i.e. the path axes are traversed as positioning axes.				
Return value	0: G00 not active 1: G00 and linear interpolation active 2: G00 and non-linear interpolation active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aGG (C, S)		$A_GG		
active G function in synchronized action				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$/N/YNCFL/Gruppe_NUM	
			Read access	

aLinkTransRate (C, S)		$A_LINK_TRANS_RATE		
Link transfer rate Link transfer rate Number of link variables that can still be transferred in the current IPO cycle via the NCU link communication. If this variable is read in the context of the lead, it always outputs the maximum available bandwidth.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_UWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aMonifact (C, S)		$A_MONIFACT		
Factor for tool life monitoring				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcAckC (C, S)		$AC_TC_ACKC		
Acknowledgment counter variable Counter variable: aTcAckC (AcknowledgeCounter) is incremented by 1 every time the PLC acknowledges a tool management command.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

aTcCmdC (C, S)		$AC_TC_CMDC		
Command counter variable Counter variable: aTcCmdC (CoMmandCounter) is incremented by 1 every time the tool management outputs a command to the PLC.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	

aTcCmdC (C, S)		$AC_TC_CMDC		
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

aTcFct (C, S)		$AC_TC_FCT		
Command number				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcLfn (C, S)		$AC_TC_LFN		
Source location number new tool Source location number of new tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcLfo (C, S)		$AC_TC_LFO		
Source location number old tool Source location number of old tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcLmyn (C, S)		$AC_TC_LMYN		
Location number of the new tool Owner location number of the new tool				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:

aTcLmyn (C, S)		$AC_TC_LMYN		
-	TYPE_UWORD		-1	32000
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcLtn (C, S)		$AC_TC_LTN		
Target location number new tool Target location number of new tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcLto (C, S)		$AC_TC_LTO		
Target location number old tool Target location number of old tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcMfn (C, S)		$AC_TC_MFN		
Source magazine new tool Source magazine of new tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcMfo (C, S)	$AC_TC_MFO	
Source magazine old tool Source magazine number of old tool		

aTcMfo (C, S)		$AC_TC_MFO		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcMmyn (C, S)		$AC_TC_MMYN		
Magazine of the new tool Owner magazine number of the new tool				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		-1	32000
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcMtn (C, S)		$AC_TC_MTN		
Target magazine new tool Target magazine number of new tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcMto (C, S)		$AC_TC_MTO		
Target magazine old tool Target magazine number of old tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcStatus (C, S)	$AC_TC_STATUS	
Command status		

aTcStatus (C, S)		$AC_TC_STATUS		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcThno (C, S)		$AC_TC_THNO		
Number of toolholder Number of toolholder for new tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcTno (C, S)		$AC_TC_TNO		
T number T number of new tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcToolls (C, S)		$AC_TC_TOOLIS		
Type of distance coding for the multitool 0 = Tool, 1,2,3 = Type of distance coding for the multitool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcDistance (C, S)		$AC_TC_DISTANCE		
Distance of the MT place of the loaded tool Distance of the multitool place of the loaded tool to the reference point				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcMttn (C, S)		$AC_TC_MTTN		
Number of the multitool of the tool Number of the multitool of the loaded tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcMtptn (C, S)		$AC_TC_MTLTN		
Number of the multitool place of the tool Number of the multitool place of the loaded tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aTcNumPlaces (C, S)		$AC_TC_MTNLOC		
Number of defined places in the multitool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aaAccLim (C, S)				
Axial acceleration override (ACC) Display of the axial acceleration override programmed by ACC. Depending on $MA_DYN_LIMIT_RESET_MASK, the value can remain active after Reset. The variable always shows the programmed acceleration override and not the currently active acceleration limit.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	100	1	200
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaAccLimA (C, S)		$AA_ACCLIMA[a]		
Axial acceleration overr. main run Axial acceleration override in main run				
Return value	1-200			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	100	1	200
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaJerkLim (C, S)				
Axial jerk override (JERKLIM) Display of the axial jerk override programmed by JERKLIM. Depending on $MA_DYN_LIMIT_RESET_MASK, the value can remain active after Reset. The variable always shows the programmed jerk override and not the currently active jerk limit.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	100	1	200
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaVeloLim (C, S)				
Axial velocity override (VELOLIM) Display of the axial velocity override programmed by VELOLIM. Depending on $MA_DYN_LIMIT_RESET_MASK, the value can remain active after Reset. The variable always shows the programmed velocity override and not the currently active velocity limit.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	100	1	200
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actCollPosMcsPacked (C, S)				
Collision position in MCS Position of a channel axis with a collision in the MCS. The positions can be read for all configured channel axes. Row index corresponds to 1 1st configured channel axis 2 2nd configured channel axis .. 20 20th configured channel axis				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			MAXNUM_AXES_PER_CHAN	
			Read access	

aaEgActive (C, S)		$AA_EG_ACTIVE[a,b]		
EG link active Electronic gear: Link to the specified master axis is operative, i.e. activated.				
Return value	0: Deactivated 1: Activated			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1

aaEgActive (C, S)	$AA_EG_ACTIVE[a,b]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes * $$numMachAxes	
	Read access	

aaEgAx (C, S)		$AA_EG_AX[n,a]		
EG axis number master axis Electronic gear: Axis number of nth master axis (1-n). (Axis index = axis number - 1)				
Return value	1-numMachAxes			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	1	$$numMachAxes
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * 5	
			Read access	

aaEgDenom (C, S)		$AA_EG_DENOM[a,b]		
EG denominator link factor Electronic gear: Denominator of link factor for the specified master axis. The link factor of the gear is the result of $AA_EG_NUMERA[a,b]/$AA_EG_DENOM[a,b].				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaEgNumLa (C, S)		$AA_EG_NUM_LA[a]		
EG number of master axes Electronic gear: Number of master axes specified with EGDEF. If the axis has not been specified with EGDEF as slave axis, the value is 0.				
Return value	0-5			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	5

aaEgNumLa (C, S)	$AA_EG_NUM_LA[a]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

aaEgNumera (C, S)		$AA_EG_NUMERA[a,b]		
EG numerator link factor Electronic gear: Numerator of link factor for the specified master axis. The link factor of the gear is the result of $AA_EG_NUMERA[a,b]/$AA_EG_DENOM[a,b].				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaEgSyn (C, S)		$AA_EG_SYN[a,b]		
EG synchronous position of master axis Electronic gear: Synchronous position for the specified master axis.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaEgSynFa (C, S)		$AA_EG_SYNFA[a]		
EG synchronous position of slave axis Electronic gear: Synchronous position for the slave axis.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaEgType (C, S)		$AA_EG_TYPE[a,b]		
EG link type Electronic gear: Type of link for the specified master axis				
Return value	0: Actual-value linkage 1: Setpoint linkage			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $$numMachAxes	
			Read access	

aaJerkLimA (C, S)		$AA_JERKLIMA[a]		
Axial jerk override main run Axial jerk override in run in				
Return value	1-200			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	100	1	200
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaMeasP1Valid (C, S)		$AA_MEAS_P1_VALID		
Save axial measuring point 1 Save axial measuring point P1 for workpiece and tool measurement				
Return value	0: Clear axial measuring point 1: Write actual axial values to axial measuring point			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMeasP2Valid (C, S)		$AA_MEAS_P2_VALID		
Save axial measuring point 2 Save axial measuring point P2 for workpiece and tool measurement				
Return value	0: Clear axial measuring point 1: Write actual axial values to axial measuring point			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMeasP3Valid (C, S)		$AA_MEAS_P3_VALID		
Save axial measuring point 3 Save axial measuring point P3 for workpiece and tool measurement				
Return value	0: Clear axial measuring point 1: Write actual axial values to axial measuring point			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMeasP4Valid (C, S)		$AA_MEAS_P4_VALID		
Save axial measuring point 4 Save axial measuring point P4 for workpiece and tool measurement				
Return value	0: Clear axial measuring point 1: Write actual axial values to axial measuring point			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMeasPoint1 (C, S)		$AA_MEAS_POINT1		
1st measuring point 1st measuring point for workpiece and tool measurement				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMeasPoint2 (C, S)		$AA_MEAS_POINT2		
2nd measuring point 2nd measuring point for workpiece and tool measurement				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMeasPoint3 (C, S)		$AA_MEAS_POINT3		
3rd measuring point 3rd measuring point for workpiece and tool measurement				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMeasPoint4 (C, S)		$AA_MEAS_POINT4		
4th measuring point 4th measuring point for workpiece and tool measurement				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMeasSetangle (C, S)		$AA_MEAS_SETANGLE		
Measurement setpoint angle Setpoint angle of an axis				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
			Read access and write access	

aaMeasSetpoint (C, S)		$AA_MEAS_SETPOINT		
Measurement setpoint position Setpoint position of edge, corner or hole				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMeasSpValid (C, S)		$AA_MEAS_SP_VALID		
Save axial setpoint Save axial setpoint for workpiece and tool measurement				
Return value	0: Clear axial setpoint 1: Validate axial setpoint			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaVeloLimA (C, S)		$AA_VELOLIMA[a]		
Axial velocity overr. main run Axial velocity override in main run				
Return value	1-200			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	100	1	200
Addressing				
Line index:			Max. line index:	

aaVeloLimA (C, S)	$AA_VELOLIMA[a]	
	$$numMachAxes	
	Read access	

acAlarmStat (C, S)	$AC_ALARM_STAT			
ESR alarm status !=0: Alarms are pending, the appropriate coded alarm reactions can be used as source for "Extended stop and retract".				
Return value	The data is bit-coded. Individual states can therefore be masked or evaluated separately if necessary (bits excluded below produce a value of 0) Bit2 = 1: NOREADY (active rapid deceleration + cancellation of servo enable) Bit6 = 1: STOPBYALARM (ramp stop of all channel axes) Bit9 = 1: SETVDI (VDI interface signal alarm setting) Bit13 = 1: FOLLOWUPBYALARM (follow-up)			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acAsup (C, S)		$AC_ASUP		
Reason for the activation of an ASUB Code number for the cause of activation of an ASUB				
Return value	The reasons are bit-coded. BIT0: Activation because of: User interrupt "ASUB with Blsync". BIT1: Activation because of: User interrupt "ASUB". BIT2: Activation because of: User interrupt "ASUB from channel state Ready". BIT3: Activation because of: User interrupt "ASUB in manual mode". BIT4: Activation because of: Activation because of: User interrupt "ASUB". BIT5: Activation because of: Abort of the subroutine repetition. BIT6: Activation because of: Activation of decoding single block. BIT7: Activation because of: Activation of DDTG. BIT8: Activation because of: Activation of axis synchronization. BIT9: Activation because of: Change of operating mode. BIT10: Activation because of: Program continuation under TeachIn or after TeachIn deactivation. BIT11: Activation because of: Selection overstore. BIT12: Activation because of: Alarm with reaction compensation block with REPOS (COMPBLOCKWITHREORG). BIT13: Activation because of: Retraction motion with G33 and Stop. BIT14: Activation because of: Activation of dry run feed. BIT15: Activation because of: Deactivation of dry run feed. BIT16: Activation because of: Activation of skip block. BIT17: Activation because of: Deactivation of skip block. BIT18: Activation because of: Activate machine data. BIT19: Activation because of: Activate tool offset. BIT20: Activation because of: System ASUB after search type SERUPRO has reached the search target.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acAxCtSwA (C, S)		$AC_AXCTSWA[CTn]		
Status of axis container Channel status of axis container rotation				
Return value	TRUE: The channel has enabled rotation for the axis container and rotation is still in progress. FALSE: Axis container rotation is already finished			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

acAxCtSwA (C, S)		$AC_AXCTSWA[CTn]		
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numContainer	
			Read access	

acConeAngle (C, S)		$AC_CONE_ANGLE		
Taper angle Currently effective taper angle for taper turning. The taper angle is specified by the setting data $SC_CONE_ANGLE and is effective in the operating mode JOG only.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
Degree	TYPE_DOUBLE	0	-90	90
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acDelt (C, S)		$AC_DELT		
Stored distance-to-go of the path in the WCS Stored distance-to-go of the path in the WCS after delete-distance-to-go of the path DELDTG for synchronous action (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acDtbb (C, S)		$AC_DTBB		
Distance from beginning of the block in the BCS Distance from the beginning of the block in the BCS (Note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acDtbw (C, S)		$AC_DTBW		
Distance from the beginning of block in the WCS Distance from the beginning of the block in the WCS (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acDteb (C, S)		$AC_DTEB		
Distance to the end of the block in the BCS Distance to the end of the block in the BCS (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acLookahTest (C, S)		$AC_LOOKAH_TEST[n]		
Test variable for LookAhead				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access and write access	

acSimMode (C, S)				
Simulation mode Variable $AC_SIM_MODE determines the simulation mode. The following values are possible: 0: Simulation not active. 1: Simulation mode active.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acSimTimeBlock (C, S)				
Block processing time in the simulation For the simulation: block processing time in seconds				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acSimTimeStep (C, S)				
Time step in the simulation For the simulation: time step in seconds				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acTrafoPar (C, S)		$AC_TRAFO_PAR[n]		
Parameter(s) of the current transformation Supplies the value of parameter 'n' of the current transformation, e.g. the cylinder diameter in the case of TRACYL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

actToolDataBeforeSearch (C, S)	$P_....._BEFORE_SEARCH_RUN	

Active tool offset data before the search

Data for determining the active tool offset before the search, that is in reset status before the start of the search.

After reaching the search target, the value is set to the current value for each programming of master tool holder, spindle, D no or DL no.

-P1: Master tool holder or spindle before the search ($AC_MTHNUM_BEFORE_SEARCH)

Programming "MTH(no)" or "MS(no)" after reaching the search target, this variable then returns the same value as acMthNum.

-P2: Active D no. before the search (see $P_D_BEFORE_SEARCH)

Programming "D" after reaching the search target, this variable then returns the same value as actDNumber.

-P3: Active DL no. before the search (see $P_DL_BEFORE_SEARCH)

Programming "DL" after reaching the search target, this variable then returns the same value as actDLNumber.

Units and value ranges

Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	

Addressing

Line index:	Max. line index:
	$$numSearchRunToolParams
	Read access

acsAxFrame (C, S)		

Current axial ACS frame

The variable acsAxFrame returns the axial frames of the active ACS (SZS) frames.

The variable has the following structure:

Row index	Description
1 - 2	Frame for the 1st channel axis
3 - 4	Frame for the 2nd channel axis
5 - 6	Frame for the 3rd channel axis
7 - 8	Frame for the 4th channel axis
9 - 10	Frame for the 5th channel axis
...	
39 - 40	Frame for the 20th channel axis

The axial frame consists of two values:

Double axSumTrans

Double axScale

For example, to read the first five channel axes, the row index must be entered as = 1 and the maximum row index as 10 (5 * 2).

The axial frames are returned for all configured channel axes without any axis gaps.

Note:

axSumTrans: Total translation

axScale: Scaling. In the event of a negative value, the axis is mirrored.

Units and value ranges

Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

Addressing

Line index:	Max. line index:
	40
	Read access

acsGeoFrame (C, S)		

Current ACS contour frame

The variable acsGeoFrame returns the current ACS contour frame.

The variable has the following structure:

Row index Description

1 - 3 Double geoSumTrans[GEODIM]

4 - 6 Double geoScale[GEODIM]

7 - 9 Double geoRot[GEODIM]

10 - 12 Double geoShear[GEODIM]

The maximum row index entered specifies how many values are to be read at any one time.

Example: Row index = 1 and the maximum index = 12: This enables the complete contour frame to be read (12 values).

Note:

GEODIM: 3 (number of geometry axes)

geoSumTrans: Total translation of the 1st to the 3rd geometry axis

geoScale: Scaling of the 1st to the 3rd geometry axis. In the event of a negative value, the axis is mirrored.

geoRot: Rotation around the 1st to the 3rd geometry axis. The MD has to determine whether the notation is RPY or Euler.

geoShear: Shear angle of the 1st to the 3rd geometry axis (the shear angle is 90 degrees in orthonormal coordinate systems)

Units and value ranges

Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

Addressing

Line index:	Max. line index:
	12
	Read access

actOriToolLengthENS (C, S)		

X, Y, Z tool components in the SZS

X, Y, Z components in the SZS of the active tool length, taking into account the tool orientation, including adapter data, mirrorings and TCARR

(orientable tool carrier).

Units and value ranges

Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

Addressing

Line index:	Max. line index:
	3
	Read access

actToolBasePosMcsPacked (C, S)				
Position of the tool holder in the MCS Position of a channel axis in the MCS with reference to the tool holder. The positions can be read for all configured channel axes without any axis gaps. Row index Corresponds to 1 1st configured channel axis 2 2nd configured channel axis .. 20 20th configured channel axis				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			MAXNUM_AXES_PER_CHAN	
			Read access	

actToolEdgeCenterPosBcsPacked (C, S)				
Position of the edge center in the BCS Position of a channel axis in the BCS with reference to the edge center. The positions can be read for all configured channel axes without any axis gaps. Row index Corresponds to 1 1st configured channel axis 2 2nd configured channel axis .. 20 20th configured channel axis				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			MAXNUM_AXES_PER_CHAN	
			Read access	

actToolEdgeCenterPosEnsPacked (C, S)				
Position of the edge center in the SZS Position of a channel axis in the SZS coordinate system with reference to the edge center. The positions can be read for all configured channel axes without any axis gaps. Row index Corresponds to 1 1st configured channel axis 2 2nd configured channel axis .. 20 20th configured channel axis				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			MAXNUM_AXES_PER_CHAN	
			Read access	

actToolEdgeCenterPosEnsPathn1 (C, S)				
Position of the TCP in the SZS TCP SZS positions of the path axes with path parameter = 1. Packed for all configured channel axes (without gaps). The positions do not have to correspond to the programmed positions (TRC, WAB, etc.) The positions can be read for all configured channel axes without any axis gaps. Row index Corresponds to 1 1st configured channel axis 2 2nd configured channel axis ... 20 20th configured channel axis				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			MAXNUM_AXES_PER_CHAN	
			Read access	

cmdToolEdgeCenterPosBcsPathn1 (C, S)				
End position of the TCP in the BCS TCP BCS positions of the path axes with path parameter = 1. Packed for all configured channel axes (without gaps). The positions do not have to correspond to the programmed positions (TRC, WAB, etc.) The positions can be read for all configured channel axes without any axis gaps. Row index Corresponds to 1 1st configured channel axis 2 2nd configured channel axis ... 20 20th configured channel axis				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			MAXNUM_AXES_PER_CHAN	
			Read access	

cmdToolEdgeCenterPosMcsPathn1 (C, S)				
Position of the TCP in the MCS TCP MCS positions of the path axes with path parameter = 1. Packed for all configured channel axes (without gaps). The positions do not have to correspond to the programmed positions (TRC, WAB, etc.) The positions can be read for all configured channel axes without any axis gaps. Row index Corresponds to 1 1st configured channel axis 2 2nd configured channel axis ... 20 20th configured channel axis				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			MAXNUM_AXES_PER_CHAN	
			Read access	

auxfu (C, S)		

Current auxiliary function

The variable is only used to read the auxiliary function during the output to the PLC.

This variable may only be used with the event AUXFU.

The variable has the following structure:
Double auxFuType;
Double extension;
Double value;

auxFuType is an integer value and can return the following values:
0: GARBAGE
1: M
2: M (generated by system)
3: H
4: S
5: S (generated by system)
6: T
7: D
8: DL
9: F

Value : floating point with S / F / H ($MC_AUXFU_H_TYPE_INT = 0)

Integer value with M / T / D / DL / H ($MC_AUXFU_H_TYPE_INT = 1)

Units and value ranges

Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

Addressing

Line index:	Max. line index:
	3
	Read access

axesActiveInChan (C, S)				
Bit mask of all active axes in the channel Bit mask of all active axes in the channel. There are many types of axes, e.g. geometry axis, positioning axis, command axis, PLC axis etc. In this mask, only those axes are marked which, in principle, can be interpolated by an interpolator in this channel. Bit Description 0 1. Channel axis 1 2. Channel axis ... 20 20. Channel axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

blockProgInfo (C, S)		$AC_BLOCK_PROGSTATE		
Information of active block in the interpolator Returns the information of a main run block.				
Return value	Bit-coded: Bit 0: Block is end of main program (M02, M17, M30 or RET(ASUB)) Bit 1: Block is end of subprogram Bit 2: Block is last initializing block			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

chanAxesIpoAtEnd (C, S)				
Channel axis has reached the end of its movement Bit field of channel axes have reached their end of movement.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		

chanAxesIpoAtEnd (C, S)				
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

speedCtrlSpindlesMask (C, S)				
Speed-controlled spindles in the channel Bit array of the speed-controlled spindles.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

cmdTrafoParS (C, S)		$P_TRAFO_PAR[n]		
Parameter(s) of the programmed transformation Supplies the value of parameter 'n' of the programmed transformation, e.g. the cylinder diameter in the case of TRACYL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

acDtew (C, S)		$AC_DTEW		
Distance to the end of the block in the WCS Distance to the end of the block in the WCS (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acEsrTrigger (C, S)	$AC_ESR_TRIGGER	
Activation of ESR Activation of "NC-controlled ESR"		

acEsrTrigger (C, S)		$AC_ESR_TRIGGER		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acFGo (C, S)		$AC_F_G0		
Max. rapid traverse rate in the block				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acFct0 (C, S)		$AC_FCT0[x] x = PolynomNo		
a0-coeff. nth polynominal for synchonous action a0-coefficient of the nth polynominal for the synchronous action SYNFCT / function FCTDEF n (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_FCTDEF_ELEMENTS	
			Read access	

acFct1 (C, S)		$AC_FCT1[x] x = PolynomNo		
a1-coeff. nth polynominal for synchonous action a1-coefficient of the nth polynominal for the synchronous action SYNFCT / function FCTDEF n (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_FCTDEF_ELEMENTS	
			Read access	

acFct2 (C, S)		$AC_FCT2[x] x = PolynomNo		
a2-coeff. nth polynominal for synchonous action a2-coefficient of the nth polynominal for the synchronous action SYNFCT / function FCTDEF n (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_FCTDEF_ELEMENTS	
			Read access	

acFct3 (C, S)		$AC_FCT3[x] x = PolynomNo		
a3-coeff. nth polynominal for synchonous action a3-coefficient of the nth polynominal for the synchronous action SYNFCT / function FCTDEF n (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_FCTDEF_ELEMENTS	
			Read access	

acFctll (C, S)		$AC_FCTLL[x] x = PolynomNo		
Lower limit nth polynominal for synchronous action Lower limit of the nth polynominal for the synchronous action SYNFCT / function FCTDEF n (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_FCTDEF_ELEMENTS	
			Read access	

acFctul (C, S)		$AC_FCTUL[x] x = PolynomNo		
Upper limit nth polynominal for synchronous action Upper limit of the nth polynominal for the synchronous action SYNFCT / function FCTDEF n (Note: for SYNACT only)				

acFctul (C, S)		$AC_FCTUL[x] x = PolynomNo		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MC_MM_NUM_FCTDEF_ELEMENTS	
			Read access	

aclwStat (C, S)		$AC_IW_STAT		
Current position of machine				
Return value	Bit-coded: Bit 0: Tool inv. position Bit 1: Axis 2/3 position Bit 2: Axis 5 position Bit 3-31: Not yet assigned			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aclwTu (C, S)		$AC_IW_TU		
Current position of channel axes				
Return value	Bit-coded: Bit 0: Channel axis 1 position Bit 1: Channel axis 2 position Bit 2: Channel axis 3 position Bit 3: Channel axis 4 position ...			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acJogCoord (C, S)		$AC_JOG_COORD		
Coordinate system manual travel Setting the coordinate system for the manual travel				
Return value	0: Work 1: SZS			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
			Read access and write access	

acLiftFast (C, S)		$AC_LIFTFAST		
Information about LIFTFAST Information about the execution of LIFTFAST. At the start of the LIFTFAST operation, the variable is set internally by the NC to the value "1". The variable must be reset to the initial state ($AC_LIFTFAST=0) by the evaluating program (if available), in order to be able to detect a subsequent LIFTFAST.				
Return value	0: Initial state 1: LIFTFAST has been executed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acMea (C, S)		$AC_MEA		
Touch probe has switched				
Return value	No. of touch probe			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

acMeasActPlane (C, S)		$AC_MEAS_ACT_PLANE		
Plane setting Plane setting for measurement calculation				
Return value	0: G17, 1: G18, 2: G19			

acMeasActPlane (C, S)		$AC_MEAS_ACT_PLANE		
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	2
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMeasChbfr (C, S)		$AC_MEAS_CHBFR		
Channel basic frame screen form Channel basic frame screen form for setting up the new frame				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
			Read access and write access	

acMeasChsfr (C, S)		$AC_MEAS_CHSFR		
System frame bit screen form System frame bit screen form for setting up the new frame				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
			Read access and write access	

acMeasCornerAngle (C, S)		$AC_MEAS_CORNER_ANGLE		
Corner cutting angle Calculated cutting angle of corner				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acMeasCornerSetangle (C, S)		$AC_MEAS_CORNER_SETANGLE		
Setpoint cutting angle User-selectable setpoint cutting angle of corner				
Return value	Permissible input range between 0 and 180 degrees			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
Degree, user defined	TYPE_DOUBLE		0	180.0

acMeasCornerSetangle (C, S)	$AC_MEAS_CORNER_SETANGLE	
Addressing		
Line index:	Max. line index:	
	1	
	Read access and write access	

acMeasDNumber (C, S)		$AC_MEAS_D_NUMBER		
Tool edge number Selected tool edge number				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMeasDiameter (C, S)		$AC_MEAS_DIAMETER		
Diameter Calculated diameter				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acMeasDirApproach (C, S)		$AC_MEAS_DIR_APPROACH		
Approach direction Approach direction towards workpiece				
Return value	0: +x 1: -x 2: +y 3: -y 4: +z 5: -z			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	5
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMeasFineTrans (C, S)		$AC_MEAS_FINE_TRANS		
Fine offset Correction in fine offset				
Return value	0: Correction in coarse translation 1: Correction in fine translation			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMeasFrameSelect (C, S)		$AC_MEAS_FRAME_SELECT		
Frame selection The frame calculated during workpiece measurement is entered in the selected frame.				
Return value	0: $P_SETFR 10.. 25: $P_CHBFR[0..15] 50.. 65: $P_NCBFR[0..15] 100.. 199: $P_UIFR[0..99] 1010..1025: $P_CHBFR[0..15] 1050..1065: $P_NCBFR[0..15]			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	1065
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMeasInput (C, S)		$AC_MEAS_INPUT[n]		
Data for measurement Data for the workpiece and tool measurement				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

acMeasLatch (C, S)		$AC_MEAS_LATCH		
Save measuring point Save measuring points for workpiece and tool measurement				
Return value	0: Clear measuring point 1: Write current axial values to measuring point			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			4	
			Read access and write access	

acMeasNcbfr (C, S)		$AC_MEAS_NCBFR		
Global basic frame screen form Global basic frame screen form for setting up the new frame				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
			Read access and write access	

acMeasP1Coord (C, S)		$AC_MEAS_P1_COORD		
Coordinate system of the 1st measuring point				
Return value	0: Work 1: BCS 2: MCS			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
			Read access and write access	

acMeasP2Coord (C, S)		$AC_MEAS_P2_COORD		
Coordinate system of the 2nd measuring point				
Return value	0: Work 1: BCS 2: MCS			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
			Read access and write access	

acMeasP3Coord (C, S)		$AC_MEAS_P3_COORD		
Coordinate system of the 3rd measuring point				
Return value	0: Work 1: BCS 2: MCS			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	

acMeasP3Coord (C, S)		$AC_MEAS_P3_COORD		
-	TYPE_DWORD	0	0	
			Read access and write access	

acMeasP4Coord (C, S)		$AC_MEAS_P4_COORD		
Coordinate system of the 4th measuring point				
Return value	0: Work 1: BCS 2: MCS			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
			Read access and write access	

acMeasPframe (C, S)		$AC_MEAS_PFRAME		
Programmable frame Programmable frame is not included				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
			Read access and write access	

acMeasResults (C, S)		$AC_MEAS_RESULTS[n]		
Measurement results				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

acMeasScaleunit (C, S)		$AC_MEAS_SCALEUNIT		
Unit of measurement Unit of measurement for input and output values				
Return value	0: Unit of measurement as configured 1: Unit of measurement in relation to active G code G70/G700/G71/G710			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMeasSema (C, S)		$AC_MEAS_SEMA		
Workpiece measurement semaphore Variable for disabling and enabling the measurement interface				
Return value	0: Not assigned 1: Assigned			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMeasSetCoord (C, S)		$AC_MEAS_SET_COORD		
Coordinate system of the set point				
Return value	0: Work 1: BCS 2: MCS			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
			Read access and write access	

acMeasTNumber (C, S)		$AC_MEAS_T_NUMBER		
Tool number Selected tool number				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMeasToolLength (C, S)		$AC_MEAS_TOOL_LENGTH		
Tool length Calculated tool length				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acMeasToolMask (C, S)		$AC_MEAS_TOOL_MASK		
Tool setting Tool setting for the measurement calculation				
Return value	Bit 0: Tool radius is not included in the calculation			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
			Read access and write access	

acMeasType (C, S)		$AC_MEAS_TYPE		
Measurement type Measurement type specification				
Return value	0: Default 1: x edge 2: y edge 3: z edge 4: Corner 1 5: Corner 2, 6: Corner 3 7: Corner 4 8: Hole 9: Shaft 10: Tool length 11: Tool diameter 12: Groove 13: Web 14: Actual value setting for geo and special axes 15: Actual value setting for special axes only 16: Edge_2P 17: Plane_Angles 18: Plane_Normal 19: Dimension_1 20: Dimension_2 21: Dimension_3			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	21
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMeasUifr (C, S)		$AC_MEAS_UIFR		
Settable data management frame Settable data management frame for setting up the new frame				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	99
			Read access and write access	

acMeasValid (C, S)		$AC_MEAS_VALID		
Validity bits Validity bits for measurement input values				
Return value	Bit 0: $AA_MEAS_POINT1[axis] Bit 1: $AA_MEAS_POINT2[axis] Bit 2: $AA_MEAS_POINT3[axis] Bit 3: $AA_MEAS_POINT4[axis] Bit 4: $AA_MEAS_SETPOINT[axis] Bit 5: $AC_MEAS_WP_SETANGLE Bit 6: $AC_MEAS_CORNER_SETANGLE Bit 7: $AC_MEAS_T_NUMBER Bit 8: $AC_MEAS_D_NUMBER Bit 9: $AC_MEAS_DIR_APPROACH Bit 10: $AC_MEAS_ACT_PLANE Bit 11: $AC_MEAS_FRAME_SELECT Bit 12: $AC_MEAS_TYPE Bit 13: $AC_MEAS_FINE_TRANS			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMeasWpAngle (C, S)		$AC_MEAS_WP_ANGLE		
Workpiece position angle Calculated workpiece position angle				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acMeasWpSetangle (C, S)		$AC_MEAS_WP_SETANGLE		
Setpoint workpiece position angle User-selectable setpoint workpiece position angle				
Return value	Permissible input range less than +/- 90 degrees			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
Degree, user defined	TYPE_DOUBLE		-90.0	90.0
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acMonMin (C, S)		$AC_MONMIN		
Actual tool monitoring value to the setpoint Ratio of the actual tool monitoring value to the setpoint				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acMsNum (C, S)		$AC_MSNUM		
Number of the master spindle Returns the number of the current master spindle				
Return value	o = no spindle available o = 1 ... n = number of the master spindle			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acMthNum (C, S)		$AC_MTHNUM		
Number of the current master tool holder Number of the current master tool holder. Is only meaningful with active magazine management.				
Return value	0: No master tool holder available 1..n: Number of the master tool holder			

acMthNum (C, S)		$AC_MTHNUM		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acOvr (C, S)		$AC_OVR		
Path override for synchronous actions Path override for synchronous actions (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPRTimeA (C, S)				
Program runtime - downtime In automatic mode: Exact recording of downtime. For simulation: Estimation of downtime in seconds.				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acPRTimeM (C, S)				
Program runtime - machining time In automatic mode: Exact recording of machining time. For simulation: Estimation of machining time in seconds.				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acPRTimeB (C, S)				
Machining time of the block In automatic mode: Exact recording of machining time of the block. For simulation: Estimation of machining time of the block in seconds.				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPathAcc (C, S)		$AC_PATHACC		
Path acceleration for real-time events				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
m/s2, 1000 inch/ s2, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPathJerk (C, S)		$AC_PATHJERK		
Path jerk for real-time events				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/s3, 1000 inch / s3, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPathn (C, S)		$AC_PATHN		
Normalized path parameter Normalized path parameter (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPlcOvr (C, S)		$AC_PLC_OVR		
PLC override for synchronized actions Path override for synchronized actions specified by the PLC				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	100	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPltbb (C, S)		$AC_PLTBB		
Path from beginning of the block in the BCS Path length from the beginning of the block in the BCS (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPlteb (C, S)		$AC_PLTEB		
Path to the end of the block in the BCS Path length to the end of the block in the BCS (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acProg (C, S)		$AC_PROG		
Program status according to $AC_PROG Program status (identical to progStatus but with coding that corresponds to $AC_PROG)				
Return value	0: aborted (reset) 1: halted (stop) 2: running (active) 3: waiting 4: interrupted			

acProg (C, S)		$AC_PROG		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPtpSup (C, S)				
Cart. PTP travel is supported Cartesian point-to-point travel (PTP) is supported by transformation				
Return value	0: Cart. PTP travel is not supported 1: Cart. PTP travel is supported			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acStat (C, S)		$AC_STAT		
Channel status according to $AC_STAT Channel status (identical to chanStatus but with coding that corresponds to $AC_STAT)				
Return value	0: reset 1: interrupted 2: active			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acSynaMem (C, S)		$AC_SYNA_MEM		
Free memory for synchronous actions Free memory for synchronous actions: Shows how many elements of the memory set with $MC_MM_NUM_SYNC_ELEMENTS are still free.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

acSynaMem (C, S)	$AC_SYNA_MEM	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

acSyncActLoad (C, S)		$AC_SYNC_ACT_LOAD		
Current runtime for synchronized actions Current runtime for synchronized actions of the last IPO cycle in the channel				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acSyncAverageLoad (C, S)		$AC_SYNC_AVERAGE_LOAD		
Average runtime for synchronized actions Average runtime for synchronized actions of an IPO cycle in the channel				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acSyncMaxLoad (C, S)		$AC_SYNC_MAX_LOAD		
Longest runtime for synchronized actions Longest runtime for synchronized actions of an IPO cycle in the channel				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acTaneb (C, S)		$AC_TANEB		
Tangent angle at the block end point				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	

acTaneb (C, S)	$AC_TANEB	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

acTc (C, S)		$AC_TC		
Active tool carrier				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acTcAckt (C, S)		$AC_TC_ACKT		
Trigger variable ACKnowledgeTrigger Trigger variable ACKnowledgeTrigger always assumes a value of 1 for an IPO cycle when the PLC acknowledges a command of the tool management.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acTcCmdt (C, S)		$AC_TC_CMDT		
Trigger variable: CoMmadTrigger Trigger variable: CoMmadTrigger always assumes a value of 1 for an IPO cycle when a new command of the magazine management is output to the PLC.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acThreadPitch (C, S)		$AC_THREAD_PITCH		
Programmed lead				
Units and value ranges				
Physical unit:	Data type:	Initial value		

acThreadPitch (C, S)		$AC_THREAD_PITCH		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acThreadPitchAct (C, S)		$AC_THREAD_PITCH_ACT		
Current lead				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acThreadPitchInc (C, S)		$AC_THREAD_PITCH_INC		
Current lead change				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acTime (C, S)		$AC_TIME		
Time from the beginning of the block in seconds Time from the beginning of the block in seconds (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
s	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acTimec (C, S)		$AC_TIMEC		
Time from begin. of block in interpolation cycles Time from the beginning of the block in interpolation cycles (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
IPO cycle	TYPE_DOUBLE			

acTimec (C, S)	$AC_TIMEC	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

acTimer (C, S)		$AC_TIMER[x] x = TimerNo		
Time variable in seconds Time variable in seconds (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
s	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_AC_TIMER	
			Read access	

acTotalOvr (C, S)		$AC_TOTAL_OVR		
Total path override for synchronized actions				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	100	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acTrafo (C, S)		$AC_TRAFO		
Code number of the active transformation Code number of the active transformation (encoded as for $AC_TRAFO)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acTrafoChain (C, S)		$AC_TRAFO_CHAIN		
Active chained transformation Active chained transformation Code numbers of the chained transformations of the active TRACON corresponding to machine data $MC_TRAFO_TYPE_m.				
Return value	0: No master tool holder available 1..n: Number of the master tool holder			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			4	
			Read access	

acTrafoParSet (C, S)		$AC_TRAFO_PARSET		
Number of current transformer data record The variable is '0' if no transformation is active. If a conventionally defined (i.e. not by kinematic chains) transformation is active, the variable contains the number of the current transformation data record. If a transformation defined by kinematic chains is active, the variable contains the number of the $NT data record with an offset of 1000, i.e. the first transformation returns the value 1001.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acVactB (C, S)		$AC_VACTB		
Path veloc. in basic coordinate system Path velocity in basic coordinate system				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acVactBf (C, S)		$AC_VACTBF		
Path velocity in the BCS Path velocity in the BCS. FGroup and FGREF are taken into account.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acVactWf (C, S)		$AC_VACTWF		
Path velocity in the workpiece coordinate system Path velocity in the workpiece coordinate system. FGroup and FGREF are taken into account.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acVactw (C, S)		$AC_VACTW		
Path velocity in the WCS (SYNACT) Path velocity in the work piece coordinate system (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acVc (C, S)		$AC_VC		
Add. path feed correction for synchronous actions Additive path feedrate correction value for synchronous actions (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

acVc (C, S)	$AC_VC	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

actDLNumber (C, S)		$P_DLNO		
Active total offset Number of active total offset DL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

actDNumber (C, S)		$P_TOOL		
Number of active tool edge Number of active tool edge (Dx). Value range: D0 ... DMax. Max corresponds to the value of MD18105 $MN_MM_MAX_CUTTING_EDGE_NO				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	9
Validity:	From NCK version 3.6		Read access	

actDNumberFanuc (C, S)				
Replaced by actDNumberFanuc32				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actDNumberFanuc32 (C, S)				
Offset memory number radius (ISO Dialect)) With programming in ISO Dialect mode: Offset memory number radius. Assigned only in conjunction with ISO Dialect M external language.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	

actDNumberFanuc32 (C, S)				
			1	
			Read access	

actDNumberS (C, S)				
Corresp. to actNumber for block search with calc. Corresponds to actNumber for block search with calculation Attention: This variable is available for protocolling block search events only, not for the Variable Service!				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

actDuploNumber (C, S)				
Duplo number of active tool				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

actFeedRateIpo (C, S)				
Actual value of the interpolation feedrate Actual value of the interpolation feedrate. The actual value is the feed actually moved with. (depends on the acceleration profiles, LookAhead, velocity limits etc.) The variable 'feedRateIpoUnit' defines the physical unit.				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access	

actFeedRateTechIpo (C, S)				
Interpolation feed extended, actual value Interpolation feed extended, actual value. The actual value is the actually traversed feed rate (depends on the acceleration profiles, LookAhead, velocity limits etc.). The variable 'feedRateIpoUnit' defines the physical unit (mm/min, mm/rev or mm/tooth).				
Units and value ranges				
Physical unit:	Data type:			

actFeedRateTechIpo (C, S)				
mm/min, inch/min, user defined	TYPE_DOUBLE			
			Read access	

actFrameIndex (C, S)		$P_UIFRNUM		
Index of the active set frame Index of the active set frame (index in G group 8 "Settable work offset"). Frames 0 - 4 (corresponds to G500 ... G57) can be set in the standard version. The number of frames can be changed via machine data MM_NUM_USER_FRAMES.				
Return value	0 = no frame selected 1 = G54 2 = G55 3 = G56 4 = G57 5 = G505 to 99 = G599			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

actHNumberFanuc (C, S)				
Replaced by actHNumberFanuc32				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actHNumberFanuc32 (C, S)				
Offset memory number length (Fanuc) With programming in ISO Dialect mode: Offset memory number length. Assigned only in conjunction with ISO Dialect M external language.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actIpoType (C, S)				
Active interpolation mode search run				
Active interpolation mode used for the path motion. This date corresponds to a large degree to the SNCF:ncFktBin for the first G-group. The value differs for automatically generated intermediate blocks only. This is e.g. the case if two lines are connected with an arc by the command RND. The value is the index of the active G-code (analog with SNCF:ncFktBin)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actIpoTypeS (C, S)				
Active interpolation mode				
Active mode of interpolation applied during block searches. This data is very similar to SNCF:ncFktBinS for the 1st G group. Its value is different only in the case of automatically generated intermediate blocks, such as when, for example, two straight lines are connected to an arc by means of command RND. The value is the index of the active G function (analogous to SNCF:ncFktBinS).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actMasterToolHolderNo (C, S)				
Active no. of the master tool holder				
Active number of the master tool holder. Especially for $MC_RESET_MODE_MASK, Bit0=0, this is the value of SETMS or SETMTH last programmed in the RESET status of the NCK. Especially for $MC_RESET_MODE_MASK, Bit0=1, this is the value in the RESET status of the NCK for $MC_SPIND_DEF_MASTER_SPIND (if $MC_TOOL_MANAGEMENT_TOOLHOLDER=0); or $MC_TOOL_MANAGEMENT_TOOLHOLDER (if $MC_TOOL_MANAGEMENT_TOOLHOLDER > 0)				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		1	max. Anzahl der Kana-lachsen
Addressing				
Line index:			Max. line index:	

actMasterToolHolderNo (C, S)				
			1	
			Read access	

actOriToolLength1 (C, S)				
X component of tool in WCS X component in workpiece coordinate system (WCS) of active tool length, taking into account the tool orientation, incl. adapter data, mirroring and TCARR (orientable toolholder).				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actOriToolLength2 (C, S)				
Y component of tool in WCS Y component in workpiece coordinate system (WCS) of active tool length, taking into account the tool orientation, incl. adapter data, mirroring and TCARR (orientation-capable toolholder).				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actOriToolLength3 (C, S)				
Z component of tool in WCS Z component in workpiece coordinate system (WCS) of active tool length, taking into account the tool orientation, incl. adapter data, mirroring and TCARR (orientation-capable toolholder).				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actParts (C, S)	$AC_ACTUAL_PARTS			
Actual number of workpieces Total number of workpieces machined since the start time. When the set number of workpieces ($AC_REQUIRED_PARTS) is reached, the counter is automatically reset to zero ($AC_REQUIRED_PARTS unequal to 0): The counter is incremented by 1 with the output of the M command defined in the channel MD 27882$MC_PART_COUNTER_MCODE[1] to the PLC. The counter is automatically reset with default values at control run-up. The timer is activated via MD27880 $MC_PART_COUNTER: o Bit 4 = 1: Counter $AC_TOTAL_PARTS is active Further meaning of bits 5-6 only if bit 4 =1: o Bit 5 = 0: Counter $AC_TOTAL_PARTS is incremented by 1 with VDI output of M02/M30 o Bit 5 = 1: Counter $AC_TOTAL_PARTS is incremented by 1 with output of M command from MD PART_COUNTER_MCODE[0] o Bit 6 = 0: $AC_TOTAL_PARTS also active with program test/block search o Bit 6 = 1: No machining $AC_TOTAL_PARTS with program test/block search.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

actProgNetTime (C, S)	$AC_ACT_PROG_NET_TIME			
Runtime of selected NC program The current net runtime of the selected program in [s], Net runtime means excluding the time during which the program was stopped. If a part program is restarted in AUTOMATIC mode from the RESET channel status, $AC_ACT_PROG_NET_TIME is automatically reset to "0". Further properties: o The Reset key does not reset $AC_ACT_PROG_NET_TIME to "0", but only stops the timer. o When an ASUB is started, $AC_ACT_PROG_NET_TIME is set to "0", and counts the runtime of the ASUB. At the end of an ASUB, it behaves the same as with the RESET key, that is the timer is only stopped, but not reset to "0". o When an event-controlled program is started (PROG_EVENT), $AC_ACT_PROG_NET_TIME is not reset. The program runtime only continues to be counted if it is a start, M30 or search run PROG_EVENT. o The behavior of $AC_ACT_PROG_NET_TIME with GOTOS and override = 0% can be parameterized by MD27850.				
Return value	Seconds			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	

actProgNetTime (C, S)	$AC_ACT_PROG_NET_TIME	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

actMTNumber (C, S)				
Number of the multitool with active tool				
Number of the multitool in which the active tool is contained. The value is zero if the active tool is not contained in a multi-tool.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actMTPlaceNumber (C, S)	$AC_TC_	
Number of the MT place with active tool		
Number of the multitool place in which the active tool is contained. The value is zero if the active tool is not contained in a multitool.		
Units and value ranges		
Physical unit:	Data type:	
-	TYPE_UWORD	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

actTNumber (C, S)	$P_TOOLNO		W1
Active (internal) tool number			
The command should not be used if magazine management (TMMG=1) is active. If magazine management is active, GETTEXET should be used instead. For $MC_CUTTING_EDGE_DEFAULT=0 an incorrect T number may be determined. If the programming is made after programming D > 0, it is also safe			
Units and value ranges			
Physical unit:	Data type:	Minimum value:	Maximum value:
-	TYPE_UWORD	0	32000
Validity:	From NCK version 3.6	Read access	

actTNumberLong (C, S)		
No. of active tool for flat D-no. max 8 digits		
Number of the active tool using flat D-numbers with up to 8 digits		

actTNumberLong (C, S)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actTNumberS (C, S)				
corr. to actTNumber for block search without calc. Corresponds to actTNumber for block search with calculation. Attention: This variable is available for protocolling the block search events only, not for the Variable Service!				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

actToolEdgeCenterPosEns (C, S)				
composite variable Corresponds to $$/C/SEGA/actToolEdgeCenterPosEns in the SEGA block for the 3 geometry axes The variable consists of three values of the DOUBLE type, i.e. is 24 bytes long. Note: The variable is released for use only with the Simultaneous recording function.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actToolIdent (C, S)				W1
Identifier of active tool				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	"\0"		
Addressing				
Line index:			Max. line index:	

actToolIdent (C, S)			W1
		1	
Validity:	From NCK version 3.6	Read access	

actToolLength1 (C, S)		$P_TOOLL[1]		W1
Active tool length 1 Active tool length 1. The value corresponds to the value in $TC_DP3 including the active tool offsets, but irrespective of the plane selection and setting data such as SD42940 $SC_TOOL_LENGTH_CONST				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access	

actToolLength2 (C, S)		$P_TOOLL[2]		W!
Active tool length 2 Active tool length 2. The value corresponds to the value in $TC_DP4 including the active tool offsets, but irrespective of the plane selection and setting data such as SD42940 $SC_TOOL_LENGTH_CONST				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access	

actToolLength3 (C, S)		$P_TOOLL[3]		W1
Active tool length 3 Active tool length 3. The value corresponds to the value in $TC_DP5 including the active tool offsets, but irrespective of the plane selection and setting data such as SD42940 $SC_TOOL_LENGTH_CONST				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access	

actToolRadius (C, S)		$P_TOOLR		W1
Active tool radius Current total radius of the tool. The total radius includes all components, such as geometry radius ($TC_DP6), wear radius ($TC:DP15) and online offset values.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access	

actTransform (C, S)				
Active transformation				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	\0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actWaCSCoordSys (C, S)		$AC_WORKAREA_CS_COORD_SYSTEM		
Coordinate system for working area limitation Coordinate system of the active coordinate system-specific working area limitation				
Return value	Identifier for the coordinate system in which the working area limitation is to apply. The following are valid: 0: Working area limitation in the WCS 3: Working area limitation in the SZS			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actWaCSLimitMinus (C, S)		$AC_WORKAREA_CS_LIMIT_MINUS		
Coord.-specific working area limitation, negative Position of the coordinate system-specific working area limitation in the negative direction for the addressed axis and working area group.				
Return value	Position of the working area limitation in the negative direction			
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actWaCSLimitPlus (C, S)		$AC_WORKAREA_CS_LIMIT_PLUS		
Coord.-specific working area limitation positive Position of the coordinate system-specific working area limitation in the positive direction for the addressed axis and working area group				
Return value	Position of the working area limitation in the positive direction			

actWaCSLimitPlus (C, S)		$AC_WORKAREA_CS_LIMIT_PLUS		
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actWaCSMinusEnable (C, S)		$AC_WORKAREA_CS_MINUS_ENABLE		
Coord.-spec. work. area limitation, negative valid The coordinate-specific working area limitation in the negative direction of $$actWaCSLimitMinus is valid.				
Return value	TRUE: The value in the variable $$actWaCSLimitMinus is valid for the axis. FALSE: There is no limitation in the coordinate-specific working area in the negative direction for this axis.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actWaCSPlusEnable (C, S)		$AC_WORKAREA_CS_PLUS_ENABLE		
Coord.-spec. work. area limitation, positive valid The coordinate system-specific working area limitation in the positive direction of $$actWaCSLimitPlus is valid.				
Return value	TRUE: The value in the variable $$actWaCSLimitPlus is valid for the axis. FALSE: There is no limitation for this axis in the positive direction in the coordinate-specific working area.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

allAxesRefActive (C, S)		DB21-30, DBX36.2		
All axes referenced Code specifying whether all axes are referenced				
Return value	1 = all axes referenced 0 = at least 1 axis not referenced			
Units and value ranges				
Physical unit:	Data type:			

allAxesRefActive (C, S)		DB21-30, DBX36.2		
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

allAxesStopped (C, S)				
Axes in exact stop Code specifying whether axes are in exact stop				
Return value	0 = at least one axis is not in exact stop 1 = All axes in exact stop			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

basisFrameMask (C, S)		$P_CHBFRMASK		
Active channel-specific basic frames Display indicating which channel-specific basic frames are active				
Return value	Every bit in the mask indicates whether the appropriate basic frame is active. Bit0 = 1st basic frame, Bit1 = 2nd basic frame, etc.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

blockType (C, S)		$AC_BLOCKTYPE		
Type of the active block in the interpolator Identifies the type of a block (programmed or generated internally)				
Return value	0: No internally generated block 1: Internally generated block, but cannot be specified in detail 2: Block was generated on chamfering/rounding 3: Smooth approach and retraction (SAR) 4: Block was generated during tool offset 5: Block was generated on smoothing 6: Block was generated by TLIFT (tangential correction) 7: Block was generated during path segmentation 8: Block was generated by compile cycles			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	8
Addressing				
Line index:			Max. line index:	

blockType (C, S)	$AC_BLOCKTYPE	
	1	
	Read access	

blockTypeInfo (C, S)	$AC_BLOCKTYPEINFO	
Information on block type Detailed information on block type		

blockTypeInfo (C, S)	$AC_BLOCKTYPEINFO	
Return value	The value range and the meaning of this variable depend on the current value of system variable $$blockType System variable $$blockTypeInfo can be used to request additional information on variable $$blockType. Depending on the value of system variable $$blockType, various values are then possible: 1. General internally generated block: $$blockType = 1 $$blockTypeInfo = 1000 and does not include any additional information. 2. Chamfer/round: $$blockType = 2 2001: straight 2002: circle 3. SAR: $$blockType = 3 3001: Approach with straight 3002: Approach with quadrant 3003: Approach with semicircle 4. Tool offset: $$blockType = 4 4001: Approach block after STOPRE 4002: Link blocks if intersection not found 4003: Point circle on inside corners (only with TRACYL) 4004: Bypass circle (or conic) on outside corners 4005: Approach blocks for offset suppression 4006: Approach blocks for reactivation of TRC 4007: Blocks are split because curvature is too high 4008: Compensation blocks for 3D front milling (tool vector \|\| plane vector) 5. Corner rounding: $$blockType = 5 5001: Rounding contour through G641 5002: Rounding contour through G642 5003: Rounding contour through G643 5004: Rounding contour through G644 6. TLIFT: $$blockType = 6 6001: TLIFT block with linear movement of the tangential axis but without retraction movement. 6002: TLIFT block with non-linear movement of the tangential axis (polynomial) but without retraction movement. 6003: TLIFT block with retraction movement. Tangential axis movement and retraction movement start simultaneously.	

blockTypeInfo (C, S)		$AC_BLOCKTYPEINFO		
	6004: TLIFT block with retraction movement. Tangential axis does not start until a specific retraction position is reached. 7. Path segmentation: $$blockType = 7 7001: Programmed path segmentation without punching or nibbling active. 7002: Programmed path segmentation with punching or nibbling active. 7003: Automatically generated path segmentation. 8. Compile cycles: $$blockType = 8 In this case, system variable $AC_BLOCKTYPEINFO includes the ID of the compile cycles application that generated the block.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

cIn (C, S)		$C_IN[n]		
Signal from PLC to cycle Signal from PLC to cycle (reserved for SIEMENS application, e.g. ShopMill/ManualTurn)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

cOut (C, S)		$C_OUT[n]		
Signal from cycle to PLC Signal from cycle to PLC (reserved for SIEMENS application, e.g. ShopMill/ManualTurn)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

chanAlarm (C, S)		DB21-30, DBX36.6 und DBX36.7		A2
NCK alarm pending Code whether NCK alarm pending.				
Return value	0 = no alarm in this channel 1 = alarm without stop 2 = alarm with stop			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

chanAxisNoGap (C, S)				
Alternative name:	axisActivInChan			
Display of existing axes. Display of existing axes, i.e. no axis gap in channel.				
Return value	Bits 0-31 represent the axes of the channel. Bitn = 0: Axis does not exist. Bitn = 1: Axis does exist.			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

chanStatus (C, S)		DB21-30, DBX35.5 - DBX35.7		K1
Channel status (1=active) Channel status				
Return value	0 = RESET 1 = active 2 = interrupted			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

changeAxConfCounter (C, S)				
Counter indicating changes of axes configuration A counter which is incremented if the axes configuration has changed. This is the case, if e.g. geometry axes are switched or axes have been changed between channels. The counter is set to 0 at PowerOn and it might overflow. You cannot be sure, that the axes configuration actually has changed when the counter is incremented.				
Units and value ranges				
Physical unit:	Data type:			

changeAxConfCounter (C, S)				
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

cmdDwellTime (C, S)				
Programmed dwell time				
Return value	See $$timeOrRevolDwell			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

cmdFeedRateIpo (C, S)		$AC_F		
Desired value of the interpolation feedrate Desired feedrate of the interpolation feedrate. The physical unit is defined in the variable 'feedRateIpoUnit'				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access	

cmdFeedRateIpoS (C, S)				
Interpolation feed during search Interpolation feed during search. The physical unit is in the variable $$feedRateIpoUnitS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

cmdTrafoParSetS (C, S)		$P_TRAFO_PARSET		
No. prog. transformation data record block search The variable is '0' if no transformation is active. If a conventionally defined (i.e. not by kinematic chains) transformation is active, the variable contains the number of the current transformation data record during block search. If a transformation defined by kinematic chains is active, the variable contains the number of the $NT data record with an offset of 1000, i.e. the first transformation returns the value 1001 (during block search).				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

cmdTrafoS (C, S)		$P_TRAFO		
Prog. transformation block search Code number of programmed transformation for block search				
Return value	Coding as for variable $AC_TRAFO			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

contourDev (C, S)				
Do not use this variable any more!				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access	

corrBlActive (C, S)				
Incorrect block occurred Incorrect block has occurred (correction block)				
Return value	0 = no incorrect block 1 = incorrect block			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

cycServRestricted (C, S)				
Restricted cyclic variable service Identifier, whether there is restricted, cyclic variable service. This is a privileged variable: cyclic result acknowledgments of this variable are also returned when the cyclic services are no longer served by the NCK because of block change time problems. Notice: The privileged variables lose his property if they are mixed with non-privileged variables in one order. -> Do not integrate the variable cycServRestricted in a cluster with other variables!				
Return value	0 = normal cycl. service 1 = no cyclic service (but acknowledgment)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

delObjState (C, S)				
Error status of PI _N_DELOBJ Using PI _N_DELOBJ protection areas are deleted analogous to the language command DELOBJ(...). This OPI variable shows the status of the PI . 0 = PI successfully executed. -2 = Name of the object to be deleted not known. -3 = Index -1 not permitted -4 = Start index too large -5 = Illegal index when deleting a group (only -1 permitted) -6 = Start index smaller than end index -7 = End index too large				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

delayFSt (C, S)				
Stop in curr.progr.area is effect.after a delay Delay Feed Stop, Stop is delayed in the current program area				
Return value	0: Stop in the current program area is effective immediately 1: Stop in the current program area is effective after a delay 2: Stop in the current program area is effective immediately (same as 0), although a stop delay area was programmed in the parts program. (This means that the NCK could not switch on the stop delay area.)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

delayFSt (C, S)				
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

direction (C, S)				
Traversing direction				
Return value	0 = normal travel 1 = forward travel 2 = reverse travel 3 = reference point cycle 4 = stop state			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

ensRotTrans (C, S)				
ensRotTrans SZS offset of the rotary axis with TRANSMIT and TRACYL				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

extProgActive (C, S)		DB21-30, DBX32.0		
Program execution from external active Flag indicating whether program execution from external is active.				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

feedRateIpoOvr (C, S)				
Interpolation feedrate, override				
Units and value ranges				
Physical unit:	Data type:			

feedRateIpoOvr (C, S)				
%	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access	

feedRateIpoUnit (C, S)				
Interpolation feedrate, units				
Return value	0 = mm/min 1 = mm/rev 2 = inch/min 3 = inch/rev			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

feedRateIpoUnitS (C, S)				
Interpolation feedrate, units (search) Interpolation feed, units during search				
Return value	0 = mm/min 1 = mm/rev 2 = inch/min 3 = inch/rev			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

findBlActive (C, S)	DB21-30, DBX33.4			K1
Block search active Code whether block search is active.				
Return value	0 = not active 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

geoAxesInChan (C, S)				
Assignment of geometry axes to channel axes Determines the geometry axis assignment of the channel axes. Row index 1: Channel axis index of the 1st geometry axis Row index 2: Channel axis index of the 2nd geometry axis Row index 3: Channel axis index of the 3rd geometry axis In the event of a non-existent geometry axis the channel axis index is -1.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

ludAccCounter (C, S)				
Counter indicating that a new LUD ACC is available Counter indicating that a new LUD ACC is available. If subprograms are called during an automatic program execution, a new set of LUDs becomes valid. In order to indicate to the HMI that it has to modify the display of the LUDs, respectively that the validity of the LUDs has changed, the variable 'ludAccCounter' is incremented. It is only necessary for the HMI to inquire a change of the variable's value, the value itself is of no importance.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

machFunc (C, S)		DB11, DBXn.0 - DBXn.2 (n=7,27,47,67, ...)		
Active channel machine function				
Return value	0 = none 1 = REPOS 2 = TEACH IN 3 = REF 4 = TEACH-REPOS 5 = TEACH-REF			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

markActiveList (C, S)				
Status array for the wait marker Status array for the active marker in channel m. The first element (markActiveList[1]) of the array specifies the currently active marker number of this channel (channel m). The second element (markActiveList[2]) specifies bit-coded whether channel m is still waiting for the mark to be reached in the other channels (channel n), in short "waiting status". markActiveList[2] Bit-n == 1 Channel m is waiting for mark markActiveList[1] in channel n markActiveList[2] Bit-n == 0 Channel n has already reached mark markActiveList[1], or channel m is not waiting for mark markActiveList[1] at all				
Return value	markActiveList[1] == 0 Current channel m does not edit any wait marker markActiveList[1] == 1..99 Current channel m is positioned on the wait marker with markActiveList[1] markActiveList[2] Bit-n == 1 Channel m is waiting for mark markActiveList[1] in channel n markActiveList[2] Bit-n == 0 Channel n has already reached mark markActiveList[1], or channel m is not waiting for mark markActiveList[1] at all			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	99
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

nameIndex (C, S)				
Index of the found string Using PI _N_NAMINT (NAMETOINT) string is searched in a 1-dimensional string field. In this OPI variable, the index of the found strings is returned to the string field. If the string is not found, the OPI variable is -1.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

ncProgEndCounter (C, S)		
Counter for end of program Counter which is incremented as soon as the NCK has processed an end of program.		

ncProgEndCounter (C, S)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

ncResetCounter (C, S)				
Counter for Reset key Counter which is incremented with each 0->edge of the Reset key				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

ncStartCounter (C, S)				
Counter for NC-start key Counter for the NC-start key. Pressing this key increments the variable 'ncStartCounter'. The value of the variable can be ignored, the HMI must just inquire the change of the variable to see whether the start-key has been pressed.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

ncStartSignalCounter (C, S)				
Counter for NC start signal Counter that is incremented as soon as the channel-specific NC start signal has been activated in the VDI interface.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numChanAlarms (C, S)				
Do not use this variable any more!				

numChanAlarms (C, S)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numToolHolders (C, S)		$P_MAGNS		
Number of tool holders in the TOA of the channel Number of tool holders/ spindles (buffer locations of the location type = spindle) from the magazine configuration of the TOA which are allocated to the channel. The number of tool holders / spindles is solely a function of the magazine configuration and does not change during an NC program execution. Value = 0, if there is no magazine configuration or the TMMG (tool management magazines) is not present in the NC.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$numMachAxes
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numTraceProtocEventType (C, S)				
Logging: Standard event types Logging: Number of standard event types				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_UWORD		0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

numTraceProtocOemEventType (C, S)		$MM_PROTOC_NUM_ETP_OEM_TYP		
Logging: OEM event types Logging: Number of OEM event types				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_UWORD		0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

oldProgNetTime (C, S)		$AC_OLD_PROG_NET_TIME		
Net runtime of last NC program Net runtime of the program that has just finished correctly in seconds. "Finished correctly" means that the program was not canceled with RESET, but ended normally with M30. If a new program is started, $AC_OLD_PROG_NET_TIME remains unaffected until M30 is reached again. Further properties: o $AC_OLD_PROG_NET_TIME is set to "0" if the currently selected program is edited. o At the end of an ASUB or an event-controlled program (PROG_EVENT), $AC_OLD_PROG_NET_TIME is not changed.				
Return value	Seconds			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

oldProgNetTimeCounter (C, S)		$AC_OLD_PROG_NET_TIME_COUNT		
Modification counter for oldProgNetTime oldProgNetTimeCounter is zero in the Power ON status. oldProgNetTimeCounter is always increased when $$oldProgNet-Time has been newly written. This enables the user to ensure that $$oldProgNetTime has been written. This means that, if the user cancels the current program with reset, $$oldProgNetTime and ldProgNetTimeCounter remain unchanged. Note: Two programs running consecutively can have identical runtimes and be correctly terminated. The user can then only detect this by the changed oldProgNetTimeCounter.				
Return value	Modification counter			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pCutInv (C, S)		$AC_CUT_INV		
Invert direction of spindle rotation tool States that a turning tool is rotated against the machining plane (typically through 180 degrees around the C axis with G18) such that the direction of spindle rotation has to be inverted.				
Return value	FALSE, TRUE			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

pCutInv (C, S)		$AC_CUT_INV		
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pCutInvS (C, S)				
Invert dir. of spindle rot. tool (block search) States that a turning tool is rotated against the machining plane (typically through 180 degrees around the C axis with G18) such that the direction of spindle rotation has to be inverted. For block search.				
Return value	FALSE, TRUE			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pCutMod (C, S)		$AC_CUTMOD		
Last programmed value of CUTMOD Reads the current valid value that was last programmed with the language command CUTMOD (number of the tool carrier for which the cutting edge data modification is to be activated). If the last programmed value was CUTMOD = -2 (activation with the currently active tool carrier with orientation capability), the value -2 is not returned but the number of the active tool carrier with orientation capability at the time of programming.				
Return value	-2, 999999			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pCutModS (C, S)				
Last programmed value of CUTMOD (block search) Reads the current valid value that was last programmed with the language command CUTMOD (number of the tool carrier for which the cutting edge data modification is to be activated). If the last programmed value was CUTMOD = -2 (activation with the currently active tool carrier with orientation capability), the value -2 is not returned but the number of the active tool carrier with orientation capability at the time of programming. For block search				
Return value	-2, 999999			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pEgBc (C, S)		$P_EG_BC[a]		
EG block change criterion Electronic gear: Block change criterion. Important for EGON, EGONSYN				
Return value	0: NOC Block change is performed immediately 1: IPOSTOP Block change is performed with setpoint side synchronism 2: COARSE Block change is performed with "Synchronism coarse" 3: FINE Block change is performed with "Synchronism fine"			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	3	0	3
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

pMthSDC (C, S)		$P_MTHSDC		
Master tool holder for cutting edge selection Master tool holder no. or master spindle no. is determined with reference to the active tool for the next D offset selection. This is important if the master spindle changes after the last tool change.				
Return value	>0 Successful read access 0 No master tool holder or no master spindle available. The next D offset works with T0. -1 TMMG not available			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	$$numMachAxes

pMthSDC (C, S)		$P_MTHSDC		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pOffn (C, S)		$P_OFFN		
Offset normal Last programmed offset normal				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access	

pTCutMod (C, S)		$P_AD[2]		
Angle of rotation for edge position Angle of rotation for modification of edge position and cutting direction				
Return value	Angle between 0 and 360 degrees			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
Degree	TYPE_DOUBLE	0	0	360
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pTCutModS (C, S)				
Angle of rotation for block search edge position Angle of rotation for edge position and cutting direction for block search				
Return value	Angle between 0 and 360 degrees			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
Degree	TYPE_DOUBLE	0	0	360
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pTc (C, S)		$P_TC		
Active tool carrier with orientation capability The active orientatable toolholder				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	

pTc (C, S)		$P_TC		
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pTcAng (C, S)		$P_TCANG[n]		
Angles of axes for toolholders The current angles of the two axes of an orientation-capable toolholder				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

pTcDiff (C, S)		$P_TCDIFF[n]		
Angular difference between toolholder axes The difference between the exact and the actually used angles of the two axes of an orientation-capable toolholder				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

pTcNum (C, S)		$P_TCNUM		
Number of available tool carriers Number of available orientable tool carriers in the channel				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pTcSol (C, S)		$P_TCSOL		
Toolholder number of solutions Number of solutions (configuration options for rotary axes) on selection of an orientatable toolholder. The variable value can be between 0 and 2, where 0 to 2 means either none, 1 solution or 2 solutions.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pTcStat (C, S)		$P_TCSTAT		
Tool carrier status Specifies the status of an orientable tool carrier.				
Return value	The variable is bit-coded with the following meanings: 0x0001 The first rotary axis is available 0x0002 The second rotary axis is available 0x0004 The angles used for the calculation come from an orientation in the frame direction 0x0008 The angles used for the calculation have been absolutely defined 0x0010 The polar axis angle is not defined in the case of orientation in the frame direction 0x1000 Only the tool can be rotated (kinematic type T) 0x2000 Only the workpiece can be rotated (kinematic type P) 0x4000 Tool and workpiece can be rotated (kinematic type M) The bits stated here are not currently assigned.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pToolO (C, S)		$P_TOOL_O		
Tool orientation Supplies the current tool orientation in various coordinate systems: Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS The orientation vector is normalized, i.e. it has the value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

paAccLimA (C, S)		$PA_ACCLIMA[a]		
Axial acceleration override: run in Axial acceleration override in run in				
Return value	1-200			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	100	1	200
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

paJerkLimA (C, S)		$PA_JERKLIMA[a]		
Axial jerk override: run in Axial jerk override in run in				
Return value	1-200			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	100	1	200
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

paVeloLimA (C, S)		$PA_VELOLIMA[a]		
Axial velocity override: run in Axial velocity override in run in				
Return value	1-200			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	100	1	200
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actDLNumberS (C, S)				
Alternative name:	progDLNumberS			
Active total offset for block search Corresponds to actDLNumber for block search with calculation Caution: This variable is not available for the Variable Service, but only for logging in the case of block search events!.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

gccState (C, S)		$PC_GCC_STATE		
Status of the G code converter The variable shows the internal status of the G code converter. Status = 0 -> The G code converter is not selected. Status = 1 -> The G code converter is selected by HMI, but a trace has not yet been generated. Status = 2 -> The G code converter is active (after NC START), the trace is being output. Status = 3 -> The G code converter is active but interrupted by the language command GCCDISABLE, there is no output into the trace file.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

progDuploNumber (C, S)				
Duplo number of programmed tool) Duplo number of programmed tool (does not yet have to be active)				

progDuploNumber (C, S)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

progNetTimeTrigger (C, S)	$AC_PROG_NET_TIME_TRIGGER			
Trigger for time measurement progNetTimeTrigger is used to selectively measure program sections. This means that writing progNetTimeTrigger enables the program to switch time measurement on and off. In order to exploit all trigger possibilities, certain values for progNetTimeTrigger are assigned a special function: 0 = Neutral: The trigger is not active, the value is taken from Reset with the Start key. 1 = Exit: Exits the measurement and copies $$actProgNetTime to $$oldProgNetTime. $$actProgNetTime is set to zero, and then starts running again. 2 = Start: Starts the measurement and sets $$actProgNetTime to zero. $$oldProgNetTime is not changed. 3 = Stop: Stops the measurement, does not change $$oldProgNetTime, and holds $$actProgNetTime constant until continue. 4 = Continue: Continuation of the measurement. This means that a measurement that has been stopped is resumed. $ $actProgNetTime continues to run. $$oldProgNetTime is not changed.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	4
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

progStatus (C, S)	DB21-30, DBX35.0 - DBX35.4			K1
Program status				
Return value	1 = interrupted 2 = stopped 3 = in progress 4 = waiting 5 = aborted			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

progTNumber (C, S)		
Number of programmed tool		

progTNumber (C, S)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

progTNumberLong (C, S)				
No. progr. tool for flat D-no. with max. 8 digits Number of the programmed tool using flat D-numbers with up to 8 digits				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

progToolIdent (C, S)				
Identifier programmed tool Identifier of programmed tool (does not yet have to be active)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	"\0"		
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

progWaitForEditUnlock (C, S)				
NCK is waiting for the release of an NC program The variable is used for two possible applications: 1. Notification to HMI that an NC program is to be processed for which a program execution delay was activated by means of the PI service _N_F_MODE. This is only possible for files of the passive file system of the NCK. 2. Notification to HMI that an NC program is to be processed for which a WRITE lock was set. This is only possible for files that are available on the CF, network drive or USB device and are processed in EES mode. The variable contains the complete path name in both cases.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	

progWaitForEditUnlock (C, S)				
			1	
			Read access	

protAreaCounter (C, S)				
Protection zone modification counter Counter is incremented by 1 every time a protection zone (block PA) is modified				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

protocHmiEvent (C, S)				
Activate event in preprocessing Logging: When writing, the defined event is activated during preprocessing.				
Return value	49: HMI_TRIG_1 50: HMI_TRIG_2 51: HMI_TRIG_3			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_UWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

protocUserActive (C, S)		$MM_PROTOC_USER_ACTIVE		
Logging: Active users Logging: Displays active users				
Return value	0: User inactive 1: User active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

rapFeedRateOvr (C, S)				
Rapid traverse override				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access	

remainDwellTime (C, S)				
Remaining dwell time				
Return value	See $$timeOrRevolDwell			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

reqParts (C, S)	$AC_REQUIRED_PARTS			
Setpoint number of workpieces Number of workpieces to be machined (setpoint workpiece count) The number of workpieces can be defined in this counter, when this number is reached the actual workpiece count ($AC_ACTUAL_PARTS) is reset to "0".				
Return value	o Bit 0 = 1: Counter $AC_REQUIRED_PARTS is active Further meaning of bit 1 only if o bit 0 = 1: o Bit 1 = 0: Alarm/VDI output if $AC_ACTUAL_PARTS equals $AC_REQUIRED_PARTS o Bit 1 = 1: Alarm/VDI output if $AC_SPECIAL_PARTS equals $AC_REQUIRED_PARTS			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

rotSys (C, S)	$AC_ROT_SYS			
Orientation reference system Reference system for orientation movements with cartesian manual traversal				
Return value	0: Axis-specific manual traversal active 1: Cartesian manual traversal in basic coordinate system active 2: Cartesian manual traversal in workpiece coordinate system active 3: Cartesian manual traversal in tool coordinate system active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3

rotSys (C, S)	$AC_ROT_SYS	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

seruproMasterChanNo (C, S)				
Block search waits for partner channel The search type SERUPRO (search via program testing) may be started simultaneously in several channels in order to start a channel grouping correctly. A search target must be specified in one channel (master channel) in the grouping. The other channels do not need a search target, they wait until they have reached a stop condition and the master channel has reached the search target. These channels generally stop at WAIT marks. The variable seruproMasterChanNo defines the master channel.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$numChannels
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

seruproMasterNcuNo (C, S)				
Partner channel NCU, block search waits The search type SERUPRO (search via program testing) may be started simultaneously in several channels in order to start a channel grouping correctly. A search target must be specified in one channel (master channel) in the grouping. The other channels do not need a search target, they wait until they have reached a stop condition and the master channel has reached the search target. These channels generally stop at WAIT marks. The variable seruproMasterChanNo defines the master channel. seruproMasterNcuNo specifies the master channel in more detail if it is not on the active NCU.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$MN_MM_LINK_NUM_OF_MODULES
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

simSpiDat (C, S)				
Spindle data The variable simSpiDat is a double array of data with the following structure: Double spino ; Number of the spindle for which the event was triggered Double masterSpiNo ; Number of the leading spindle Double pSMode ; Programmed spindle mode Double turnState ; Rotation state of the spindle: 0 = right, 1 = left, 2 = stop Double coupPosOffset ; Position offset of an axis to the leading axis / leading spindle (actual value)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			5	
			Read access	

simulationSupport (C, S)				
simulationSupport Block information for the support of the JobShop simulation				
Return value	Bit0: Change of the transformation in the curr. block Bit1: Change of the frame in the curr. block Bit2: Curr. block in an action block Bit3: Curr. block in the last action block Bit4: Curr. block has PTP active (as of 510600)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

simulationSupportS (C, S)				
Support of the JobShop simulation during search Block information for the support of the JobShop simulation during search				
Return value	Bit0: - Bit1: - Bit2: - Bit3: - Bit4: Curr. block has PTP active			

simulationSupportS (C, S)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

specParts (C, S)		$AC_SPECIAL_PARTS		
Number of special workpieces (user-defined) Number of current workpieces as defined by user: This counter enables the user to define his own workpiece count. The counter is reset to zero automatically only when the control system boots on defaults.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

splitBlock (C, S)		$AC_SPLITBLOCK		
Identifies splitted blocks Identifier of internally splitted blocks				
Return value	0: A BLOCK programmed unchanged (a BLOCK generated by the compressor is regarded as programmed BLOCK): <>0: BLOCK was shortened or is an internally generated BLOCK; the following values are possible: 1: It is an internally generated BLOCK or a shortened original BLOCK 3: It is the last block in a chain of internally generated blocks or shortened original blocks.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

startLockState (C, S)				
Status of the global start disable Status of the global start disable. Also see PI _N_STRTLK and _N_STRTUL.				
Return value	0: No start disable 1: Start disable is switched on and program is not running 2: Start disable is switched on and program is running nevertheless The NCK changes from 2->1 as soon as the program is stopped.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

startRejectCounter (C, S)				
Counter for rejected NC starts Counter that is incremented as soon as an NC start is rejected due to a global start disable (see _N_STRTLK), program-specific start disable (see _N_F_STLO), or channel-specific start disable (see _N_STRTLK).				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

stopCond (C, S)				
Replaced by stopCondNew Replaced by $$stopCondNew				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

stopCondNew (C, S)				
Number of the stop state of the NC Number of the NC stop state More than one stop state can be active simultaneously. The highest priority stop state appears below the first line, this is followed by those with lower priorities. The documentation explains the meanings of the individual stop states.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$stopCondNum	
			Read access	

stopCondPar (C, S)				
Replaced by stopCondParNew Replaced by $$stopCondParNew				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

stopCondParNew (C, S)				
Stop state parameter(s) Stop state parameter(s). More than one stop state can be active simultaneously. The highest priority stop state appears below the first line, this is followed by those with lower priorities.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

stopCondNum (C, S)				
Number of active stop states Number of active stop states. Specifies the number of occupied lines in $$stopCond				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

stopCondChangeCounter (C, S)				
Modification counter for stop states. Modification counter for stop states. Is incremented as soon as one of the stop states has changed.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

stopCondTime (C, S)				
Time stamp BCD for stop state BCD time stamp for stop state. More than one stop state can be active simultaneously. The highest priority stop state appears below the first line, this is followed by those with lower priorities.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DATETIME			
Addressing				
Line index:			Max. line index:	
			$$stopCondNum	
			Read access	

stopCondParA (C, S)				
ASCII stop state parameter(s) Stop state parameter(s). More than one stop state can be active simultaneously. The highest priority stop state appears below the first line, this is followed by those with lower priorities.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

stopRunActive (C, S)				
Stop run active				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1

stopRunActive (C, S)				
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

stopRunCounter (C, S)				
Modification counter for stop run Modification counter for stop run. This is always incremented when the NCK has stopped at a stop block.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

suppProgFunc (C, S)				
Suppression of language commands Disabling of language commands				
Return value	Bit0 = 0: SBLOF command is active Bit0 = 1: SBLOF command is disabled			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	Bit0 = 0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

syntaxCheckAlarmNo (C, S)				
Alarm number during the syntax check Alarm number in the case of a syntax error during the syntax check				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

syntaxCheckAlarmPara1 (C, S)				
Parameter 1 for an alarm during the syntax check Parameter 1 for an alarm in the case of a syntax error during the syntax check				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_STRING	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

syntaxCheckAlarmPara2 (C, S)				
Parameter 2 for an alarm during the syntax check Parameter 2 for an alarm in the case of a syntax error during the syntax check				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_STRING	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

syntaxCheckAlarmPara3 (C, S)				
Parameter 3 for an alarm during the syntax check Parameter 3 for an alarm in the case of a syntax error during the syntax check				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_STRING	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

syntaxCheckAlarmPara4 (C, S)				
Parameter 4 for an alarm during the syntax check Parameter 4 for an alarm in the case of a syntax error during the syntax check				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_STRING	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

syntaxCheckSeek (C, S)				
Faulty line in the syntax check Line number of the faulty line in the syntax check				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

syntaxCheckStatus (C, S)				
Status of the "Syntax check" function				
Return value	0: Syntax check not active (initialization status) 1: Syntax check selected 2: Syntax check active 3: Syntax check stopped with alarm on account of system error 4: Syntax check terminated 5: Syntax check canceled 6: Syntax check canceled on account of errors			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

threadPitch (C, S)				
Current lead				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

threadPitchS (C, S)				
Current lead during search run				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	

threadPitchS (C, S)			
Addressing			
Line index:		Max. line index:	
		1	
		Read access	

timeOrRevolDwell (C, S)				
Dwell time unit Dwell time unit in seconds or spindle revolutions				
Return value	0: $$cmdDwellTime and $$remainDwellTime in seconds 1: $$cmdDwellTime and $$remainDwellTime in spindle revolutions			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

timeS (C, S)	$AC_TIMES			
Time after a block change in seconds Time after a block change between programmed blocks in seconds				
Return value	Each programmed block can be divided up into a chain of part blocks that are processed one after the other. O n l y with the 1st cycle of the 1st block of the chain, timeS is set to zero and then counted up in seconds. Therefore, the variable enables time measurements throughout the entire block chain.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

timeSC (C, S)	$AC_TIMESC			
Time after a block change in IPO cycles Time after a block change between programmed blocks in IPO cycles				
Return value	Each programmed block can be divided up into a chain of part blocks that are processed one after the other. Only (!) with the 1st cycle of the 1st block of the chain, timeSC is set to zero and then counted up in seconds. Therefore, the variable enables time measurements throughout the entire block chain.			

timeSC (C, S)		$AC_TIMESC		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

toolCounter (C, S)				
Change counter of the tool data in a channel Counter of the changes of the tool data assigned to a channel. The counter is incremented each time a tool data is changed. All changes of tool data made by BTSS, part programs, INI files and by the Tool Management software are considered. Tool data are tool compensations, grinding-specific tool parameters, OEM tool parameters and Tool Managment data including magazine data. There is one exception: the present tool-in-use-time, since it is changed in each IPO cycle.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

toolCounterC (C, S)				
Modification counter for tool offsets Counter for modifications to tool offset data assigned to the channel (analog toolCounter).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

toolCounterM (C, S)				
Modification counter for magazine data Counter for modifications to magazine data assigned to the channel (analog toolCounter).				

toolCounterM (C, S)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

toolHolderData (C, S)	GETSELT, GETEXET	

Current tool holder data

Data for each tool holder/spindle from the magazine configuration of the TOA which is assigned to the channel.

There is a set of numToolHolderParams parameters for each tool holder.
Currently there are the 3 parameters P1, P2 and P3.
There are numToolHolders tool holders. The number of tool holders in this list is solely a function of the magazine configuration, and it does not change while an NC program runs.

- P1: THNo ToolHolderNumber / SpindleNumber
  (In the language commands of the NC program, corresponds to the address extension <n> from T<n>=... or M<n>=6 with explicit notation; in the magazine configuration, corresponds to the location type index of the associated buffer location of the location type = spindle.)
- P2: SelTno T number of the selected tool with reference to the tool holder / spindle with the number of THNo
  (The same TNo would also return the language command GETSELT.)
  The value 0 indicates that no tool is selected with reference to the tool holder. For further behavior see the description of GETSELT.
- P3: ExeTno TNumber of the tool to be loaded / loaded with reference to the tool holder / the spindle with the number THNo from the point of view of the NC program.
  When working without M6, the same TNumber is in SelTno and ExeTno.
  (The same TNumber would also return the language command GETEXET.)
  The value 0 indicates that no tool is to be loaded / is loaded with reference to the tool holder. For further behavior see the description of GETEXET.
- P4: SelTNoBeforeSearchRun
  During the search: TNummer of the selected tool with reference to the tool holder / spindle before the search, after reaching the search target and a T programming for this tool holder: the same value from P2.
- P5: ExeTNoBeforeSearchRun
  During the search: TNummer of the tool loaded with reference to the tool holder / spindle before the search, after reaching the search target and a tool change for this tool holder: the same value from P3.

An array access is possible to toolHolderData, with which the data of all numToolHolders tool holders can be read at one time.

If the flat D number is active, the value =0 is returned for all parameters.

Units and value ranges

Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	

Addressing

Line index:	Max. line index:

toolHolderData (C, S)	GETSELT, GETEXET	
	$$numToolHolderParams * $$numToolHolders	
	Read access	

toolholderOfDNo (C, S)		$P_TH_OF_D		
Spindle or tool number of the active D Number of the tool holder or spindle on which the active tool is mounted, which contains the active D no.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalParts (C, S)		$AC_TOTAL_PARTS		
Total of all machined workpieces Total number of all machined workpieces: This counter specifies the number of workpieces machined since it was started. The counter is automatically set to zero only if the control system boots on defaults.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

transSys (C, S)		$AC_TRANS_SYS		
Translation reference system Reference system for translation with cartesian manual traversal				
Return value	0: Axis-specific manual traversal active 1: Cartesian manual traversal in basic coordinate system active 2: Cartesian manual traversal in workpiece coordinate system active 3: Cartesian manual traversal in tool coordinate system active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

transfActive (C, S)		DB21-30, DBX33.6		K1, M1
Transformation active				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

vaCcCompVal (C, S)		$VA_CC_COMP_VAL[a,b]		
OA compensation value OA compensation value of the corresponding compile cycle				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEgSyncDiff (C, S)		$VA_EG_SYNCDIFF[a]		
EG synchronism deviation Electronic gear: Synchronism deviation (actual values). The comparison between this value and $MA_COUPLE_POS_TOL_... determines whether the appropriate "Synchronism" VDI signal is set.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEgSyncDiffS (C, S)	$VA_EG_SYNCDIFF_S[a]	
Synchronous run difference with sign Electronic gear: Synchronous run difference (actual values) with sign. Whether the corresponding VDI signal "synchronous run" is set depends upon the comparison of this value with $MA_COUPLE_POS_TOL_....		

vaEgSyncDiffS (C, S)		$VA_EG_SYNCDIFF_S[a]		
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

workPnameSubstitution (C, S)				
External path for selected program Path name defined by HMI on selection of a program or workpiece in /_N_EXT_DIR. It is only used for execute from external, and then denotes the data source of the download. HMI uses this path name in order to restore the program selection for execute from external after PowerOff. The string must be terminated with "\0". NCK does not use this path name. workPnameSubstitution is deleted on program selection in the passive file system or on an EES drive. Handling: workPnameSubstitution is written by the HMI on program selection for execute from external. NCK stores this information persistently. After PowerOff, the NCK deletes the reload buffer for execute from external, and selects _N_MPF0. HMI restores on the basis of the information: - _N_MPF0 is selected - workPnameSubstitution is set the selection for execute from external. With this program selection, the NCK does not remove the SPARPI interrupt pointer.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_STRING	0	0	
Addressing				
Line index:			Max. line index:	
			Read access	

progDNumber (C, S)				
Dummy - do not use!				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

progDLNumber (C, S)				
Dummy - do not use!				

progDLNumber (C, S)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

seruproActive (C, S)				
Block search via program test is active Block search via program test is active. That means that the NCK executes the part of the program before the search target in program test mode.				
Return value	Bit0 = 1: Block search via program test is active Bit1-n: Reserved for later expansions			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
			Read access	

progUsekt (C, S)				
Programmed value of the command $P_USEKT Programmed value of the command $P_USEKT. Bit-coded data for the programmed tool subgroup available for the tool change.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	0xF
			Read access	

acToolOAct (C, S)	$AC_TOOL_O_ACT			
Setpoint of the tool orientation Supplies the setpoint of the current tool orientation in various coordinate systems. Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS The orientation vector is normalized, i.e. it has the value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	

acToolOAct (C, S)	$AC_TOOL_O_ACT	
	9	
	Read access	

acToolOEnd (C, S)	$AC_TOOL_O_END			
End orientation Supplies the end orientation of the current block in various coordinate systems: Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS The orientation vector is normalized, i.e. it has the value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

acToolODiff (C, S)	$AC_TOOL_O_DIFF			
Remaining tool orientation angle Supplies the remaining angle between the current vector and end vectors of the tool orientation block in various coordinate systems: Possible values of the line index: 1: Angle in BCS 2: Angle in PCS/WCS 3: Angle in ENS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	0	180
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

vcToolO (C, S)		$VC_TOOL_O		
Actual value of the tool orientation Supplies the actual value of the current tool orientation in various coordinate systems: Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS The orientation vector is normalized, i.e. it has the value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

vcToolODiff (C, S)		$VC_TOOL_O_DIFF		
Angle between setpoint and actual orientation Supplies the angle between the setpoint vector and actual vector of the tool orientation in various coordinate systems: Possible values of the line index: 1: Angle in BCS 2: Angle in PCS/WCS 3: Angle in ENS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	0	180
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

vcToolOStat (C, S)		$VC_TOOLO_STAT		
Status of the actual orientation Supplies the status of the computation of the actual orientation				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-1	0
			Read access	

vaSyncDiff (C, S)		$VA_SYNCDIFF[]		
Actual value synchronism difference Actual value synchronism difference for all types of coupling				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaSyncDiff (C, S)		$AA_SYNCDIFF[]		
Setpoint synchronism difference Setpoint synchronism difference for all types of coupling				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaSyncDiffStat (C, S)		$VA_SYNCDIFF_STAT[]		
Status of the actual value synchronism difference				
Return value	-4: Reserved -3: No valid value in $VA_SYNCDIFF, tangential control -2: No valid value in $VA_SYNCDIFF, master value coupling and simulated master value -1: No valid value in $VA_SYNCDIFF 0: No valid value in $VA_SYNCDIFF, coupling not active 1: Valid value in $VA_SYNCDIFF			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-4	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaSyncDiffStat (C, S)		$AA_SYNCDIFF_STAT[]		
Status of setpoint synchronism difference				
Return value	-4: No valid value in $AA_SYNCDIFF, coupled motion from part program -3: Reserved -2: Reserved -1: No valid value in $AA_SYNCDIFF 0: No valid value in $AA_SYNCDIFF, coupling not active 1: Valid value in $AA_SYNCDIFF			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-4	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

acToolRAct (C, S)		$AC_TOOL_R_ACT		
Setpoint of the tool rotation. Setpoint of the tool rotation in various coordinate systems: Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS The orientation vector is normalized, i.e. it has the value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

acToolREnd (C, S)		$AC_TOOL_R_END		
End rotation vector of the current block End rotation vector of the current block in various coordinate systems: Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS The orientation vector is normalized, i.e. it has the absolute value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

acToolRDiff (C, S)		$AC_TOOL_R_DIFF		
Remaining tool rotation angle Angle remaining between the current and end rotation vectors of the tool orientation block in various coordinate systems: Possible values of the line index: 1: Angle in BCS 2: Angle in PCS/WCS 3: Angle in ENS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	0	180
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

vcToolR (C, S)		$VC_TOOL_R		
Actual value of tool rotation Actual value of the tool rotation in various coordinate systems: Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS The rotation vector is normalized, i.e. it has the value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

vcToolRDiff (C, S)		$VC_TOOL_R_DIFF		
Angle bet. setpoint & actual tool rotation vectors Angle between setpoint and actual tool rotation vectors in different coordinate systems: Possible values of the line index: 1: Angle in BCS 2: Angle in PCS/WCS 3: Angle in ENS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
Degree	TYPE_DOUBLE	0	0	180
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

vcToolRStat (C, S)		$VC_TOOLR_STAT		
Status of the computation of the actual rotation				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-1	0
Addressing				
Line index:			Max. line index:	

vcToolRStat (C, S)	$VC_TOOLR_STAT	
	1	
	Read access	

pToolRot (C, S)		$P_TOOL_O_R		
Current tool rotation Current tool rotation in various coordinate systems: Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS The rotation vector is normalized, i.e. it has the absolute value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

searchRunMode (C, S)				
Function with search run Type of function in which search run has been integrated				
Return value	1: Search run is used directly 2: Simulation search run 3: Execute program area The user can preselect a program area via HMI with "Execute program area", which he/she would like to execute "real" on the machine. To do this, the NCK uses an internal block search to approach the start of the program area (abbreviation: APb) correctly. Internal cancellation at the end of the program area (abbreviation: EPb) via reset. 0: Otherwise			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	1	3
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

searchRunStatus (C, S)		
Status of the block search run The variable returns the various states for a block search. Depending on the state, defined responses must be executed, including those by an operator, to complete the block search: o 1: activeSearchRun Simulation active, that is, from part program start to the agreed search target (or APb), the NCK simulates the part program in order to find the correct start position of the search target block. o 2: targetFound The search target has been found, and the NCK waits for the Start key. The simulation has ended. o 3: activeAdaptation After the start, the NCK outputs action blocks, which set the machine to the search target (M± function output, speeds for spindles) and possibly starts an ASUB, in which the user adapts the machine to the part program situation in the target block via the ASUB program (Example: programmed tool is read and replaced by the current one with a tool change cycle. The NCK stops automatically after the action blocks or after the ASUB with alarm 10208. o 4: finishedAdaptation The NCK waits for the start. o 5: activeStopRun After the adaptation, the REPOS function leads to the target block and then the program resumes execution. The NCK executes the program area after the target block, but is still in the function Execute program area. The blocks are scanned to see whether the end of the program area (EPb) might already have been reached. At EPb, the program is cancelled with reset, and searchRunStatus is deleted. 0: Otherwise		

searchRunStatus (C, S)				
Return value	1: activeSearchRun Simulation active, that is the NCK simulates the part program from the start to the agreed search target (or APb) in order to find, among other things, the correct starting position of the search target block. 2: targetFound The search target has been found and the NCK is waiting for the Start button. Simulation has finished. 3: activeAdaption After the start, the NCK outputs action blocks which set the machine to the search target (M± function output, spindle speeds) and, if applicable, starts an ASUB in which the user uses the ASUB program to adapt the machine to the part program situation in the target block. (For example, a programmed tool is read and a tool-changing cycle exchanges it with the current tool.) The NCK stops automatically after the action blocks or after the ASUB with alarm 10208. 4: finishedAdaption The NCK waits for the start. 5: activeStopRun After the adaptation, the REPOS function goes to the target block, and then the execution of the program is resumed. The NCK executes the program area after the target block, but is still within the function Execute program area. The blocks are scanned to see whether the end of the program area (EPb) could already have been reached. The program is cancelled at EPb with reset and searchRunStatus is cleared. 0: Otherwise			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	1	5
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

enableOvrRapidFactor (C, S)				
Activate additional rapid traverse override Activate additional rapid traverse override $SC_OVR_RAPID_FACTOR				
Return value	0: not activated 1: activated			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1

enableOvrRapidFactor (C, S)		
Addressing		
Line index:	Max. line index:	
	1	
	Read access and write access	

actWalimGroupNo (C, S)		$AC_WORKAREA_CS_GROUP		
Active working area group in the IPO Active working area group in the IPO Writing is only possible if the channel has been aborted or stopped				
Return value	0: Not activated n: $MC_MM_NUM_WORKAREA_CS_GROUPS			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	10
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

acSafeSynaMem (C, S)		$AC_SAFE_SYNA_MEM		
Free Safety synchronized action elements Free Safety synchronized action elements The maximum number of elements is configured in $MC_MM_NUM_SAFE_SYNC_ELEMENTS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

tOffL1L2L3 (C, S)		$AC_TOFFL		
Tool length offset TOFFL Tool length offset programmed in coordinates of the tool length components L1/L2/L3.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

tOffLXYZ (C, S)		$AC_TOFF		
Tool length offset TOFFL in the WCS Tool length offset TOFFL programmed in coordinates of the WCS.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acJogCircleSelected (C, S)		$AC_JOG_CIRCLE_SELECTED		
JOG in circles has been selected				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

tOffR (C, S)		$AC_TOFFR		
Tool radius offset TOFFR Programmed tool radius offset.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

simTolerance (C, S)		keine		
Simulation tolerance The NCK simulation can run part programs at a higher speed (see PI _N_NCKMOD). Only then is simTolerance evaluated and it ONLY affects geometry blocks that have not been programmed as circles or straight lines. These blocks are slowed down so that two successive interpolation points can be connected by a straight line. This straight line deviates from the programmed contour by not more than 'simTolerance'.				
Units and value ranges				
Physical unit:	Data type:	Initial value		

simTolerance (C, S)		keine		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

incoapSize (C, S)		$P_INCOAP_SIZE[]		
Array size of the cutting generator parameters Array size of the supply and return parameters of the COA application cutting generator				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			7	
			Read access and write access	

incoapB (C, S)		$P_INCOAP_B		
Boolean parameter(s) of the cutting generator Boolean supply and return parameter(s) of the COA application cutting generator				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$incoapSize[1]	
			Read access and write access	

incoapC (C, S)		$P_INCOAP_C		
CHAR parameter(s) of the cutting generator CHAR supply and return parameter(s) of the COA application cutting generator				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	255
Addressing				
Line index:			Max. line index:	
			$$incoapSize[2]	
			Read access and write access	

incoapI (C, S)		$P_INCOAP_I		
INT parameter(s) of the cutting generator INT supply and return parameter(s) of the COA application cutting generator				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		

incoapI (C, S)	$P_INCOAP_I	
Addressing		
Line index:	Max. line index:	
	$$incoapSize[3]	
	Read access and write access	

incoapR (C, S)		$P_INCOAP_R		
DOUBLE parameter(s) of the cutting generator DOUBLE supply and return parameter(s) of the COA application cutting generator				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$incoapSize[4]	
			Read access and write access	

incoapS16 (C, S)		$P_INCOAP_S16[]		
CHAR16 parameter(s) of the cutting generator CHAR16 supply and return parameter(s) of the COA application cutting generator				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$$incoapSize[5]	
			Read access and write access	

incoapS32 (C, S)		$P_INCOAP_S32[]		
CHAR32 parameter(s) of the cutting generator CHAR32 supply and return parameter(s) of the COA application cutting generator				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$$incoapSize[6]	
			Read access and write access	

incoapS160 (C, S)		$P_INCOAP_S160[]		
CHAR160 parameter(s) of the cutting generator CHAR160 supply and return parameter(s) of the COA application cutting generator				
Units and value ranges				
Physical unit:	Data type:	Initial value		

incoapS160 (C, S)		$P_INCOAP_S160[]		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$$incoapSize[6]	
			Read access and write access	

isoActHDNo (C, S)		$P_ISO2_HNO[n],$P_ISO2_DNO, $P_ISO3_NO		
Active tool offset number in ISO mode The following applies to rows 1- 4: This value is only practical if ISO2 mode is permissible. This value contains the H number of the tool length offsets in the 3 geometry dimensions in rows 1 - 3 and the D number of the tool radius offset. If H99 is programmed, then all 3 geometry dimensions (=rows 1-3) have the value "-1" the radius (= row 4) has the value "-1" If H=D ($MN_EXTERN_TOOLPROG_MODE,Bit6=0), then these variables contain the last programmed D or H. If an offset D > 1 is selected in Siemens mode, all rows have the value "-2". If ISO2 mode cannot be activated ($MN_MM_EXTERN_CNC_SYSTEM != 4), the variable has the value=-3. The following applies to row 5: This value is only practicable if ISO3 mode is permissible. This value contains the current number of tool offsets in ISO3 mode. If an offset D > 1 is selected in Siemens mode, the value "-2" is returned. If ISO3 mode cannot be activated ($MN_MM_EXTERN_CNC_SYSTEM != 5), the variable value =-3.				
Return value	-3: ISO2 or ISO3 mode inactive -2: Siemens offset selected with D > 1. -1: H99 programmed in ISO mode, Siemens offset D1 active			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_WORD	0		
Addressing				
Line index:			Max. line index:	
			5	
			Read access	

toolCounterIso (C, S)		keine		
Modification counter for ISO tool offset data Each change of a tool offset value for ISO2.2 and ISO3.2 modes is counted. This is to enable the HMI to record data changes.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	

toolCounterIso (C, S)	keine		
		1	
		Read access	

actLanguage (C, S)				
Active language mode				
Return value	0: Siemens 1: ISO mode 2: Reserviert for later language expansions			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

acPrepActLoad (C, S)		$AC_PREP_ACT_LOAD		
Current preprocessing run time				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPrepMaxLoad (C, S)		$AC_PREP_MAX_LOAD		
Longest preprocessing run time				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPrepMinLoad (C, S)		$AC_PREP_MIN_LOAD		
Shortest preprocessing run time				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPrepActLoadGross (C, S)		$AC_PREP_ACT_LOAD_GROSS		
Current preprocessing gross run time				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPrepMaxLoadGross (C, S)		$AC_PREP_MAX_LOAD_GROSS		
Longest preprocessing gross run time				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acPrepMinLoadGross (C, S)		$AC_PREP_MIN_LOAD_GROSS		
Shortest preprocessing gross run time				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acCTol (C, S)		$AC_CTOL		
Active contour tolerance acCTol provides the contour tolerance for compressor and smoothing which was used to preprocess the current main run block.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acOTol (C, S)		$AC_OTOL		
Active orientation tolerance acOTol provides the orientation tolerance for compressor and smoothing which was used to preprocess the current main run block.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
Degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acSTolF (C, S)		$AC_STOLF		
Active G00 tolerance factor acSTolF indicates the G00 tolerance factor for compressor and smoothing with which the current main run block was prepared.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aaATol (C, S)		$AA_ATOL		
Active axis tolerance aaATol provides the axis tolerance for compressor and smoothing, which was effective during preprocessing of the current main run block.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaFgref (C, S)		$AA_FGREF		
Active factor for rotary axis path The variable provides the radius used by a rotary axis to contribute to the path travel. The default value is 180 mm/PI = 57,296 mm, which corresponds to a contribution of 1mm per degree. For linear axes the variable always provides 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	

aaFgref (C, S)		$AA_FGREF		
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaFgroup (C, S)		$AA_FGROUP		
Influence of an axis on the path velocity If the travel of an axis influences the path velocity in the current main run block (FGROUP), the variable provides value 1, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

acFgroupMask (C, S)		$AC_FGROUP_MASK		
Key of the path velocity axes acFgroupMask bit-coded provides the channel axes which are to contribute to the path velocity				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	0xFFFF
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acIpoState (C, S)		$AC_IPO_STATE		
Active functions identification status The variable provides selected information about whether certain functions are active: Bit 0: Free-form surface mode is active Bit 1: Compressor active				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0x0003

acIpoState (C, S)	$AC_IPO_STATE	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

acFZ (C, S)		$AC_FZ		
Tooth feedrate, setpoint Tooth feedrate, setpoint. The physical unit is defined in variable 'feedRateIpoUnit'.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
			Read access	

pOriSol (C, S)		$P_ORI_SOL		
Angle calc. status during orientation programming Contains the number of solutions and additional status information for orientation programming. See also the documentation for the corresponding system variables.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
			Read access	

pOriStat (C, S)		$P_ORI_STAT		
Status of orientation axes during orientation prog Contains the status of the orientation axes during orientation programming. See also the documentation for the corresponding system variables.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

pOriPos0 (C, S)		$P_ORI_POS[0,n]		
Angle of orientation axes (1st solution) The angle of the orientation axes of the first (or single) solution during orientation programming.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	

pOriPos0 (C, S)	$P_ORI_POS[0,n]	
	2	
	Read access	

pOriPos1 (C, S)		$P_ORI_POS[1,n]		
Angle of orientation axes (2nd solution) The angle of the orientation axes of the second solution during orientation programming.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

pOriDiff0 (C, S)		$P_ORI_DIFF[0,n]		
Angle diff. of orientation axes (1st solution) The angle difference between the exact angles and those available in $P_ORI_ANG for the orientation axes of the first (or single) solution during orientation programming. The content of these variables can only be unequal to zero if the positions of the orientation axes are incremental (Hirth coupling).				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

pOriDiff1 (C, S)		$P_ORI_DIFF[1,n]		
Angle diff. of orientation axes (2nd solution) The angle difference between the exact angles and those available in $P_ORI_ANG for the orientation axes of the second solution during orientation programming. The content of these variables can only be unequal to zero if the positions of the orientation axes are incremental (Hirth coupling).				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

toolFrameState (C, S)				
Info about tool orientation in the work				
toolFrameState provides bit-coded information about whether the PI service _N_SETUDT with the function designations 12 and 13 can be activated in its current state and also specifies any parameters that may be required: Bit 0 provides information about whether the NCK can generate a tool frame with PI service _N_SETUDT and function designation 12 in its current state. Once the bit is set, the NCK receives information about the current tool orientation, i.e. either a tool holder that can be oriented or an orientation transformation is active and the tool frame can be generated. Bit 1 provides information about whether, in its current state, the NCK has stored data for the restoration of the program environment (bit 1 = 1), which can be restored using PI service _N_SETUDT and function designation 13. When bit 0 is set, bit 2 provides information about whether the tool axis corresponds to a geometry axis of the current WCS: Bit 2 = 0: The tool axis does not correspond to any geometry axes of the current WCS. In this case, bit 3 / bit 4 provide the number of the geometry axis next to the tool axis. Bit 5 provides the retraction direction of this axis (plus/minus). This information can be displayed by the HMI as a suggestion or default setting for the retraction axis. Bit 2 = 1: The tool axis corresponds to a geometry axis of the current work. In this case bit 3 / bit 4 provide the number of this geometry axis, and bit 5 provides its retraction direction (plus/minus).				
Return value	Bit 0: 0: PI service _N_SETUDT with function designation 12 disabled 1: PI service _N_SETUDT with function designation 12 enabled Bit 1: 0: PI service _N_SETUDT with function designation 13 disabled 1: PI service _N_SETUDT with function designation 13 enabled Bit 2 0: Tool axis does not correspond to any geometry axes 1: Tool axis corresponds to a geometry axis Bit 3 / Bit 4:0: Function not active 1: Tool axis in the 1st geometry axis 2: Tool axis in the 2nd geometry axis 3: Tool axis in the 3rd geometry axis Bit 5: 0: Retraction direction plus 1: Retraction direction minus				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	63
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pcTrafoRotChainIndex (C, S)	$PC_TRAFO_ROT_CHAIN_INDEX	
Index of the orientation axis in the kin. chain.		
Images the indices of the orientation axes in the array $NT_ROT_AX_NAME onto the internal orientation axis sequence. See also documentation of the associated system variables.		

pcTrafoRotChainIndex (C, S)		$PC_TRAFO_ROT_CHAIN_INDEX		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

pcTrafoRotChanAxIn (C, S)		$PC_TRAFO_ROT_CHAN_AX_IN		
Channel axis index of an orientation axis (int.). Determines the channel axis index of the i-th orientation axis, where i is the index of this axis in the internal representation of the kinematic chain. See also documentation of the associated system variables.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

pcTrafoRotChanAxEx (C, S)		$PC_TRAFO_ROT_CHAN_AX_EX		
Channel axis index of an orientation axis (ext.). Determines the channel axis index of the i-th orientation axis, where i is the index of this axis in the external representation of the kinematic chain (index of the entry in the array$NT_ROT_AX_NAME[n, i]). See also documentation of the associated system variables.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

retractState (C, S)				
Status information on JOG retract Status information on the submode JOG retract				
Return value	Bit 0: 0: No retract data available; JOG retract cannot be activated 1: Retract data available; JOG retract can be activated with PI _N_RETRAC Bit 1: 0: JOG retract submode is not active 1: JOG retract submode is active Bit 3/2: 0: Function is not active 1: Retraction axis is 1st geometry axis according to $MC_AXCONF_GEOAX_ASSIGN_TAB 2: Retraction axis is 2nd geometry axis according to $MC_AXCONF_GEOAX_ASSIGN_TAB 3: Retraction axis is 3rd geometry axis according to $MC_AXCONF_GEOAX_ASSIGN_TAB Bit 4/5: 0: Default tool (milling cutter) 1: Tap (tapping with G33/G331/G332 is active) 2: Drilling tool from the 200 group Bit 6/7: Reserved Bit 8/11: 0: No problem known 1: No tool selected 2: No retract tool selected (turning or grinding tool) 3: Tool offset is not active 4: Retraction axis does not exist 5: G63 block 6: Path motion not in direction of tool axis 7: Retract data not persistent due to $MN_MM_ACTFILESYS_LOG_FILE_MEM[1]=0 8: Retract data are not consistent Bit 12/15: Reserved			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pCutModK (C, S)		$AC_CUTMODK		
Last programmed value CUTMODK Reads the currently valid value last programmed with the language command CUTMODK. (Name of the orientation transformation defined by kinematic chains for which the edge data modification is to be activated).				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	"\0"		

pCutModK (C, S)	$AC_CUTMODK	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

pCutModKS (C, S)				
Last programmed value CUTMODK (block search)				
Reads the currently valid value last programmed with the language command CUTMODK.				
(Name of the orientation transformation defined by kinematic chains for which the edge data modification is to be activated).				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	"\0"		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acTrafoName (C, S)	$AC_TRAFO_NAME	
Name of an active kinematic transformation		
Reads the name of a currently active kinematic transformation.		
If no transformation is active or if one transformation is active that has not been defined with kinematic chains, then this variable receives a zero string.		
Units and value ranges		
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	"\0"		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actToolGeoLength (C, S)		
Geometry component of the active tool		
Returns the length components of the geometry component of the active tool, that is the contribution of the components $TC_DP3[..] - $TC_DP5[..] in different coordinate systems.		
Both the component and the coordinate system are selected from the line index.		
Three indices are required for each coordinate system (lengths L1, L2, L3).		
The following assignment applies:		
Line indices 1 - 3: Components in the workpiece coordinate system (PCS).		
Line indices 4 - 6: Components in the basic coordinate system (BCS).		
Line indices 7 - 9: Components in the machine coordinate system (MCS).		
Line indices 10 - 12: Components in the tool coordinate system (TCS).		
Line indices 13 - 15: Components in the settable zero system (SZS).		

actToolGeoLength (C, S)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			15	
			Read access	

actToolGeoLengthWear (C, S)				
Wear component of the active tool				
Returns the components of the wear component of the tool length of the active tool, that is the contribution of the components $TC_DP12[..] - $TC_DP14[..] in different coordinate systems. Both the component and the coordinate system are selected from the line index. Three indices are required for each coordinate system (lengths L1, L2, L3). The following assignment applies: Line indices 1 - 3: Components in the workpiece coordinate system (PCS). Line indices 4 - 6: Components in the basic coordinate system (BCS). Line indices 7 - 9: Components in the machine coordinate system (MCS). Line indices 10 - 12: Components in the tool coordinate system (TCS). Line indices 13 - 15: Components in the settable zero system (SZS).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			15	
			Read access	

actToolSumCorrLength (C, S)				
Total offset of the active tool				
Returns the components of the total offset of the active tool, that is the contribution of the components $TC_SCPx3[..] - $TC_SCPx5[..] in different coordinate systems. The letter "x" in the components $TC_SCPx3[..] etc. stands for the DL number. Both the component and the coordinate system are selected from the line index. Three indices are required for each coordinate system (lengths L1, L2, L3). The following assignment applies: Line indices 1 - 3: Components in the workpiece coordinate system (PCS). Line indices 4 - 6: Components in the basic coordinate system (BCS). Line indices 7 - 9: Components in the machine coordinate system (MCS). Line indices 10 - 12: Components in the tool coordinate system (TCS). Line indices 13 - 15: Components in the settable zero system (SZS).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

actToolSumCorrLength (C, S)		
Addressing		
Line index:	Max. line index:	
	15	
	Read access	

actToolAdapterBaseLength (C, S)				
Adapter or base dimension of the active tool.				
Returns the components of the adapter or base dimension of the active tool, that is the contribution of the components $TC_ADPT1[..] - $TC_ADPT3[..] or $TC_DP21[..] - $TC_DP233[..] in different coordinate systems.				
The adapter and base dimension are mutually exclusive. This means that only one of these two components can contribute a value unequal to zero to the tool length.				
Both the component and the coordinate system are selected from the line index.				
Three indices are required for each coordinate system (lengths L1, L2, L3).				
The following assignment applies:				
Line indices 1 - 3: Components in the workpiece coordinate system (PCS).				
Line indices 4 - 6: Components in the basic coordinate system (BCS).				
Line indices 7 - 9: Components in the machine coordinate system (MCS).				
Line indices 10 - 12: Components in the tool coordinate system (TCS).				
Line indices 13 - 15: Components in the settable zero system (SZS).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			15	
			Read access	

actToolEntryCorrLength (C, S)				
Setup offset of the active tool				
Returns the components of the total offset of the active tool, that is the contribution of the components $TC_ECPx3[..] - $TC_ECPx5[..] in different coordinate systems. The letter "x" in the components $TC_SCPx3[..] etc. stands for the DL number.				
Both the component and the coordinate system are selected from the line index.				
Three indices are required for each coordinate system (lengths L1, L2, L3).				
The following assignment applies:				
Line indices 1 - 3: Components in the workpiece coordinate system (PCS).				
Line indices 4 - 6: Components in the basic coordinate system (BCS).				
Line indices 7 - 9: Components in the machine coordinate system (MCS).				
Line indices 10 - 12: Components in the tool coordinate system (TCS).				
Line indices 13 - 15: Components in the settable zero system (SZS).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

actToolEntryCorrLength (C, S)				
Addressing				
Line index:			Max. line index:	
			15	
			Read access	

actToolToolCarrierLength (C, S)				
Contribution of tool carrier to the tool length				
Returns the components of the contribution of the orientable tool carrier (ToolCarrier) to the length of the active tool in different coordinate systems.				
Both the component and the coordinate system are selected from the line index.				
Three indices are required for each coordinate system (lengths L1, L2, L3).				
The following assignment applies:				
Line indices 1 - 3: Components in the workpiece coordinate system (PCS).				
Line indices 4 - 6: Components in the basic coordinate system (BCS).				
Line indices 7 - 9: Components in the machine coordinate system (MCS).				
Line indices 10 - 12: Components in the tool coordinate system (TCS).				
Line indices 13 - 15: Components in the settable zero system (SZS).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			15	
			Read access	

actToolTotalLength (C, S)				
Total length of the active tool				
Returns the components of the total effective length of the active tool in different coordinate systems.				
Both the component and the coordinate system are selected from the line index.				
Three indices are required for each coordinate system (lengths L1, L2, L3).				
The following assignment applies:				
Line indices 1 - 3: Components in the workpiece coordinate system (PCS).				
Line indices 4 - 6: Components in the basic coordinate system (BCS).				
Line indices 7 - 9: Components in the machine coordinate system (MCS).				
Line indices 10 - 12: Components in the tool coordinate system (TCS).				
Line indices 13 - 15: Components in the settable zero system (SZS).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			15	
			Read access	

acToolOCorr (C, S)		$AC_TOOL_O_CORR		
Setpoint of the orientation with overlays Supplies the setpoint of the current tool orientation incl. overlays in various coordinate systems. Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS The orientation vector is normalized, i.e. it has the value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

acToolRCorr (C, S)		$AC_TOOL_R_CORR		
Setpoint of rotation with overlays Supplies the setpoint of the current rotation vector of the tool orientation incl. overlays in various coordinate systems. Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS The orientation vector is normalized, i.e. it has the value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

acToolOCorrD (C, S)		$AC_TOOL_O_CORRD		
Overlaid orientation Supplies the setpoint of the current overlay of the tool orientation in various coordinate systems. Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS This vector represents the difference between the two vectors acToolOCorr and acToolOAct				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

acToolRCorrD (C, S)		$AC_TOOL_R_CORRD		
Overlaid rotation of the orientation Supplies the setpoint of the current overlay of the rotation of the tool in various coordinate systems. Possible values of the line index: 1, 2, 3: Components of the vector in BCS 4, 5, 6: Components of the vector in PCS/WCS 7, 8, 9: Components of the vector in ENS This vector represents the difference between the two vectors acToolRCorr and acToolRAct				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

startLockCounter (C, S)		
Start counter for channel-specific start disable Counter that is incremented as soon as an NC start is activated with a set channel-specific start disable (see _N_STRTLK).		

startLockCounter (C, S)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pCutModKA (C, S)		$AC_CUTMODKA		
CUTMODK active Cutter position modification for a transformation defined with kinematic chains is active.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pCutModKAS (C, S)		$P_CUTMODKA		
CUTMODK active (block search) Cutter position modification for a transformation defined with kinematic chains is active.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acMToolLengthIndex (C, S)		$AC_M_TOOL_LENGTH_INDEX		
Assign. tool length components for milling tools The variable returns the number of the tool length components (1, 2 or 3 corresponding to the length components L1, L2, L3) for milling tools, which is assigned to the geometry axis that was transferred as an index. In this context, milling tools are all tools with a tool type that does not lie between 400 and 599. The assignment does not take into account any rotations (e.g. by kinematic transformations) or frames. It depends on the active plane and the setting data SD42950 $SC_TOOL_LENGTH_TYPE and SD42940 $SC_TOOL_LENGTH_CONST. If the tool length component is active with a negative sign, the index is output with a negative sign. This can be the case if the hundreds digit of setting data SD42940 $SC_TOOL_LENGTH_CONST equals 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	-3	3
Addressing				
Line index:			Max. line index:	

acMToolLengthIndex (C, S)	$AC_M_TOOL_LENGTH_INDEX	
	3	
	Read access	

acMToolLengthIndexS (C, S)	$P_M_TOOL_LENGTH_INDEX			
Assign. lngth comps f. milling tool (block search)				
The variable returns the number of the tool length components (1, 2 or 3 corresponding to the length components L1, L2, L3) for milling tools, which is assigned to the geometry axis that was transferred as an index. In this context, milling tools are all tools with a tool type that does not lie between 400 and 599. The assignment does not take into account any rotations (e.g. by kinematic transformations) or frames. It depends on the active plane and the setting data SD42950 $SC_TOOL_LENGTH_TYPE and SD42940 $SC_TOOL_LENGTH_CONST. If the tool length component is active with a negative sign, the index is output with a negative sign. This can be the case if the hundreds digit of setting data SD42940 $SC_TOOL_LENGTH_CONST equals 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	-3	3
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTToolLengthIndex (C, S)	$AC_T_TOOL_LENGTH_INDEX			
Assign. tool length components for turning tools				
The variable returns the number of the tool length components (1, 2 or 3 corresponding to the length components L1, L2, L3) for turning and grinding tools, which is assigned to the geometry axis that was transferred as an index. In this context, turning and grinding tools are all tools with a tool type that lies between 400 and 599. The assignment does not take into account any rotations (e.g. by kinematic transformations) or frames. It depends on the active plane and the setting data SD42950 $SC_TOOL_LENGTH_TYPE, SD42940 $SC_TOOL_LENGTH_CONST and SD42942 $SC_TOOL_LENGTH_CONST_T. If the tool length component is active with a negative sign, the index is output with a negative sign. This can be the case if the hundreds digit of setting data SD42940 $SC_TOOL_LENGTH_CONST or the hundreds digit of setting data SD42942 $SC_TOOL_LENGTH_CONST_T equals 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	-3	3
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTToolLengthIndexS (C, S)	$P_T_TOOL_LENGTH_INDEX			
Assign. lngth comps f. turning tool (block search)				
The variable returns the number of the tool length components (1, 2 or 3 corresponding to the length components L1, L2, L3) for turning and grinding tools, which is assigned to the geometry axis that was transferred as an index. In this context, turning and grinding tools are all tools with a tool type that lies between 400 and 599. The assignment does not take into account any rotations (e.g. by kinematic transformations) or frames. It depends on the active plane and the setting data SD42950 $SC_TOOL_LENGTH_TYPE, SD42940 $SC_TOOL_LENGTH_CONST and SD42942 $SC_TOOL_LENGTH_CONST_T. If the tool length component is active with a negative sign, the index is output with a negative sign. This can be the case if the hundreds digit of setting data SD42940 $SC_TOOL_LENGTH_CONST or the hundreds digit of setting data SD42942 $SC_TOOL_LENGTH_CONST_T equals 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	-3	3
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acActToolLengthIndex (C, S)	$AC_ACT_TOOL_LENGTH_INDEX			
Assignment of length components for active tool				
The variable returns the number of the tool length components (1, 2 or 3 corresponding to the length components L1, L2, L3) of the active tool, which is assigned to the geometry axis that was transferred as an index. The assignment does not take into account any rotations (e.g. by kinematic transformations) or frames. It depends on the type of active tool, the active plane, any active adapter transformation and the setting data SD42950 $SC_TOOL_LENGTH_TYPE, SD42940 $SC_TOOL_LENGTH_CONST and SD42942 $SC_TOOL_LENGTH_CONST_T. Active mirrorings of a frame can influence the output value if setting data SD42900 $SC_MIRROR_TOOL_LENGTH, is set, see below. If the tool length component is active with a negative sign, the index is output with a negative sign. This can be the case if the hundreds digit of setting data SD42940 $SC_TOOL_LENGTH_CONST or the hundreds digit of setting data SD42942 $SC_TOOL_LENGTH_CONST_T equals 1, or if a mirroring of the relevant axis is active on account of setting data $SC_MIRROR_TOOL_LENGTH. If both causes are active simultaneously, the resulting sign is positive once more. If no tool is active, the value 0 is returned.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	-3	3
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acActToolLengthIndexS (C, S)	$P_ACT_TOOL_LENGTH_INDEX			
Length component assignment for active tool (SSL) The variable returns the number of the tool length components (1, 2 or 3 corresponding to the length components L1, L2, L3) of the active tool, which is assigned to the geometry axis that was transferred as an index. The assignment does not take into account any rotations (e.g. by kinematic transformations) or frames. It depends on the type of active tool, the active plane, any active adapter transformation and the setting data SD42950 $SC_TOOL_LENGTH_TYPE, SD42940 $SC_TOOL_LENGTH_CONST and SD42942 $SC_TOOL_LENGTH_CONST_T. Active mirrorings of a frame can influence the output value if setting data SD42900 $SC_MIRROR_TOOL_LENGTH, is set, see below. If the tool length component is active with a negative sign, the index is output with a negative sign. This can be the case if the hundreds digit of setting data SD42940 $SC_TOOL_LENGTH_CONST or the hundreds digit of setting data SD42942 $SC_TOOL_LENGTH_CONST_T equals 1, or if a mirroring of the relevant axis is active on account of setting data $SC_MIRROR_TOOL_LENGTH. If both causes are active simultaneously, the resulting sign is positive once more. If no tool is active, the value 0 is returned.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	-3	3
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acInKeyG (C, S)				
Read grinding input (PLC->NCK) Grinding: returns the current value of the relevant grinding input.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

acInKeyGEnable (C, S)				
Activate grinding input Grinding: indicates whether the relevant grinding input has been activated.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

acInKeyGIsEnable (C, S)				
Is the grinding input active? Grinding: indicates whether the relevant grinding input is active.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

acInKeyGRunIn (C, S)				
Read grinding input (PLC->NCK) Grinding: returns the current value of the relevant grinding input (PLC)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

acInKeyGRunOut (C, S)				
Read grinding input (NCK->PLC) Grinding: returns the current value of the relevant grinding input (NCK)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

chanStartLockState (C, S)				
Status of the channel-specific start disable Status of the channel-specific start disable. Also see PI _N_STRTLK and _N_STRTUL.				
Return value	0: No start disable 1: Start disable is activated			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1

chanStartLockState (C, S)				
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acTrafoCorrElemP0 (C, S)		$AC_TRAFO_CORR_ELEM_P[0,n]		
Correction element with index 0 in the part chain. Correction element in the section including the index 0 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoCorrElemP1 (C, S)		$AC_TRAFO_CORR_ELEM_P[1,n]		
Correction element with index 1 in the part chain. Correction element in the section including the index 1 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoCorrElemP2 (C, S)		$AC_TRAFO_CORR_ELEM_P[2,n]		
Correction element with index 2 in the part chain. Correction element in the section including the index 2 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoCorrElemP3 (C, S)		$AC_TRAFO_CORR_ELEM_P[3,n]		
Correction element with index 3 in the part chain. Correction element in the section including the index 3 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		

acTrafoCorrElemP3 (C, S)		$AC_TRAFO_CORR_ELEM_P[3,n]		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoCorrElemT0 (C, S)		$AC_TRAFO_CORR_ELEM_T[0,n]		
Correction element with index 0 in the tool chain. Correction element in the section with the index 0 in the tool chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoCorrElemT1 (C, S)		$AC_TRAFO_CORR_ELEM_T[1,n]		
Correction element with index 1 in the tool chain. Correction element in the section with the index 1 in the tool chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoCorrElemT2 (C, S)		$AC_TRAFO_CORR_ELEM_T[2,n]		
Correction element with index 2 in the tool chain. Correction element in the section with the index 2 in the tool chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoCorrElemT3 (C, S)		$AC_TRAFO_CORR_ELEM_T[3,n]		
Correction element with index 3 in the tool chain. Correction element in the section with the index 3 in the tool chain of an active orientation transformation.				

acTrafoCorrElemT3 (C, S)		$AC_TRAFO_CORR_ELEM_T[3,n]		
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoOriaxDirP0 (C, S)		$AC_TRAFO_ORIAX_DIR_P[0,n]		
Orient. axis direction with index 0 in part chain. Direction vector of the orientation axis with the index 0 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoOriaxDirP1 (C, S)		$AC_TRAFO_ORIAX_DIR_P[1,n]		
Orient. axis direction with index 1 in part chain. Direction vector of the orientation axis with the index 1 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoOriaxDirP2 (C, S)		$AC_TRAFO_ORIAX_DIR_P[2,n]		
Orient. axis direction with index 2 in part chain. Direction vector of the orientation axis with the index 2 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoOriaxDirT0 (C, S)		$AC_TRAFO_ORIAX_DIR_T[0,n]		
Orientation direction index 0 in tool chain. Direction vector of the orientation axis with the index 0 in the tool chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoOriaxDirT1 (C, S)		$AC_TRAFO_ORIAX_DIR_T[1,n]		
Orientation direction index 1 in tool chain. Direction vector of the orientation axis with the index 1 in the tool chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoOriaxDirT2 (C, S)		$AC_TRAFO_ORIAX_DIR_T[2,n]		
Orientation direction index 2 in tool chain. Direction vector of the orientation axis with the index 2 in the tool chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoOriaxLoc (C, S)		$AC_TRAFO_ORIAX_LOC		
Position of an orientation axis in the kin. chain. The variable supplies the decimal-coded index of an orientation axis in the kinematic chain of an orientation transformation. The ten's place designates the part chain in which the orientation axis is included (0: Part chain; 1: Tool chain) and the unit's place the axis index when counting from the start to the end of the chain.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	-1	-3	12
Addressing				
Line index:			Max. line index:	

acTrafoOriaxLoc (C, S)	$AC_TRAFO_ORIAX_LOC	
	$$numMachAxes	
	Read access	

acTrafoSectionP0 (C, S)		$AC_TRAFO_SECTION_P[0,n]		
Section with the index 0 in the part chain. Section with the index 0 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoSectionP1 (C, S)		$AC_TRAFO_SECTION_P[1,n]		
Section with the index 1 in the part chain. Section with the index 1 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoSectionP2 (C, S)		$AC_TRAFO_SECTION_P[2,n]		
Section with the index 2 in the part chain. Section with the index 2 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoSectionP3 (C, S)		$AC_TRAFO_SECTION_P[3,n]		
Section with the index 3 in the part chain. Section with the index 3 in the part chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		

acTrafoSectionP3 (C, S)	$AC_TRAFO_SECTION_P[3,n]	
Addressing		
Line index:	Max. line index:	
	3	
	Read access	

acTrafoSectionT0 (C, S)		$AC_TRAFO_SECTION_T[0,n]		
Section with the index 0 in the tool chain. Section with the index 0 in the tool chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoSectionT1 (C, S)		$AC_TRAFO_SECTION_T[1,n]		
Section with the index 1 in the tool chain. Section with the index 1 in the tool chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoSectionT2 (C, S)		$AC_TRAFO_SECTION_T[2,n]		
Section with the index 2 in the tool chain. Section with the index 2 in the tool chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

acTrafoSectionT3 (C, S)		$AC_TRAFO_SECTION_T[3,n]		
Section with the index 3 in the tool chain. Section with the index 3 in the tool chain of an active orientation transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		

acTrafoSectionT3 (C, S)		$AC_TRAFO_SECTION_T[3,n]		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

actGrindingFrameIndex (C, S)		$P_GFRNUM		
Index of the actively set grinding frame Index of the actively set grinding frame. A grinding data management frame becomes an active grinding frame when executing GFRAME0 to GFRAME100.				
Return value	0 = GFRAME0 = No frame selected 1 = GFRAME1 to 100 = GFRAME100			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

progEvent (C, S)				
Active prog. events				
Return value	The data is bit-coded, so that when required, also individual states can be masked or evaluated separately (bits that are not listed supply a value of 0) Bit0 = 1: start Bit1 = 1: M30 Bit2 = 1: Reset Bit3 = 1: Power-On Bit4 = 1: Search Bit5 = 1: Safety			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

acSynaState (C, S)		$AC_SYNA_STATE		
Status of a synchronized action The status of a synchronized action can be read using the variable. The line index is the ID of the modal or static synchronized action, for which the status should be read.				
Return value	The data is bit-coded, so that when required, also individual states can be masked or separately evaluated (bits that are not listed supply a value of 0) Bit 0 = 0: No lock Bit 0 = 1: PLC or synchronize actions are locked Bit 1 = 0: PLC is not locked Bit 1 = 1: PLC is locked Bit 2 = 0: Synchronized action is not locked Bit 2 = 1: Synchronized action is locked			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
			Read access	

acMeasOriwks (C, S)		$AC_MEAS_ORIWKS		
Transformation behavior of the measuring interface The variable serves to define the frame transformation behavior of the measuring interface in respect of orientation axis coordinates.				
Return value	0: Orientation axis coordinates are transformed like ORIMKS 1: Orientation axis coordinates are transformed like ORIWKS 2: The frame transformation of the orientation axis coordinates depends on the active G code ORIMKS or ORIWKS.			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	2
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

pcTcarrOffset (C, S)		$PC_TCARR_OFFSET		
An offset vector of the active tool carrier A maximum of 4 offset vectors (l1 to l4) are defined for an active tool carrier. With this OPI variable, the vector components of the vector can be read out. If no tool carrier is active, each read access returns the value 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
mm, inch, degree, user defined	TYPE_DOUBLE	0	-DBL_MAX	DBL_MAX
Addressing				
Line index:			Max. line index:	

pcTcarrOffset (C, S)	$PC_TCARR_OFFSET	
	12	
	Read access	

pcTcarrAxVect (C, S)		$PC_TCARR_AX_VECT		
An axis vector of the active tool carrier A maximum of 2 axis vectors (v1 to v2) are defined for an active tool carrier. With this OPI variable, the vector components of the vector can be read out. If no tool carrier is active, each read access returns the value 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
mm, inch, degree, user defined	TYPE_DOUBLE	0	-DBL_MAX	DBL_MAX
Addressing				
Line index:			Max. line index:	
			6	
			Read access	

pcTcarrAxOffset (C, S)		$PC_TCARR_AX_OFFSET		
One axis offset of the active tool carrier A maximum of 2 axis offsets are defined for an active tool carrier. With this OPI variable, the rotary axis positions can be read out with initial setting. If no tool carrier is active, each read access returns the value 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
mm, inch, degree, user defined	TYPE_DOUBLE	0	-DBL_MAX	DBL_MAX
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

drawPosTriggerTime (C, S)		keine		
Time window for logging event DRAW_POS For very long movements or with stopped axes, the logging event DRAW_POS should nevertheless trigger now and again A specification of 0[s] deactivates the function. After control power on, a trigger time of 0.3[s] is preset. drawPosTriggerTime replaces the machine data 10690 DRAW_POS_TRIGGER_TIME (Reserved for the SIEMENS application Simultaneous recording)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.3		
			Read access and write access	

drawPosTriggerPath (C, S)		keine		
Path length for logging event DRAW_POS For very long movements, the logging event DRAW_POS should trigger at the latest after this specified path length A specification of 0.0[mm] deactivates the function. After control power on, a path length of 0.0[mm] is preset. (Reserved for the SIEMENS application Simultaneous recording)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

newMotSeqLength (C, S)		keine		
Path length for logging event NEW_MOT_SEQ For path lengths less than or equal to newMotSeqLength, the logging event NEW_MOT_SEQ triggers instead of the possible events NEW_MOT_LIN, NEW_MOT_CIRCLE or NEW_MOT_COMPLEX. A specification of 0.0[mm] deactivates the function. After control power on, a path length of 0.0[mm] is preset. (Reserved for the SIEMENS application Simultaneous recording)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

pcTrafoNum (C, S)		$PC_TRAFO_NUM		
Transformation number The variable is '0' if no transformation is active. If a conventionally defined transformation is active (in other words, not defined with kinematic chains), the variable contains the number of the actual transformation data record (the number n in the machine data $MC_TRAFO_TYPE_n). If a transformation defined with kinematic chains is active, the variable contains the number of the $NT data record with an offset of 1000, this means that the first transformation returns the value 1001.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acToolOCorrDir (C, S)	$AC_TOOL_O_CORR_DIR			
Overlay of the tool orientation Returns the direction vector of the plane between the programmed setpoint orientation and the current orientation with overlays. in various coordinate systems: Possible values of the line index: 1, 2, 3: Components of the vector in the BCS 4, 5, 6: Components of the vector in the PCS/WCS 7, 8, 9: Components of the vector in the ENS This direction vector is scaled, this means it has the value 1.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

acToolOCorrAngle (C, S)	$AC_TOOL_O_CORR_ANGLE			
Overlay of the tool orientation (angle). Returns the angle in degrees between the programmed setpoint orientation and the current orientation with overlays.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
Degree	TYPE_DOUBLE	0	0	180
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

acToolRCorrDir (C, S)	$AC_TOOL_R_CORR_DIR	
Overlay of the tool rotation Returns the direction vector of the plane between the programmed rotation of the orientation and the current rotation of the orientation with overlays. in various coordinate systems (only relevant with 6-axis kinematics). Possible values of the line index: 1, 2, 3: Components of the vector in the BCS 4, 5, 6: Components of the vector in the PCS/WCS 7, 8, 9: Components of the vector in the ENS This direction vector is scaled, this means it has the value 1.		

acToolRCorrDir (C, S)		$AC_TOOL_R_CORR_DIR		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-1	1
Addressing				
Line index:			Max. line index:	
			9	
			Read access	

acToolRCorrAngle (C, S)		$AC_TOOL_R_CORR_ANGLE		
Overlay of the tool rotation (angle). Returns the angle in degrees between the programmed rotation of the orientation and the current rotation of the orientation with overlays.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
Degree	TYPE_DOUBLE	0	0	180
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

selectedAsup (C, S)				
Assignment of NC program to interrupt channel Returns the assigned NC program (interrupt routine or ASUB) as complete path name for an interrupt channel via PI _N_ASUP__ or via part program command SETINT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			32	
			Read access	

tOffCR (C, S)		$AC_TOFFCR		
Tool corner radius offset TOFFCR Programmed tool corner radius offset.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

3.2.3.19 SE: Setting data

Channel-specific setting data

CONTPREC (C, SE)		SD 42450: $SC_CONTPREC		
SD 42450: $SC_CONTPREC				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

COUPLE_RATIO_1 (C, SE)		SD 42300: $SC_COUPLE_RATIO_1		
SD 42300: $SC_COUPLE_RATIO_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

CRIT_SPLINE_ANGLE (C, SE)		SD 42470: $SC_CRIT_SPLINE_ANGLE		
SD 42470: $SC_CRIT_SPLINE_ANGLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

CUTCOM_G40_STOPRE (C, SE)		SD 42490: $SC_CUTCOM_G40_STOPRE		
SD 42490: $SC_CUTCOM_G40_STOPRE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

EXT_PROG_PATH (C, SE)		SD 42700: $SC_EXT_PROG_PATH		
SD 42700: $SC_EXT_PROG_PATH				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

FRAME_OFFSET_INCR_PROG (C, SE)		SD 42440: $SC_FRAME_OFF-SET_INCR_PROG		
SD 42440: $SC_FRAME_OFFSET_INCR_PROG				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

IS_SD_MAX_PATH_ACCEL (C, SE)		SD 42502: $SC_IS_SD_MAX_PATH_AC-CEL		
SD 42502: $SC_IS_SD_MAX_PATH_ACCEL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

IS_SD_MAX_PATH_JERK (C, SE)		SD 42512: $SC_IS_SD_MAX_PATH_JERK		
SD 42512: $SC_IS_SD_MAX_PATH_JERK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

JOG_FEED_PER_REV_SOURCE (C, SE)		SD 42600: $SC_JOG_FEED_PER_REV_SOURCE		
SD 42600: $SC_JOG_FEED_PER_REV_SOURCE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MDD_DRY_RUN_FEED (C, SE)		SD 42100: $SC_MDD_DRY_RUN_FEED		
Alternative name:	DRY_RUN_FEED			
Dry run feedrate				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user de-fined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access and write access	

MDD_THREAD_START_ANGLE (C, SE)		SD 42000: $SC_MDD_THREAD_START_ANGLE		
Alternative name:	THREAD_START_ANGLE			
Starting angle for thread				

MDD_THREAD_START_ANGLE (C, SE)		SD 42000: $SC_MDD_THREAD_START_ANGLE		
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access and write access	

MINFEED (C, SE)		SD 42460: $SC_MINFEED		
SD 42460: $SC_MINFEED				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

NIBPUNCH_PRE_START_TIME (C, SE)		SD 42402: $SC_NIB-PUNCH_PRE_START_TIME		
SD 42402: $SC_NIBPUNCH_PRE_START_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

PUNCH_DWELLTIME (C, SE)		SD 42400: $SC_PUNCH_DWELLTIME		
SD 42400: $SC_PUNCH_DWELLTIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

SD_MAX_PATH_ACCEL (C, SE)		SD 42500: $SC_SD_MAX_PATH_ACCEL		
SD 42500: $SC_SD_MAX_PATH_ACCEL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

SD_MAX_PATH_JERK (C, SE)		SD 42510: $SC_SD_MAX_PATH_JERK		
SD 42510: $SC_SD_MAX_PATH_JERK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

SPIND_ASSIGN_TAB (C, SE)		SD 42800: $SC_SPIND_ASSIGN_TAB		
SD 42800: $SC_SPIND_ASSIGN_TAB				

SPIND_ASSIGN_TAB (C, SE)		SD 42800: $SC_SPIND_ASSIGN_TAB		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

STOP_CUTCOM_STOPRE (C, SE)		SD 42480: $SC_STOP_CUT-COM_STOPRE		
SD 42480: $SC_STOP_CUTCOM_STOPRE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

TOOL_OFFSET_INCR_PROG (C, SE)		SD 42442: $SC_TOOL_OFF-SET_INCR_PROG		
SD 42442: $SC_TOOL_OFFSET_INCR_PROG				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

3.2.3.20 SEGA: Extended state in the WCS

All status data that are dependent on machine movement and specified in the workpiece coordinate system are combined in module SGA. Supplementary information can be found in module SEGA. The individual variables are defined as arrays where the line index is the number of the axis (assigned to the current channel).The variable "name" in module SGA with the line index in question identifies the axis.

The assignment of the line indices in modules SGA and SEGA is identical.

With SW 5.2 and later, OPI modules SGA and SEGA can be addressed via the geo-axis no. instead of via the channel axis no.:

Line index 1001: 1st geo-axis

Line index 1002: 2nd geo-axis

Line index 1003: 3rd geo-axis

The number of channel axes (geometry, special axes and spindles) can be found in "$$numMachAxes" in module Y in area C.

aaDelt (C, SEGA)		$AA_DELT[x] x = Axis		
Stored axial dist.-to-go in WCS after DELDTG Stored axial distance-to-go in the WCS after axial delete-distance-to-go DELDTG(axis) for synchronous actions (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtbw (C, SEGA)		$AA_DTBW[x] x = Aaxis		
Axial distance from beginning of block in the WCS Axial distance from the beginning of the block in the WCS for positioning and synchronous axes for synchronous motion (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtepw (C, SEGA)		$AA_DTEPW[x] x = Axis		
Axial dist.-to-go for infeed during oscill. in WCS Axial distance-to-go for infeed during oscillation in the WCS (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtew (C, SEGA)		$AA_DTEW[x] x = Axis		
Axial distance to the end of block in the WCS Axial distance to the end of the block in the WCS for positioning and synchronous axes for synchronous actions (Note: for SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			

aaDtew (C, SEGA)		$AA_DTEW[x] x = Axis		
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtsb (C, SEGA)		$AA_DTSB		
Path from the motion starting point in the BCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			numMachAxes	
			Read access	

aaDtsw (C, SEGA)		$AA_DTSW		
Path from the motion starting point in the WCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			numMachAxes	
			Read access	

aaIb (C, SEGA)		$AA_IB		
BCS setpoint Current BCS setpoint of an axis				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			numMachAxes	
			Read access	

aaIbCorr (C, SEGA)	$AA_IB_CORR	
BCS setpoint value including override components Current BCS setpoint value of an axis including override components		

aaIbCorr (C, SEGA)		$AA_IB_CORR		
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			numMachAxes	
			Read access	

aaIwCorr (C, SEGA)		$AA_IW_CORR		
WCS setpoint value including override components Current WCS setpoint value of an axis including override components				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			numMachAxes	
			Read access	

aaMw1 (C, SEGA)		$AA_MW1[Achse]		
Access meas. result trigger event 1 in WCS Access to measurement result of trigger event 1 in the WCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMw2 (C, SEGA)		$AA_MW2[Achse]		
Access meas. result trigger event 2 in WCS Access to measurement result of trigger event 2 in the WCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMw3 (C, SEGA)		$AA_MW3[Achse]		
Access meas. result trigger event 3 in WCS Access to measurement result of trigger event 3 in the WCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMw4 (C, SEGA)		$AA_MW4[Achse]		
Access meas. result trigger event 4 in WCS Access to measurement result of trigger event 4 in the WCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMw (C, SEGA)		$AA_MW[x] x = Axis		
Latched probe position retransformed in the WCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaTOff (C, SEGA)		$AA_TOFF[ ]		
Value of tthe superimposition in the tool directio Value of the superimposed motions which have been retracted in the individual tool directions via $AA_TOFF[ ]				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	

aaTOff (C, SEGA)	$AA_TOFF[ ]	
	1000 + $$numGeoAxes	
	Read access and write access	

aaTOffLimit (C, SEGA)		$AA_TOFF_LIMIT[ ]		
Limiting value achieved via $AA_TOFF[ ] Limit value of the superimposed motion has been achieved in the tool direction via $AA_TOFF[ ]				
Return value	0 : Limiting value not achieved 1 : Limiting value achieved in positive direction 11 : Limiting value achieved in negative direction			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	11
Addressing				
Line index:			Max. line index:	
			1000 + $$numGeoAxes	
			Read access	

aaTOffPrepDiff (C, SEGA)		$AA_TOFF_PREP_DIFF[ ]		
Difference value main run to run in of $AA_TOFF[ ] Difference between the current value of $AA_TOFF[] and the value as the current block was prepared.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1000 + $$numGeoAxes	
			Read access	

aaTOffVal (C, SEGA)		$AA_TOFF_VAL[ ]		
Cumulative value of superimposition in the tool di Integrated value of the superimposed motions which have been retracted in the individual tool directions via $AA_TOFF[ ]				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1000 + $$numGeoAxes	
			Read access	

aaVactW (C, SEGA)		$AA_VACTW[X]		
Axis veloc. in workpiece coordinates Axis velocity in workpiece coordinate system				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

acRetpoint (C, SEGA)		$AC_RETPOINT[x] x = Axis		
Return point on the contour for repositioning				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actDistToGoEns (C, SEGA)				
Distance-to-go in the SZS Distance-to-go in the SZS based on the programmed position				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actFeedRate (C, SEGA)				S5
Actual value of axis-specific feedrate Actual value of axis-specific feedrate, if the axis is a positioning axis.				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

actFeedRate (C, SEGA)			S5
		$$numMachAxes	
Validity:	From NCK version 3.6	Read access	

actFeedRateIpo (C, SEGA)				
$$/C/SEGA/actFeedRate with revolutional feedrate				
Corresponds to $$/C/SEGA/actFeedRate taking into account the revolutional feedrate. For geometry axes the value is reported in the MCS, i.e. related to the geometry axis and not to the machine axis. Associated unit see: $$axisFeedRateIpoUnit				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actProgPosBKS (C, SEGA)				
Actual value in basic coordinate system				
Actual value of geometry and orientation axes in basic coordinate system				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actToolBasPosBN (C, SEGA)		$AA_IBN[x] x=Axis		
Current setpoint relative to the BZS				
Determines the current setpoint of the tool base position of axis "x" in relation to basic zero coordinate system (BZS) The BZS value does not contain any override components (DRF, $AA_OFF, ext. work offset, etc.).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actToolBasPosBNDiam (C, SEGA)				
$$actToolBasPosBN with diameter conv. Corresponds to $$actToolBasPosBN with diameter conversion				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actToolBasPosEN (C, SEGA)	$AA_IEN[x] x = Axis			
Current setpoint in the SZS coordinate system Determines the current setpoint of the "x" axis in the settable zero point coordinate system (SZS) for the corresponding axis. The SZS value does not contain any axial override components (DRF, AA_OFF, ext. work offset etc.).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actToolBasPosENitc (C, SEGA)				
corresponds to $$actToolBasPosEN Mode=1 corresponds to $$actToolBasPosEN with $DISPLAY_MODE_POSITION=1				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actToolBasPosENjmp (C, SEGA)				
corresponds to $$actToolBasPosEN Mode=0 corresponds to $$actToolBasPosEN with $DISPLAY_MODE_POSITION=0				
Units and value ranges				
Physical unit:	Data type:			

actToolBasPosENjmp (C, SEGA)				
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actToolBasePosBasic (C, SEGA)				
Base position of the active tool Base position of the active tool in the base system (inch/metrical)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actToolBasePosBasicDiam (C, SEGA)				
$$actToolBasePosBasic with diameter conv. Corresponds to $$actToolBasePosBasic with diameter conversion				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actToolBasePosDiam (C, SEGA)				
$$/C/SGA/actToolBasePos with diameter conv. Corresponds to $$/C/SGA/actToolBasePos with diameter conversion				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actToolEdgeCenterPosEns (C, SEGA)				
Actual position value in relation to WOS frame Actual position value in relation to the WOS frame as center-point path, i.e. with tool length but without tool radius				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

axisActiveInChan (C, SEGA)				
Axis active Flag indicating whether axis is active in this channel				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

axisFeedRateIpoUnit (C, SEGA)				
$$/C/SEGA/axisFeedRateUnit with revolut. feedrate Corresponds to $$/C/SEGA/axisFeedRateUnit with revolutional feedrate				
Return value	0 = mm/min 1 = mm/rev. 2 = inch/min 3 = inch/rev. 4 = deg./min 5 = deg./rev.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

axisFeedRateUnit (C, SEGA)				
Unit of axial feedrate				
Return value	0 = mm/min 1 = inch/min 2 = degree/min			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdFeedRate (C, SEGA)				
Desired value of axis-specific feedrate Desired value of axis-specific feedrate for a positioning axis.				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdFeedRateIpo (C, SEGA)				
$$/C/SEGA/cmdFeedRate with revolutional feedrate Corresponds to $$/C/SEGA/cmdFeedRate taking into account the revolutional feedrate. Associated unit see: $$axisFeedRateIpoUnit				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

cmdToolBasPosENS (C, SEGA)				
Workpiece coordinate system without progr. frame Workpiece coordinate system without programmable frame and cycles frame (see also MD24030 MC_FRAME_ACS_SET) Caution: If block search is active, only valid in a block search event of the logging function.				
Units and value ranges				
Physical unit:	Data type:			

cmdToolBasPosENS (C, SEGA)				
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

cmdToolEdgeCenterPosEnsS (C, SEGA)				
Programmed SZS position for block search Programmed SZS position for block search with calculation in relation to the WOS frame as center-point path, i.e. with tool length but without tool radius Notice! This variable is not available for the variable service, but only for logging in connection with block search events!				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

cmdToolEdgeCenterPosEns (C, SEGA)				
Programmed SZS position Programmed SZS position in relation to the WOS frame as center-point path, i.e. with tool length but without tool radius				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

diamonInfo (C, SEGA)				
Position as diameter or radius Information whether position values are shown as diameter or radius values. This information is relevant for the following variables of the blocks SGA/SEGA: - $$/C/SGA/cmdToolBasePos - $$/C/SGA/toolBaseDistToGo - $$/C/SGA/toolBaseREPOS - $$cmdToolEdgeCenterPos - $$actToolEdgeCenterPos - $$toolEdgeCenterDistToGo - $$toolEdgeCenterREPOS - $$cmdProgPos - $$actProgPos - $$progDistToGo - $$progREPOS - $$actToolBasPosEN - $$cmdToolEdgeCenterPosEnsS - $$/C/SEGA/actToolEdgeCenterPosEns - $$actToolBasPosBN - $$cmdToolBasPosENS - $$actProgPosBKS - $$actToolBasePosDiam - $$actToolBasePosBasicDiam - $$actToolBasPosBNDiam				
Return value	0: Diameter programming inactive 1: Diameter programming active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

displayAxis (C, SEGA)	$MC_DISPLAY_AXIS Bit0-15	
Identifier for geometry/auxiliary axis Identifier indicating whether the axis is displayed by the HMI as a geometry or auxiliary axis.		
Return value	0 = Do not display at all 0xFFFF = Always display everything bit 0 = Display in actual-value window bit 1 = Display in reference point window bit 2 = Display in Preset / Basic offset / Scratching bit 3 = Display in handwheel selection	

displayAxis (C, SEGA)		$MC_DISPLAY_AXIS Bit0-15		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0xFFFF	0	0xFFFF
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

drfVal (C, SEGA)		$AC_DRF[x] x = Axis		
DRF value DRF value. The physical unit is defined in $$/C/SGA/extUnit (in module SGA).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

effComp (C, SEGA)				
effComp Sum of all length radius compensations. The physical unit is defined in $$/C/SGA/extUnit (in the SGA block).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

feedRateOvr (C, SEGA)				
Feedrate override Feedrate override if axis is a positioning axis. Multiplying override component which is active in addition to the override factors programmed, set via handwheel or via PLC.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

geoAxisNr (C, SEGA)				
Number of the geometry axis				
Return value	If the axis is a geometry axis: 1-3 If the axis is not a geometry axis: 0			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

handwheelAss (C, SEGA)				
Number of assigned handwheel Number of handwheel assigned to axis				
Return value	0 = no handwheel assigned 1 to 3 = handwheel number			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	3
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

index (C, SEGA)				
Absolute axis index Absolute axis index referred to MD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

motEnd (C, SEGA)		$AA_MOTEND		
Motion end criterion for single-axis interpolation Current motion end criterion for single-axis interpolation				
Return value	1 = Motion end with exact stop FINE 2 = Motion end with exact stop COARSE 3 = Motion end with exact stop IPO Stop 4 = Block change in braking ramp of axis motion 5 = Block change in braking ramp of axis motion with tolerance window with reference to setpoint 6 = Block change in braking ramp of axis motion with tolerance window with reference to actual value			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	1	6
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

spec (C, SEGA)				
Axis specification				
Return value	0 = path axis 1 = positioning axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

subSpec (C, SEGA)		MD 30500: INDEX_AX_AS-SIGN_POS_TAB		T1
Subspecification Subspecification, identifies whether an axis is an indexing axis				
Return value	0 = normal axis 1 = indexing axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

type (C, SEGA)				
Axis type				
Return value	1 = linear axis 2 = rotary axis 3 = spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

aaDiamStat (C, SEGA)	$AA_DIAM_STAT[]			
Status of the diameter programming Status of the diameter programming as a function of configuration and programming				
Return value	Bit 0=0: Diameter programming inactive Bit 0=1: Diameter programming active Bit 1=0: Channel-specific diameter programming			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	15
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaSccStat (C, SEGA)	$AA_SCC_STAT[]			
Status of the G96/G961/G962 assignment Status of the G96/G961/G962 assignment as a function of configuration and programming				
Return value	Bit 0=0: Axis is not assigned to G96/G961/G962 Bit 0=1: Axis is assigned to G96/G961/G962			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaPcsRel (C, SEGA)		$AA_PCS_REL[Achse]		
Current relative WCS setpoint of an axis The axial variable $AA_PCS_REL[ax] determines the current relative setpoint of the corresponding axis in the workpiece coordinate system (WCS). The setpoint corresponds to $AA_IW[ax], which is transformed by the current relative system frame $P_RELFRAME. The axial positions lie in the relative workpiece coordinate system.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaAcsRel (C, SEGA)		$AA_ACS_REL[Achse]		
Current relative SZS setpoint of an axis The axial variable $AA_ACS_REL[ax] determines the current relative setpoint in the settable zero coordinate system (SZS) for the corresponding axis. The setpoint corresponds to $AA_IEN[ax], which is transformed by the current relative system frame $P_RELFRAME. The axial positions lie in the relative settable zero system.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaIbc (C, SEGA)		$AA_IBC[Achse]		
Current setpoint of a Cartesian axis The axial variable $AA_IBC[ax] determines the setpoint of a Cartesian axis lying between BCS and MCS. Cartesian means that the axis is a linear axis, and it lies plane parallel to a coordinate axis in a clockwise coordinate system. If a geometry axis is still Cartesian at the output of the nth transformation, then this value is returned. The axis identifier used must represent a geometry axis in the BCS, otherwise the variable returns the value 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

valw (C, SEGA)		$VA_IW[Achse]		
Current WCS actual value of an axis The variable $VA_IW[ax] determines the encoder position of an axis retransformed into BCS. The BCS value contains all the axial override components (DRF, AA_OFF, external work offset, etc.) and compensation values (CEC, etc.). The positions are only computed once per IPO cycle for performance reasons. When a variable is read within one IPO cycle, the value of the variable does not change, although the actual value could have changed. If transformations are active, it must be taken into account that transforming the actual values into the BCS in the IPO cycle can be very time-consuming. In this case, an adequate IPO cycle must be set.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

valb (C, SEGA)		$VA_IB[Achse]		
Current BCS encoder position of an axis The variable $VA_IB[ax] determines the encoder position of an axis retransformed into BCS. The BCS value contains all the axial override components (DRF, AA_OFF, external work offset, etc.) and compensation values (CEC, etc.). The positions are only computed once per IPO cycle for performance reasons. When a variable is read within one IPO cycle, the value of the variable does not change, although the actual value could have changed. If transformations are active, it must be taken into account that transforming the actual values into the BCS in the IPO cycle can be very time-consuming. In this case, an adequate IPO cycle must be set.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

valbc (C, SEGA)		$VA_IBC[Achse]		
Cartesian BCS encoder position of an axis The variable $VA_IBC[geo axis] determines the encoder position of a Cartesian axis lying between BCS and MCS. Cartesian means that the axis is a linear axis and lies plane parallel to a coordinate axis in a clockwise coordinate system. The axis identifier used can be a geometry, channel or machine axis identifier. This identifier must represent a geometry axis in the BCS, otherwise the variable returns the value 0.0. The positions are only computed once per IPO cycle for performance reasons. When a variable is read within one IPO cycle, the value of the variable does not change, although the actual value could have changed. If transformations are active, it must be taken into account that transforming the actual values into the BCS in the IPO cycle can be very time-consuming. In this case, an adequate IPO cycle must be set.				
Units and value ranges				
Physical unit:	Data type:	Initial value		

valbc (C, SEGA)		$VA_IBC[Achse]		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaltr1 (C, SEGA)		$AA_ITR[Achse, 1]		
Current setpoint after the first transformation The axial variable determines the current setpoint of an axis at the output of the 1st chained transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaltr2 (C, SEGA)		$AA_ITR[Achse, 2]		
Current setpoint after the second transformation The axial variable determines the current setpoint of an axis at the output of the 2nd chained transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaltr3 (C, SEGA)		$AA_ITR[Achse, 3]		
Current setpoint after the third transformation The axial variable determines the current setpoint of an axis at the output of the 3rd chained transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

valtr1 (C, SEGA)		$VA_ITR[Achse, 1]		
Actual value at the output of first transformation The axial variable determines the current encoder position of an axis at the output of the 1st chained transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

valtr2 (C, SEGA)		$VA_ITR[Achse, 2]		
Actual value at output of second transformation The axial variable determines the current encoder position of an axis at the output of the 2nd chained transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

valtr3 (C, SEGA)		$VA_ITR[Achse, 3]		
Actual value at the output of third transformation The axial variable determines the current encoder position of an axis at the output of the 3rd chained transformation.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

3.2.3.21 SEMA: Extended state data in MCS

All state data that are dependent on machine movement and are defined specifically for machine axes (geometry and special axes) are combined in module SMA. Supplementary information is to be found in module SEMA. The individual variables are defined as fields where the line index is the number of the machine axis (assigned to the current channel). The variable "name" in module SMA with the line index in question identifies the axis.

The assignment of the line indices in modules SMA and SEMA is identical.

PRESETActive (C, SEMA)				
Preset state				
Return value	0 = no preset active 1 = preset active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

PRESETVal (C, SEMA)		$AC_PRESET[x] x = Axis		
PRESET The function PRESETON (...) programs a work offset for an axis. The value of the offset is stored in the variable 'PRESETVal'. The variable can be overwritten by the part program and by the HMI.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access and write access	

aaAccPercent (C, SEMA)		$AA_ACC_PERCENT[Achse]		
Current acceleration value in percent Current acceleration value for single-axis interpolation in percent				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaActIndexAxPosNo (C, SEMA)		$AA_ACT_INDEX_AX_POS_NO[<Achse>]		
Current indexing position Current indexing position; the display depends on $MN_INDEX_AX_NO_MODE and the division (via table or equidistant)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaAlarmStat (C, SEMA)		$AA_ALARM_STAT		
Display of active alarms Display indicating whether alarms are active for a PLC-controlled axis. The relevant coded alarm reactions can be used as a source for the "Extended Stop and Retract" function.				
Return value	The data is bit-coded, allowing, where necessary, individual states to be masked or evaluated separately (bits not listed supply a value of 0) Bit2 = 1: NOREADY (active rapid deceleration + cancelation of servo enable) Bit6 = 1: STOPBYALARM (rampm stop in all channel axes) Bit9 = 1: SETVDI (VDI interface signal "Setting alarm") Bit13 = 1: FOLLOWUPBYALARM (Follow-up)			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaAxChangeStat (C, SEMA)		$AA_AXCHANGE_STAT[Achse]		
Axis status with respect to axis replacement Axis status with respect to axis replacement				
Return value	0: Axis can be replaced 1: Axis is linked to the channel, but can become the PLC, command or reciprocating axis 2: Axis cannot be replaced			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	

aaAxChangeStat (C, SEMA)	$AA_AXCHANGE_STAT[Achse]	
	$$numMachAxes	
	Read access	

aaAxChangeTyp (C, SEMA)	$AA_AXCHANGE_TYP[Achse]			
Axis type with respect to axis replacement				
Return value	0: Axis assigned to the NC program 1: Axis assigned to the PLC or active as command axis or reciprocating axis 2: Other channel has interpolation right 3: Neutral axis 4: Neutral axis controlled from the PLC 5: Other channel has interpolation right; axis is requested for the NC program 6: Other channel has interpolation right; axis is requested as neutral axis 7: Axis is PLC axis or is active as command axis or reciprocating axis; axis is requested for the NC program 8: Axis is PLC axis or is active as command axis or reciprocating axis; axis is requested as neutral axis			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	8
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaAxDisable (C, SEMA)	$AA_AX_DISABLE[<Achse>]			
Status of axis/spindle disable Resulting status of axis/spindle disable.				
Return value	0: Axis/spindle disable not active. 1: Axis/spindle disable active.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaAxDisableSrc (C, SEMA)	$AA_AX_DISABLE_SRC[<Achse>]			
Source of axis/spindle disable Bitmask which provides the status and source of the currently active axis/spindle disable. If bit 0 has been set, the axis/spindle disable is active.				
Return value	The data is bit-coded. Individual states can therefore be masked or evaluated separately if necessary: Bit0 = 1: Resulting state from all sources: axis/spindle disable active. Bit1 = 1: Axial signal 'Axis/spindle disable triggered by PLC' is active. Bit2 = 1: Channel-specific program test is active. Bit3 = 1: Axial suppression of 'Program test triggered by the PLC' is active. Bit4 = 1: Axial signal 'Program test (energy saving mode)' is active. Bit5 = 1: SERUPRO is active. Bit6 = 1: Coupling object 'Total state is axis/spindle disable' is active. Bit7 = 1: Coupling object 'Total state is real traveling' is active.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaBcsOffset (C, SEMA)	$AA_BCS_OFFSET[Achse]			
Axial offsets of an axis Sum of all axial offsets of an axis, such as DRF, online tool offset, $AA_OFF and ext. WO.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaBrakeCondB (C, SEMA)	$AA_BRAKE_CONDB[axis]			
Conditions for interpolator stop in the BCS Shows the pending braking requests (conditions) for the interpolator stop of the axis / spindle. A braking request consists of a collision direction relating to a coordinate axis in the BCS and a braking priority relating to the machining step. If the axis / spindle receives a current braking request on account of these requirement(s), bit 0 is set in $AA_BRAKE_STATE[X] (in the next IPO cycle).\				
Return value	The highest deceleration priority in positive direction is shown in bits 0 to 3: 0x0: No pending deceleration request 0x1: Priority 1 covers all positioning actions (G0, POS, SPOS) 0x2: Priority 2 covers DYNNORM and all priority 1 motions 0x3: Priority 3 covers DYNPOS and all priority 1 to 2 motions 0x4: Priority 4 covers DYNROUGH and all priority 1 to 3 motions 0x5: Priority 5 covers DYNSEMIFIN and all priority 1 to 4 motions 0x6: Priority 6 covers all motions (including DYNFINISH). The request could also have been triggered by a CP SW limit stop. 0x7: Priority 7 covers all motions. The request was triggered by the VDI interface signal DB31,..DBX4.3 "Feed stop / Spindle stop". Deceleration always takes place, irrespective of the direction of motion. 0xD: Priority 13 covers all motions. Axial deceleration takes place with an emergency stop deceleration ramp. The highest deceleration priority in negative direction is shown in bits 16 to 19: 0x0 to 0xD: Same meaning as bits 0 to 3 All other bits are reserved and not set. If the value of the variable is shown in hexadecimal format, the fifth character from the right shows the deceleration priority in negative direction while the first number from the right shows the deceleration priority in positive direction.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xD000D
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaBrakeCondM (C, SEMA)	$AA_BRAKE_CONDM[axis]			
Conditions for interpolator stop in the MCS Shows the pending braking requests (conditions) for the interpolator stop of the axis / spindle. A braking request consists of a collision direction relating to a coordinate axis in the MCS and a braking priority relating to the machining step.				
Return value	The highest deceleration priority in positive direction is shown in bits 0 to 3: 0x0: No pending deceleration request 0x1: Priority 1 covers all positioning actions (G0, POS, SPOS) 0x2: Priority 2 covers DYNNORM and all priority 1 motions 0x3: Priority 3 covers DYNPOS and all priority 1 to 2 motions 0x4: Priority 4 covers DYNROUGH and all priority 1 to 3 motions 0x5: Priority 5 covers DYNSEMIFIN and all priority 1 to 4 motions 0x6: Priority 6 covers all motions (including DYNFINISH). The request could also have been triggered by a CP SW limit stop. 0x7: Priority 7 covers all motions. The request was triggered by the VDI interface signal DB31,..DBX4.3 "Feed stop / Spindle stop". Deceleration always takes place, irrespective of the direction of motion. 0xD: Priority 13 covers all motions. Axial deceleration takes place with an emergency stop deceleration ramp. The highest deceleration priority in negative direction is shown in bits 16 to 19: 0x0 to 0xD: Same meaning as bits 0 to 3 All other bits are reserved and not set. If the value of the variable is shown in hexadecimal format, the fifth character from the right shows the deceleration priority in negative direction while the first number from the right shows the deceleration priority in positive direction.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xD000D
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaBrakeState (C, SEMA)	$AA_BRAKE_STATE[axis]			
Current braking status Returns for the axis / spindle whether an active deceleration request has been set on account of the request of aaBrake-CondB or a CP SW limit stop or a VDI interface signal DB31,..DBX4.3 "Feed stop / Spindle stop".				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	1
Addressing				
Line index:			Max. line index:	

aaBrakeState (C, SEMA)	$AA_BRAKE_STATE[axis]	
	$$numMachAxes	
	Read access	

aaChanNo (C, SEMA)		$AA_CHANNO[Achse]		
Axis in the channel The variable supplies the number of the channel in which the axis is currently being interpolated. With value 0, the axis could not be assigned to any channel.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCoupAct (C, SEMA)		$AA_COUP_ACT[x] x = Spindle following		
Current coupling state of the slave spindle				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCoupOffs (C, SEMA)		$AA_COUP_OFFS[x] x = Spindle		
Pos. offset of the synchr. spindle desired value Position offset of the synchronous spindle desired value				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCurr (C, SEMA)		$AA_CURR[x] x = Axis		
Actual value of the axis/spindle current in A Actual value of the axis/spindle current in A (only available for PROFIdrive drives)				
Units and value ranges				
Physical unit:	Data type:			
A	TYPE_DOUBLE			

aaCurr (C, SEMA)	$AA_CURR[x] x = Axis	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

aaDepAxO (C, SEMA)		$AA_DEPAXO[Achse]		
Dependence on other axes Dependency on other axes. Returns an axis code for the defined axis AX containing all the machine axes that have a mechanical dependency on the defined axis.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtbb (C, SEMA)		$AA_DTBB[x] x = Axis		
Axis distance from beginning of the block in BCS Axis-specific distance from the beginning of the block in the BCS for positioning and synchronous axes used in synchronous actions (note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtbreb (C, SEMA)		$AA_DTBREB[axis]		
Total decelaration distance, BCS The estimated total distance until the end of deceleration is reached, BCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtbrebCmd (C, SEMA)		$AA_DTBREB_CMD[axis]		
Command share of decel. dist. of axis ax in BCS Command share of the overall deceleration distance of axis ax in the BCS. The value is the estimated deceleration distance of the axis up to standstill.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtbrebCorr (C, SEMA)		$AA_DTBREB_CORR[axis]		
Offset section of the deceleration distance, BCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtbrebDep (C, SEMA)		$AA_DTBREB_DEP[axis]		
Dependent section of the decelaration dist., BCS Dependent section of the decelaration distance, BCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtbrem (C, SEMA)		$AA_DTBREM[axis]		
Total decelaration distance, MCS The estimated total distance until the end of deceleration is reached, MCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtbremCmd (C, SEMA)		$AA_DTBREM_CMD[axis]		
Spec. section of the decelaration distance, MCS Specified section of the decelaration distance, MCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtbremCorr (C, SEMA)		$AA_DTBREM_CORR[axis]		
Offset section of the deceleration distance, MCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDtbremDep (C, SEMA)		$AA_DTBREM_DEP[axis]		
Dependent section of the decelaration dist., MCS Dependent section of the decelaration distance, MCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDteb (C, SEMA)		$AA_DTEB[x] x = Axis		
Axis distance to the end of the block in the BCS Axis-specific distance to the end of the block in the BCS for positioning and synchronous axes used in synchronous actions (note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

aaDteb (C, SEMA)	$AA_DTEB[x] x = Axis	
	$$numMachAxes	
	Read access	

aaDtepb (C, SEMA)		$AA_DTEPB[x] x = Axis		
Axis dist-to-go of infeed during oscill. in BCS Axis-specific distance-to-go of infeed during oscillation in the BCS (note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaEnc1Active (C, SEMA)		$AA_ENC1_ACTIVE[Achse]		
First measuring system is active				
Return value	0: Measuring system is not active 1: Measuring system is active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaEnc2Active (C, SEMA)		$AA_ENC2_ACTIVE[Achse]		
Second measuring system is active				
Return value	0: Measuring system is not active 1: Measuring system is active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaEncActive (C, SEMA)		$AA_ENC_ACTIVE[Achse]		
Measuring system is active				
Return value	0: Measuring system is not active 1: Measuring system is active			

aaEncActive (C, SEMA)		$AA_ENC_ACTIVE[Achse]		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaEsrEnable (C, SEMA)		$AA_ESR_ENABLE[Achse]		
ESR axis enable (Axial) enabling of reactions of "Extended Stop and Retract" function. The selected axial ESR reaction must be parameterized in MD $MA_ESR_REACTION. beforehand. The corresponding Stop or Retract reactions can be activated via $AN_ESR_TRIGGER (or for individual drives in the event of communications failure/ DC-link undervoltage), generator-mode operation is automatically activated in response to undervoltage conditions.				
Return value	0: FALSE 1: TRUE			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaEsrStat (C, SEMA)		$AA_ESR_STAT[Achse]		
ESR axis status (Axial) status checkback signals of "Extended Stop and Retract" function, which can be applied as input signals for the gating logic of the ESR (synchronous actions).				
Return value	The data is bit-coded. Individual states can therefore be masked or evaluated separately if necessary: Bit0 = 1: Generator mode is activated Bit1 = 1: Retract operation is activated Bit2 = 1: Stop operation is activated Bit3 = 1: Risk of undervoltage (DC-link voltage monitoring, voltage has dropped below warning threshold) Bit4 = 1: Speed has dropped below minimum generator mode threshold (i.e. no more regenerative rotation energy is available).			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		

aaEsrStat (C, SEMA)	$AA_ESR_STAT[Achse]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

aaEsrTrigger (C, SEMA)		$AA_ESR_TRIGGER		
Activation of ESR Activation of "NC-controlled ESR" for PLC-controlled axis				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaFixPointSelected (C, SEMA)		$AA_FIX_POINT_SELECTED[<Achse>]		
Selected fixed point Selected fixed point: Number of the fixed point that is to be approached				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaIbnCorr (C, SEMA)		$AA_IBN_CORR[<Achse>]		
BZS setpoint value (C) incl. override components Current BZS setpoint value of an axis including override components				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaIenCorr (C, SEMA)		$AA_IEN_CORR[<Achse>]		
SZS setpoint value (C) incl. override components Current SZS setpoint value of an axis including override components				
Units and value ranges				
Physical unit:	Data type:	Initial value		

aaIenCorr (C, SEMA)		$AA_IEN_CORR[<Achse>]		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaInSync (C, SEMA)		$AA_IN_SYNC[Achse]		
Synchronization status of the following axis Synchronization status of the following axis with master value coupling and ELG				
Return value	0: Synchronization is not running 1: Synchronization is running, i.e. following axis is being synchronized			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaInposStat (C, SEMA)		$AA_INPOS_STAT[Achse]		
Status for the programmed position				
Return value	0: No status available (axis/spindle is outside of the programmed position) 1: Travel motion pending 2: Position setpoint reached 3: Position reached with 'exact stop coarse' 4: Position reached with 'exact stop fine'			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	4
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaJerkCount (C, SEMA)		$AA_JERK_COUNT[Achse]		
Total traverse processes of an axis with jerk				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaJerkTime (C, SEMA)		$AA_JERK_TIME[Achse]		
Total traverse time of an axis with jerk				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
s, user defined	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaJerkTotal (C, SEMA)		$AA_JERK_TOT[Achse]		
Overall total jerk of an axis				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaLeadP (C, SEMA)		$AA_LEAD_P[x] x = Axis		
Actual lead value position				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaLeadPTurn (C, SEMA)		$AA_LEAD_P_TURN		
Current master value Current master value - position component lost as a result of modulo reduction				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaLeadSp (C, SEMA)		$AA_LEAD_SP[x] x = Axis		
Simulated lead value - position				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaLeadSv (C, SEMA)		$AA_LEAD_SV[x] x = Axis		
Simulated leading value velocity				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaLeadV (C, SEMA)		$AA_LEAD_V[x] x = Axis		
Actual lead value velocity Actual lead value - velocity				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaLoad (C, SEMA)		$AA_LOAD[x] x = Axis		
Drive load in % Drive load in % (only available for PROFIdrive drives)				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaMaslState (C, SEMA)		$AA_MASL_STAT		
Current status of the master-slave coupling Each slave axis currently coupled via master-slave delivers the machine axis number of the corresponding master axis. Zero is displayed as default for inactive coupling. A master axis also shows default value zero.				
Return value	0: No coupling for this axis configured, or axis is master axis, or no coupling active >0: Machine axis number of the master axis with which the slave axis is currently coupled			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$numMachAxes
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaMeaAct (C, SEMA)		$AA_MEAACT[Achse]		
Axial measuring active				
Return value	0: Measuring system is not active 1: Measuring system is active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaMm1 (C, SEMA)		$AA_MM1[x] x = Axis		
Access meas. result trigger event 1 in MCS Access to measurement result of trigger event 1 in the MCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMm2 (C, SEMA)		$AA_MM2[x] x = Axis		
Access meas. result trigger event 2 in MCS Access to measurement result of trigger event 2 in the MCS				

aaMm2 (C, SEMA)		$AA_MM2[x] x = Axis		
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMm3 (C, SEMA)		$AA_MM3[x] x = Axis		
Access meas. result trigger event 3 in MCS Access to measurement result of trigger event 3 in the MCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMm4 (C, SEMA)		$AA_MM4[x] x = Axis		
Access meas. result trigger event 4 in MCS Access to measurement result of trigger event 4 in the MCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMm (C, SEMA)		$AA_MM[x] x = Axis		
Latched probe position in the MCS Latched probe position in the machine coordinate system				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaOff (C, SEMA)		$AA_OFF[x] x = Axis		
Position offset from synchronous actions Superimposed position offset from synchronous actions				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaOffLimit (C, SEMA)		$AA_OFF_LIMIT[x] x = Axis		
Limit for axial correction $AA_OFF reached Limit for axial correction $AA_OFF reached (Note: for SYNACT only)				
Return value	0: Limit value not reached 1: Limit value reached in positive axis direction 11: Limit value reached in negative axis direction			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaOffVal (C, SEMA)		$AA_OFF_VAL[x]		
Total value of overlaid motion Integrated value of overlaid motion for an axis. The negative value of this variable can be used to cancel an overlaid motion. e.g. $AA_OFF[axis] = -$AA_OFF_VAL[axis]				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaOnFixPoint (C, SEMA)	$AA_FIX_ON_POINT[<Achse>]	
Fixed point reached Number of the fixed point at which the axis stands		

aaOnFixPoint (C, SEMA)		$AA_FIX_ON_POINT[<Achse>]		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaOscillBreakPos1 (C, SEMA)		$AA_OSCILL_BREAK_POS1[<Achse>]		
Oscillation interrupt position 1				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaOscillBreakPos2 (C, SEMA)		$AA_OSCILL_BREAK_POS2[<Achse>]		
Oscillation interrupt position 2				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaOscillReversePos1 (C, SEMA)		$AA_OSCILL_REVERSE_POS1[x] x = Axis		
Current reverse position 1 for oscill. in the BCS Current reverse position 1 for oscillation in the BCS. For synchronous actions the value of the setting data $SA_OSCILL_REVERSE_POS1 is evaluated online; (note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaOscillReversePos2 (C, SEMA)		$AA_OSCILL_REVERSE_POS2[x] x = Axis		
Current reverse position 2 for oscill. in the BCS Current reverse position 2 for oscillation in the BCS; For synchronous actions the value of the setting data $SA_OSCILL_REVERSE_POS1 is evaluated online; (note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaOvr (C, SEMA)		$AA_OVR[x] x = Axis		
Axial override for synchronous actions				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaPlcOvr (C, SEMA)		$AA_PLC_OVR[Achse]		
PLC override for motion-synchronous actions Axial override specified by PLC for motion-synchronous actions				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	100	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaPolfa (C, SEMA)		$AA_POLFA		
Retraction position of the single axis The programmed retraction position of the single axis				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

aaPolfa (C, SEMA)	$AA_POLFA	
	$$numMachAxes	
	Read access	

aaPolfaValid (C, SEMA)	$AA_POLFA_VALID			
Retraction of the single axis is programmed States whether the retraction of the single axis is programmed				
Return value	0: No retraction programmed for the single axis 1: Retraction programmed as position 2: Retraction programmed as distance			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaPower (C, SEMA)	$AA_POWER[x] x = Axis			
Drive power in W (611D only) Drive power in W (only available for PROFIdrive drives)				
Units and value ranges				
Physical unit:	Data type:			
W	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaProgIndexAxPosNo (C, SEMA)	$AA_PROG_INDEX_AX_POS_NO[Achse]			
Programmed indexing position				
Return value	0: No indexing axis, therefore no indexing position available >0: Number of the programmed indexing position			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaRef (C, SEMA)		$AA_REF[Achse]		
Axis is referenced				
Return value	0: Axis is not referenced 1: Axis is referenced			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaReposDelay (C, SEMA)		$AA_REPOS_DELAY[Achse]		
REPOS suppression active				
Return value	0: REPOS suppression is currently not active for this axis 1: REPOS suppression is currently active for this axis			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaScPar (C, SEMA)		$AA_SCPAR[Achse]		
Current setpoint parameter set				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaSnglAxStat (C, SEMA)		$AA_SNGLAX_STAT		
Single axis status Display status of a PLC-controlled axis				
Return value	0: Not a single axis 1: Reset 2: Ended 3: Interrupted 4: Active 5: Alarm			

aaSnglAxStat (C, SEMA)		$AA_SNGLAX_STAT		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaSoftendn (C, SEMA)		$AA_SOFTENDN[x] x = Axis		
Software end position, negative direction				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaSoftendp (C, SEMA)		$AA_SOFTENDP[x] x = Axis		
Software end position, positive direction				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaStat (C, SEMA)		$AA_STAT[]		
Axis status Axis state 0: no axis state available 1: travel command is active 2: axis has reached the IPO end. only for channel axes 3: axis in position (exact stop coarse) for all axes 4: axis in position (exact stop fine) for all axes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaSync (C, SEMA)	$AA_SYNC[x] x = Axis			
Status slave axis with lead value coupling Coupling status of the following axis with master value coupling The actual-value synchronism difference is analyzed to determine the coupling status. See also $VA_SYNCDIFF				
Return value	0: No synchronism 1: Synchronism coarse 2: Synchronism fine 3: Synchronism coarse and fine			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaSyncDiffStat (C, SEMA)	$AA_SYNCDIFF_STAT[Achse]			
Status of the synchronism difference Status of the setpoint synchronism difference				
Return value	-4: No valid value in $$aaSyncDiff, coupled motion from part program -3: Reserved -2: Reserved -1: No valid value in $$aaSyncDiff 0: No valid value in $$aaSyncDiff, coupling not active 1: Valid value in $$aaSyncDiff			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-4	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaTorque (C, SEMA)	$AA_TORQUE[x] x = Axis			
Desired torque value in Nm Drive torque setpoint in Nm (only available for PROFIdrive drives)				
Units and value ranges				
Physical unit:	Data type:			
Nm	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaTotalOvr (C, SEMA)		$AA_TOTAL_OVR[Achse]		
Total axial override f. motion-synchronous actions The total axial override for motion-synchronous actions				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	100	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaTravelCount (C, SEMA)		$AA_TRAVEL_COUNT[Achse]		
Total traverse processes of an axis				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaTravelCountHS (C, SEMA)		$AA_TRAVEL_COUNT_HS[Achse]		
Total traverse processes of an axis at high speed				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaTravelDist (C, SEMA)		$AA_TRAVEL_DIST[Achse]		
Total travel path of an axis Total travel path of an axis in mm or degrees				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaTravelDistHS (C, SEMA)		$AA_TRAVEL_DIST_HS[Achse]		
Total travel path of an axis at high speed Total travel path of an axis at high speed in mm or degrees				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaTravelTime (C, SEMA)		$AA_TRAVEL_TIME[Achse]		
Total traverse time of an axis Total traverse time of an axis in seconds				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
s, user defined	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaTravelTimeHS (C, SEMA)		$AA_TRAVEL_TIME_HS[Achse]		
Total traverse time of an axis at high speed Total traverse time of an axis at high speed in seconds				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
s, user defined	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaTyp (C, SEMA)	$AA_TYP[x] x = Axis			
Axis type according to $AA_TYP Axis type				
Return value	0: axis in other channel 1: channel axis of same channel 2: neutral axis 3: PLC axis 4: reciprocating axis 5: neutral axis, currently traversing in JOG 6: slave axis coupled via master value 7: coupled motion slave axis 8: command axis 9: compile cycle axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaType (C, SEMA)	$AA_TYPE[Achse]			
Cross-channel axis type Cross-channel axis type				
Return value	0: Axis type cannot be determined 1: NC program axis 2: Neutral axis 3: PLC axis 4: Reciprocating axis 5: Neutral axis that is currently executing a JOG or homing motion 6: Following axis coupled to the master value 7: Coupled motion of the following axis, activated in a synchronized action 8: Command axis 9: Compile Cycle axis 10: Coupled slave axis (master-slave function.) 11: Program axis that is currently executing a JOG or homing motion			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	11
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaVactB (C, SEMA)		$AA_VACTB[X]		
Axis veloc. in basic coordinates Axis velocity in basic coordinate system				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaVactM (C, SEMA)		$AA_VACTM[X]		
Axis veloc. in machine coordinates Axis velocity in machine coordinate system				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaVc (C, SEMA)		$AA_VC[x] x = Axis		
Add. correction value for path feed or axial feed Additive correction value for path feed or axial feed				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

acRpValid (C, SEMA)		$AC_RPVALID[Achse]		
Reapproach position valid Reapproach position valid				
Return value	0: Reapproach position not valid 1: Reapproach position valid			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1

acRpValid (C, SEMA)	$AC_RPVALID[Achse]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

ackSafeMeasPos (C, SEMA)				
Confirmation of SI actual position				
Return value	0 = not confirmed 0x00AC = confirmed			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access and write access	

actCouppPosOffset (C, SEMA)	$VA_COUP_OFFS[x] x = Axis			S3
Offset to leading axis/spindle, actual value Position offset of an axis to a leading axis / leading spindle (actual value)				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
mm, inch, degree, user defined	TYPE_DOUBLE		0	360
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

actFeedRate (C, SEMA)				S5
Actual value of axis-specific feedrate Actual value of axis-specific feedrate, if the axis is a positioning axis.				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

actIndexAxPosNo (C, SEMA)				
Current indexing position number Current indexing position number 0 = no indexing position >0 = indexing position number				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

actSpeedRel (C, SEMA)				
Actual value of rotary speed Actual value of rotary speed (referring to the maximum speed in %), for linear drives actual value of the velocity.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

actValResol (C, SEMA)				
Actual value of the resolution Actual value resolution. The physical unit is defined in measUnit (in this module)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

amSetupState (C, SEMA)				
State var. PI Service auto. set-up of asyn. motor State variable of the PI Service Automatic set-up of an asynchronous motor				
Return value	0 = inactive 1 = wait for PLC enable 2 = wait for key NC-start 3 = active 4 = stopped by Servo + fine code in the upper byte 5 = stopped by 611D + fine code in the upper byte 6 = stopped by NCK + fine code in the upper byte			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0xff06
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

axComp (C, SEMA)				
Sum of compensation values Sum of axis-specific compensation values (CEC Cross Error compensation and temperature compensation). The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

axisActiveInChan (C, SEMA)				
Axis active Flag indicating whether axis is active in this channel				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

axisFeedRateUnit (C, SEMA)				
Unit of axial feedrate				
Return value	0 = mm/min 1 = inch/min 2 = degree/min			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

chanAxisNoGap (C, SEMA)				
Axis exists Display of existing axis, i.e. no axis gap in channel.				
Return value	0: Axis does not exist 1: Axis does exist			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

chanNoAxisIsActive (C, SEMA)				
Channel number of axis Channel number in which the channel axis is currently active				
Return value	0 = axis is not assigned to any channel 1 to $$maxnumChannels (Area.:N / Module:Y) = channel number			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdContrPos (C, SEMA)				
Desired value of position after fine interpolation				
Units and value ranges				
Physical unit:	Data type:			

cmdContrPos (C, SEMA)				
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdCouppPosOffset (C, SEMA)	$AA_COUP_OFFS[x] x = Axis			S3
Position offset referring to leading axis/spindle Position offset of an axis referring to the leading axis / leading spindle (desired value)				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
mm, inch, degree, user defined	TYPE_DOUBLE		0	360
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdFeedRate (C, SEMA)				
Desired value of axis-specific feedrate Desired value of axis-specific feedrate for a positioning axis.				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdSpeedRel (C, SEMA)				
Desired value of rotary speed. Speed setpoint (as % of the maximum speed), velocity setpoint in the case of linear motors.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

contrConfirmActive (C, SEMA)				
Controller enable				
Return value	0 = no controller enable 1 = controller enable			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

contrMode (C, SEMA)				
Controller mode servo Identifier for controller mode servo				
Return value	0 = position control 1 = speed control 2 = stop 3 = park 4 = follow-up (set the mode through VDI interface and partly through part program)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

displayAxis (C, SEMA)		$MC_DISPLAY_AXIS Bit16-31		
Machine axis identifier Identifier indicating whether axis is displayed by HMI as a machine axis.				
Return value	0 = Do not display at all 0xFFFF = Always display everything bit 0 = Display in actual-value window bit 1 = Display in reference point window bit 2 = Display in Preset / Basic offset / Scratching bit 3 = Display in handwheel selection			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0xFFFF	0	0xFFFF
Addressing				
Line index:			Max. line index:	

displayAxis (C, SEMA)	$MC_DISPLAY_AXIS Bit16-31	
	$$numMachAxes	
	Read access	

distPerDriveRevol (C, SEMA)				
Distance per revolution of the drive Rotary drive: Load-side path corresponding to one revolution of the drive. Is returned in the unit of the internal computational resolution INT_INCR_PER_MM (for linear axes) or INT_INCR_PER_DEG (for rotary axes / spindles) taking into account gear ratios etc. In the case of linear axes, the pitch of the ball screw is also included in the calculation. In the case of linear motors, a fixed value of "1mm" is used for the ball screw pitch instead of the non-existent ball screw.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

drfVal (C, SEMA)				
DRF value				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

drive2ndTorqueLimit (C, SEMA)				
2nd torque limit 2nd torque limit. With linear motors: 2nd force limit				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveActMotorSwitch (C, SEMA)				
Actual motor wiring Actual motor wiring (star/delta)				
Return value	0 = star 1 = delta			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveActParamSet (C, SEMA)				
Number of the actual drive parameter set				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		1	8
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveClass1Alarm (C, SEMA)				
Message ZK1 drive alarm				
Return value	0 = no alarm set 1= alarm set (fatal error occured)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveContrMode (C, SEMA)				
Control mode of drive				
Return value	0 = current control 1 = speed control			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

driveContrMode (C, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveCoolerTempWarn (C, SEMA)				
Heatsink temperature monitoring				
Return value	0 = temperature OK 1 = overtemperature			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveDesMotorSwitch (C, SEMA)				
Motor wiring selection (star/delta)				
Return value	0 = star 1 = delta			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveDesParamSet (C, SEMA)				
Desired parameter set of the drive				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		1	8
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveFastStop (C, SEMA)				
Ramp-function generator rapid stop				
Return value	0 = not stopped 1 = stopped			

driveFastStop (C, SEMA)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveFreqMode (C, SEMA)				
I/F mode				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveImpulseEnabled (C, SEMA)				
Enable inverter impulse Enable inverter impulse (checkback signal to impulseEnable)				
Return value	0 = not enabled 1 = enabled			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveIndex (C, SEMA)				
Drive assignment (driveIndex) Drive assignment (logical drive number)				
Return value	0 = drive does not exist 1 to 15 = logical drive number			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	15
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveIntegDisable (C, SEMA)				
Integrator disable				
Return value	0 = not disabled 1 = disabled			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveLinkVoltageOk (C, SEMA)				
State of the DC link voltage				
Return value	0 = OK 1 = not OK			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveMotorTempWarn (C, SEMA)				
Motor temperature warning				
Return value	0 = temperature OK 1 = overtemperature			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveNumCrcErrors (C, SEMA)				
CRC errors on the drive bus CRC errors on the drive bus (Transmission errors when writing data to the drive; values may range up to FFFFH)				
Return value	0 = no error			
Units and value ranges				
Physical unit:	Data type:			

driveNumCrcErrors (C, SEMA)				
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveParked (C, SEMA)				
Parking axis				
Return value	0 = no parking axis 1 = parking axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

drivePowerOn (C, SEMA)				
Drive switched on				
Return value	0 = drive not switched on 1 = drive switched on			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveProgMessages (C, SEMA)				
Configurable messages Configurable messages (via machine data)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveReady (C, SEMA)				
Drive ready				
Return value	0 = drive not ready 1 = drive ready			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveRunLevel (C, SEMA)				
Current state reached during the boot process Current state reached during the boot process (range: coarse state (0 to 5) * 100 + fine state (up to 22)				
Return value	Booting the firmware ---> 0 XX entering the configuration ---> 1XX hardware-init, communication-init loading, converting data ---> 2XX changing bus addressing ---> 3XX preparing synchronization ---> 4XX activating interrupt ---> 519 XX ==> fine state			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveSetupMode (C, SEMA)				
Set-up mode				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

driveSetupMode (C, SEMA)				
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

driveSpeedSmoothing (C, SEMA)				
Smoothing the desired value of the rotary speed Smoothing the desired value of the rotary speed, for linear drives: smoothing the desired value of the velocity				
Return value	0 = no smoothing 1 = smoothing			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

effComp1 (C, SEMA)				
Sum of the compensation values for encoder 1 Sum of the compensation values for encoder 1. The value results from: temperature compensation, backlash compensation, quadrant error compensation, beam sag compensation, leadscrew error compensation. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

effComp2 (C, SEMA)				
Sum of the compensation values for encoder 2 Sum of the compensation values for encoder 2. The value results from: temperature compensation, backlash compensation, quadrant error compensation, beam sag compensation, leadscrew error compensation. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

encChoice (C, SEMA)				
Active encoder				
Return value	0 = does not exist 1 = encoder 1 2 = encoder 2			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

fctGenState (C, SEMA)				
State of the function generator				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

feedRateOvr (C, SEMA)				
Feedrate override Feedrate override (only if axis is a positioning axis)				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

focStat (C, SEMA)		$AA_FOC[x]		
Status of ForceControl function Current status of "Travel with limited torque" function				
Return value	0-2 0: FOC not active 1: FOC modal active (programming of FOCON[]) 2: FOC non-modal active (programming of FOC[])			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

focStat (C, SEMA)		$AA_FOC[x]		
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

fxsInfo (C, SEMA)		$VA_FXS_INFO[Achse]		
Additional info on travel to fixed stop Additional information on travel to fixed stop if $VA_FXS[]=2, or OPI variable $$/C/SEMA/fxsStat=2.				
Return value	0 No additional information available 1 No approach motion programmed 2 Programmed end position reached, movement ended 3 Abort by NC RESET (Reset key) 4 Fixed stop window exited 5 Torque reduction was rejected by drive 6 PLC has canceled enable signals			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

fxsStat (C, SEMA)		$AA_FXS[x] x = Axis		
State after travelling to fixed stop				
Return value	0 = normal control 1 = fixed stop reached 2 = selection failed			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

handwheelAss (C, SEMA)			
Number of assigned handwheel Number of handwheel assigned to axis			
Return value	0 = no handwheel assigned 1 to 3 = handwheel number		

handwheelAss (C, SEMA)				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	3
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

impulseEnable (C, SEMA)				
Impulse enable for drive				
Return value	0 = not enabled 1 = enabled			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

index (C, SEMA)				
Absolute axis index Absolute axis index referred to MD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

isDriveUsed (C, SEMA)				
Machine axis has control over the drive One or more machine axes are assigned to each drive. The drive can only be controlled at any one time by one of these machine axes. The machine manufacturer makes the selection. The status of the drive control changes dynamically.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	

isDriveUsed (C, SEMA)				
			$$numMachAxes	
			Read access	

kVFactor (C, SEMA)				
position control gain factor				
Units and value ranges				
Physical unit:	Data type:			
16.667 1/s	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

lag (C, SEMA)				
Following error Following error = desired value of position after fine interpolation - actual value of position. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

logDriveNo (C, SEMA)				
Drive assignment (logDriveNo) Drive assignment (logical drive number)				
Return value	0 = not available 1 to 15 = drive number			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	15
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

measFctState (C, SEMA)				
State of the probing function				
Units and value ranges				
Physical unit:	Data type:			

measFctState (C, SEMA)				
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

measPos1 (C, SEMA)				
Actual position value for measuring system 1. Actual value of position for encoder 1. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

measPos2 (C, SEMA)				
Actual position value for measuring system 2. Actual value of position for encoder 2. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

measPosDev (C, SEMA)				
Actual position difference between 2 encoders Actual position difference between the two encoders. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

measUnit (C, SEMA)				
Unit for service values of the drives				
Return value	0 = mm 1 = inch 2 = grd			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

paramSetNo (C, SEMA)				
Number of parameter set				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		1	8
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

preContrFactTorque (C, SEMA)				
Feed forward control factor torque				
Units and value ranges				
Physical unit:	Data type:			
Nm	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

preContrFactVel (C, SEMA)				
Feed forward control factor velocity				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

preContrMode (C, SEMA)				
Feed forward control mode				
Return value	0 = inactive 1 = velocity feed forward 2 = torque feed forward			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

progIndexAxPosNo (C, SEMA)				
Indexing number Programmed indexing position number 0 = no indexing position >0 = indexing position number				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

qecLrnIsOn (C, SEMA)				
Quadrant error compensation learning active				
Return value	0 = inactive 1 = Neuronal-QEC learning active 2 = Standard-QEC active 3 = Standard-QEC with adaptation of correction value active 4 = Neuronal-QEC active 5 = Neuronal-QEC with adaptation of measuring time active 6 = Neuronal-QEC with adaptation of decay time of correction value active 7 = Neuronal-QEC with adaptation of measuring time and decay time of correction value active			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	7
Addressing				
Line index:			Max. line index:	

qecLrnIsOn (C, SEMA)				
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

refPtBusy (C, SEMA)				
Axis is being referenced				
Return value	0 = axis is not being referenced 1 = axis is being referenced			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

refPtCamNo (C, SEMA)				
Reference point cam				
Return value	0 = no cam approached 1 = cam 1 2 = cam 2 3 = cam 3 4 = cam 4			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

refPtPhase (C, SEMA)				
Referencing phases				
Return value	0 = False 1 = Phase 1 2 = Phase 2 3 = Phase 3 4 = Phase 4			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

refPtPhase (C, SEMA)			
		$$numMachAxes	
Validity:	From NCK version 3.6	Read access	

refPtStatus (C, SEMA)				
Axis must be referenced and is referenced				
Identifier indicating whether an axis requires referencing and is referenced. Note regarding exchange axes: An exchange axis need only ever be referenced in the channel to which it is currently assigned. A referenced exchange axis is thus logged onto the channel in which it is traversing with value "3" (requires referencing and referenced) and in other channels with value "1" (does not require referencing, but referenced).				
Return value	Set bits have the following meanings: Bit0: current measuring system has been referenced Bit1: current measuring system requires referencing (A busy signal affects the state)			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	Achsindex		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

resolvStatus1 (C, SEMA)				
Encoder status for measuring system 1				
Return value	0 = Undefined 1 = Referenced 2 = Activated 3 = Limit frequency exceeded			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

resolvStatus2 (C, SEMA)				
Encoder status for measuring system 2				
Return value	0 = Undefined 1 = Referenced 2 = Activated 3 = Limit frequency exceeded			

resolvStatus2 (C, SEMA)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

safeAcceptCheckPhase (C, SEMA)				
Flag for NCK-side acceptance test phase Flag for NCK-side acceptance test phase, the human-machine interface can determine which acceptance test phase is present on the NCK.				
Return value	0: NCK has acceptance test phase inactive = 0 0ACH: NCK has acceptance test phase active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0ACH
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeAcceptTestMode (C, SEMA)				
SI PowerOn alarms can be acknowledged by Reset SI PowerOn alarms can be acknowledged by Reset in acceptance test mode				
Return value	0: Acceptance test mode: SI PowerOn alarms cannot be acknowledged by Reset 0ACH: Acceptance test mode: SI PowerOn alarms can be acknowledged by Reset			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0FFH
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

safeAcceptTestPhase (C, SEMA)				
Flag for acceptance test phase				
Return value	0: Acceptance test Wizard not selected, activate NCK-side alarm suppression 0ACH: Dialogs for acceptance test support selected, deactivate NCK-side alarm suppression			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0FFH

safeAcceptTestPhase (C, SEMA)		
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access and write access	

safeAcceptTestSE (C, SEMA)				
Flag for NCK-side SE acceptance test Flag for NCK-side SE acceptance test. The human-machine interface starts checking the safe limit positions during the acceptance test				
Return value	0: NCK has SE acceptance test inactive = 0. The single channel SW limit positions are activated. 0ACH: NCK is to activate SE acceptance test. The single channel SW limit positions are de-activated in this way.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0ACH
Addressing				
Line index:	Max. line index:			
	$$numMachAxes			
	Read access			

safeAcceptTestState (C, SEMA)				
Flag for acceptance test status Flag for acceptance test status, the human-machine interface can determine which acceptance test mode is present on the NCK.				
Return value	0: NCK has acceptance test mode inactive 0CH: Acceptance test mode not activated because SI PowerOn alarms already present. The causes of the SI PowerOn alarms must be eliminated first. 0DH: Acceptance test mode not activated, the HMI writes invalid values in $$/C/SEMA/safe-AcceptTestMode to the NCK. 0ACH: NCK has acceptance test mode active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0FFH
Addressing				
Line index:	Max. line index:			
	$$numMachAxes			
	Read access			

safeActPosDiff (C, SEMA)		
Curr. safety actual value difference NCK - drive Current actual value difference betw. NCK and drive monitoring channels		

safeActPosDiff (C, SEMA)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeActVeloDiff (C, SEMA)				
Curr. safety speed difference NCK - drive Current speed difference between NCK and drive monitoring channels				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeActVeloLimit (C, SEMA)				
Safe limit of actual speed				
Return value	-1 => no actual speed limit active >= 0 => limit of actual speed is active			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE		-1	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeDesVeloLimit (C, SEMA)				
Safe limit of desired speed				
Return value	-1 => no desired speed limit active >= 0 => desired speed limit is active			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE		-1	
Addressing				
Line index:			Max. line index:	

safeDesVeloLimit (C, SEMA)				
			$$numMachAxes	
			Read access	

safeFctEnable (C, SEMA)				
Safe operation active Safe operation active (Safety Integrated / SPL)				
Return value	0 = inactive >0 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

safeInputSig2 (C, SEMA)				
Safe input signals part 2				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	0xffff
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeInputSig (C, SEMA)				
Safe input signals of the axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

safeInputSigDrive2 (C, SEMA)				
Safe input signals of the drive part 2				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	0xffff
Addressing				
Line index:			Max. line index:	

safeInputSigDrive2 (C, SEMA)				
			$$numMachAxes	
			Read access	

safeInputSigDrive (C, SEMA)				
Safe input signals of the drive				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

safeMaxVeloDiff (C, SEMA)				
Max. safety speed difference NCK - drive Maximum speed difference between NCK and drive monitoring channels since last NCK Reset				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeMeasPos (C, SEMA)		$VA_IS[x] x = Axis		
Safe actual position of axis Safe actual position of the axis. The physical unit is defined in the variable measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

safeMeasPosDrive (C, SEMA)				
Safe actual position of the drive Safe actual position of drive. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			

safeMeasPosDrive (C, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

safeOutputSig2 (C, SEMA)				
Safe output signals part 2				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	0xffff
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeOutputSig (C, SEMA)				
Safe output signals of the axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

safeOutputSigCam (C, SEMA)				
NCK SI cam signals Results of the NCK safe cam evaluation				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	3FFFFFFF
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeOutputSigCamDrive (C, SEMA)				
Drive SI cam signals Results of the drive safe cam evaluation				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	3FFFFFFF

safeOutputSigCamDrive (C, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeOutputSigDrive2 (C, SEMA)				
Safe output signals of the drive part 2				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	0xffff
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeOutputSigDrive (C, SEMA)				
Safe output signals of the drive				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

safeActiveCamTrack (C, SEMA)				
Status Safe cam track Status Safe cam track (active/inactive)				
Return value	Bit 0 = 1/0: Safe cam track 1 active/inactive Bit 1 = 1/0: Safe cam track 2 active/inactive Bit 2 = 1/0: Safe cam track 3 active/inactive Bit 3 = 1/0: Safe cam track 4 active/inactive			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0xF
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safePosCtrlActive (C, SEMA)				
Axis monitors absolute position				
Return value	0 = Axis does not monitor absolute position (no SE/SN) 1 = Axis monitors absolute position			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeStopOtherAxis (C, SEMA)				
Stop on another axis				
Return value	0: No stop on another axis 1: Stop on another axis			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

spec (C, SEMA)				
Axis specification				
Return value	0 = path axis 1 = positioning axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

stateContrActive (C, SEMA)				
State controller State controller (not available)				
Return value	1 = TRUE 0 = FALSE			
Units and value ranges				
Physical unit:	Data type:			

stateContrActive (C, SEMA)				
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

subSpec (C, SEMA)				T1
Subspecification				
Return value	0 = normal axis 1 = indexing axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

torqLimit (C, SEMA)				
Torque limitation value Torque limitation value (referring to the nominal value of the drive). For linear motors: force limitation value.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

traceState1 (C, SEMA)				
State of trace channel 1				
Return value	0 = idle state 1 = recording started 2 = trigger reached 3 = recording ended 4 = recording aborted			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

traceState2 (C, SEMA)				
State of trace channel 2				
Return value	0 = idle state 1 = recording started 2 = trigger reached 3 = recording ended 4 = recording aborted			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

traceState3 (C, SEMA)				
State of trace channel 3				
Return value	0 = idle state 1 = recording started 2 = trigger reached 3 = recording ended 4 = recording aborted			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

traceState4 (C, SEMA)				
State of trace channel 4				
Return value	0 = idle state 1 = recording started 2 = trigger reached 3 = recording ended 4 = recording aborted			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

traceState4 (C, SEMA)				
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

trackErrContr (C, SEMA)		$VA_TRACK_ERR_CONTR[Achse]		
Position controller difference Position controller difference (actual value / desired value of position)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

trackErrDiff (C, SEMA)				
Contour deviation Contour deviation (difference actual value of position and calculated dynamical model)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

type (C, SEMA)				
Axis type				
Return value	0 = linear axis 1 = rotary axis 2 = spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

vaAbsoluteEnc1ErrCnt (C, SEMA)	$VA_ABSO-LUTE_ENC_ERR_CNT[1,Achse]	
1st meas. system: Error counter for abs. encoder Enc 1: Error counter for absolute encoder		

vaAbsoluteEnc1ErrCnt (C, SEMA)		$VA_ABSO-LUTE_ENC_ERR_CNT[1,Achse]		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaAbsoluteEnc1State (C, SEMA)		$VA_ABSOLUTE_ENC_STATE[1,Achse]		
1st meas. system: Status of abs. encoder interface Enc1: Status of absolute encoder interface				
Return value	Bit0: Interface is active Bit1: Error during parity check Bit2: Error bit alarm Bit3: Error bit CRC error Bit4: Start bit missing with EnDat transfer			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaAbsoluteEnc2ErrCnt (C, SEMA)		$VA_ABSO-LUTE_ENC_ERR_CNT[2,Achse]		
2nd meas. system: Error counter for abs. encoder Enc 2: Error counter for absolute encoder				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaAbsoluteEnc2State (C, SEMA)		$VA_ABSOLUTE_ENC_STATE[2,Achse]		
2nd meas. system: Status of abs. encoder interface Enc2: Status of absolute encoder interface				
Return value	Bit0: Interface is active Bit1: Error during parity check Bit2: Error bit alarm Bit3: Error bit CRC error Bit4: Start bit missing with EnDat transfer			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCcCompValTotal (C, SEMA)		$VA_CC_COMP_VAL_TOTAL[Achse]		
Axial OA total compensation value Axial OA total compensation value via compile cycles				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCecCompVal (C, SEMA)		$VA_CEC_COMP_VAL[Achse]		
Axial sag compensation value				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCurr (C, SEMA)		$VA_CURR[Achse]		
Drive actual current value				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

vaCurr (C, SEMA)	$VA_CURR[Achse]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

vaDistTorque (C, SEMA)		$VA_DIST_TORQUE[Achse]		
Disturbing torque/max. torque (motor end) Disturbing torque/max. torque (motor end, York)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
%	TYPE_DOUBLE	0	-100	100
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaDpe (C, SEMA)		$VA_DPE[x1]		
Status of power enable Status of power enable of a machine axis				
Return value	0 - 1			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEnc1CompVal (C, SEMA)		$VA_ENC1_COMP_VAL[Achse]		
Leadscrew error compensation (LEC) value encoder 1				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEnc1ZeroMonErrCnt (C, SEMA)	$VA_ENC_ZERO_MON_ERR_CNT[1,Achse]	
1st meas. system: Error counter for zero mark mon. Enc 1: Error counter for zero mark monitoring		

vaEnc1ZeroMonErrCnt (C, SEMA)		$VA_ENC_ZERO_MON_ERR_CNT[1,Achse]		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEnc2CompVal (C, SEMA)		$VA_ENC2_COMP_VAL[Achse]		
Leadscrew error compensation (LEC) value encoder 2				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEnc2ZeroMonErrCnt (C, SEMA)		$VA_ENC_ZERO_MON_ERR_CNT[2,Achse]		
2nd meas. system: Error counter for zero mark mon. Enc 2: Error counter for zero mark monitoring				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaFoc (C, SEMA)		$VA_FOC[Achse]		
Actual status of "ForceControl"				
Return value	0: ForceControl not active 1: Modal ForceControl active 2: Non-modal ForceControl active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	

vaFoc (C, SEMA)	$VA_FOC[Achse]	
	$$numMachAxes	
	Read access	

vaFxs (C, SEMA)		$VA_FXS[Achse]		
Actual status of "Travel to fixed stop"				
Return value	0: Axis not at fixed stop 1: Successful travel to fixed stop 2: Unsuccessful travel to fixed stop 3: Travel to fixed stop selection active 4: Fixed stop has been detected 5: Travel to fixed stop deselection active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	5
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

valm1 (C, SEMA)		$VA_IM1[x]		
Alternative name:	aaIm1			
Actual value encoder 1 Actual value in the machine coordinate system (measured encoder 1)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

valm2 (C, SEMA)		$VA_IM2[x]		
Alternative name:	aaIm2			
Actual value encoder 2 Actual value in the machine coordinate system (measured encoder 2)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	

valm2 (C, SEMA)	$VA_IM2[x]	
	$$numMachAxes	
	Read access	

valm (C, SEMA)		$VA_IM[x]		
Alternative name:	aaIm			
Current MCS actual value of an axis The variable determines the encoder actual value (measured by active measuring system) in the machine coordinate system (MCS). All actual value compensations are corrected (leadscrew error compensation, backlash compensation, quadrant error compensation). For rotary axes/spindles, the returned value is independent of the modulo setting, a transformation is only performed for specific actions. When a spindle or axis disable is active, this variable returns the current setpoint. If, in this case, the actual value also has to be returned, BIT3 must be set in MD30455 $MA_MISC_FUNCTION_MASK.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaLagError (C, SEMA)		$VA_LAG_ERROR[Achse]		
Axis following error				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaLoad (C, SEMA)		$VA_LOAD[Achse]		
Drive utilization in %				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-100	100
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaPosctrlMode (C, SEMA)		$VA_POSCTRL_MODE[Achse]		
Position controller mode				
Return value	0: Position control 1: Speed control 2: Holding 3: Parking 4: Tracking			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	4
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaPower (C, SEMA)		$VA_POWER[Achse]		
Active drive power				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaPressureA (C, SEMA)		$VA_PRESSURE_A[Achse]		
Pressure on A end of the cylinder Pressure on A end of the cylinder in bar (only for Hydraulic)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaPressureB (C, SEMA)		$VA_PRESSURE_B[Achse]		
Pressure on B end of the cylinder Pressure on B end of the cylinder in bar (only for Hydraulic)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

vaPressureB (C, SEMA)	$VA_PRESSURE_B[Achse]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

vaSce (C, SEMA)		$VA_SCE[Achse]		
Status of speed controller enable				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaStopSi (C, SEMA)		$VA_STOPSI[Achse]		
Stop from Safety Integrated				
Return value	-1: No stop 0: Stop A 1: Stop B 2: Stop C 3: Stop D 4: Stop E 5: Stop F 10: Test stop of NC			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaSyncDiff (C, SEMA)				
Actual value synchronism difference Actual value synchronism difference for all types of coupling				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	

vaSyncDiff (C, SEMA)				
			$$numMachAxes	
			Read access	

vaSyncDiffStat (C, SEMA)		$VA_SYNCDIFF_STAT[Achse]		
Status of the actual value synchronism difference				
Return value	-4: Reserved -3: No valid value in $VA_SYNCDIFF, tangential control -2: No valid value in $VA_SYNCDIFF, master value coupling and simulated master value -1: No valid value in $VA_SYNCDIFF 0: No valid value in $VA_SYNCDIFF, coupling not active 1: Valid value in $VA_SYNCDIFF			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-4	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaTempCompVal (C, SEMA)		$VA_TEMP_COMP_VAL[Achse]		
Axial temperature compensation value				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaTorque (C, SEMA)		$VA_TORQUE[Achse]		
Drive torque setpoint				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaTorqueAtLimit (C, SEMA)		$VA_TORQUE_AT_LIMIT[Achse]		
Status information torque limitation Status "effective torque equals specified torque limit"				
Return value	0: Effective torque lower than torque limit 1: Effective torque has reached torque limit			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaVactm (C, SEMA)		$VA_VACTM[x] x = Axis		
Axis velocity actual value load side in the MCS Axis velocity actual value on the load side in the MCS				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaValveLift (C, SEMA)		$VA_VALVELIFT[Achse]		
Actual valve lift Actual valve lift in mm (only for Hydraulic)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaXfaultSi (C, SEMA)		$VA_XFAULTSI[Achse]		
Stop F through cross-checking error active				
Return value	Bit 0 set: An actual value error has been discovered in the cross-check between NCK and drive Bit 1 set: Some error has been discovered in the cross-check between NCK and drive and the waiting time until Stop B ($MA_SAFE_STOP_SWITCH_TIME_F) is running or has expired			

vaXfaultSi (C, SEMA)		$VA_XFAULTSI[Achse]		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

activeSvOverride (C, SEMA)				
SG override factor Currently active SG override factor in the NCK				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	-1	-1	100
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaAcc (C, SEMA)		$AA_ACC[Achse]		
Current axial acceleration value Current axial acceleration value				
Units and value ranges				
Physical unit:	Data type:	Initial value		
m/s2, 1000 inch/ s2, rev/ s2, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaSyncDiff (C, SEMA)		$AA_SYNCDIFF[Achse]		
Setpoint synchronism difference				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaJogPosSelected (C, SEMA)		$AA_JOG_POS_SELECTED[Achse]		
JOG to position is active				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaJogPosAct (C, SEMA)		$AA_JOG_POS_ACT[Achse]		
Position reached for JOG to position				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEnc1ZeroMonAccessCnt (C, SEMA)		$VA_ENC_ZERO_MON_AC-CESS_CNT[1,Achse]		
Enc1:Update counter zero monitoring accesses Enc1: Update counter				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEnc2ZeroMonAccessCnt (C, SEMA)		$VA_ENC_ZERO_MON_AC-CESS_CNT[2,Achse]		
Enc2:Update counter zero monitoring accesses Enc2: Update counter				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaAbsoluteEnc1DeltaInit (C, SEMA)		$VA_ABSOLUTE_ENC_DELTA_INIT[1,Achse]		
Enc 1: Initial difference with absolute encoder Enc1: Initial difference				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaAbsoluteEnc2DeltaInit (C, SEMA)		$VA_ABSOLUTE_ENC_DELTA_INIT[2,Achse]		
Enc 2: Initial difference with absolute encoder Enc2: Initial difference				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEnc1ZeroMonAct (C, SEMA)		$VA_ENC_ZERO_MON_ACT[1,Achse]		
Enc1: Current internal zero monitoring values Enc1: Zero monitoring values				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEnc2ZeroMonAct (C, SEMA)		$VA_ENC_ZERO_MON_ACT[2,Achse]		
Enc2: Current internal zero monitoring values Enc2: Zero monitoring values				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	

vaEnc2ZeroMonAct (C, SEMA)	$VA_ENC_ZERO_MON_ACT[2,Achse]	
	$$numMachAxes	
	Read access	

vaAbsoluteEnc1ZeroMonMax (C, SEMA)		$VA_ABSO-LUTE_ENC_ZERO_MON_MAX[1,Achse]		
Enc1:Maximum of $$vaEnc1ZeroMonAct w. abs. encoder Enc1:Maximum of $$vaEnc1ZeroMonAct with absolute encoder				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaAbsoluteEnc2ZeroMonMax (C, SEMA)		$VA_ABSO-LUTE_ENC_ZERO_MON_MAX[2,Achse]		
Enc2:Maximum of $$vaEnc2ZeroMonAct w. abs. encoder Enc2:Maximum of $$vaEnc2ZeroMonAct with absolute encoder				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCoupCorr (C, SEMA)		$AA_COUP_CORR[Achse]		
Gen. coupling: comp. value of synchronism diff. This variable is used to execute the function "Correct synchronism error". It returns the compensation value for the position offset for the generic coupling with CPFRS = "MCS". The actual values of this spindle are compared with the setpoints for the duration (MD 30455 MISC_FUNCTION_MASK, bit 7) of the activation of the VDI interface signal DB31...,DBX31.6 'Correct synchronism' for the following spindle with coupling active. The difference is the compensation value, which can be read with this variable.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCoupCorrDist (C, SEMA)		$AA_COUP_CORR_DIST[Achse]		
Generic coupling: path for $$aaCoupCorr Generic coupling: path still to be retracted for $$aaCoupCorr				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEnc1ZeroMonInit (C, SEMA)		$VA_ENC_ZERO_MON_INIT[1,Achse]		
Enc1:Hardware counter value of basic zero mark Enc1:Hardware counter value of the basic zero mark				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaEnc2ZeroMonInit (C, SEMA)		$VA_ENC_ZERO_MON_INIT[2,Achse]		
Enc2:Hardware counter value of the basic zero mark				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

safeDesChecksum1 (C, SEMA)				
Setpoint hardware checksum Setpoint checksum of the current values of the safety-relevant machine data dependent on hardware parameterization				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access and write access	

aaMaslDef (C, SEMA)		$AA_MASL_DEF		
Current status of master-slave configuration Each slave axis currently coupled via master-slave delivers the machine axis number of the corresponding master axis. Zero is displayed as default if the coupling is not configured. A master axis also shows default value zero.				
Return value	0: No coupling for this axis configured, or axis is master axis, or no coupling active >0: Machine axis number of the master axis with which the slave axis is currently coupled			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$numMachAxes
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

clampStatus (C, SEMA)				
Axis is connected (VDI input signal)				
Return value	Bit 0 = 1: Axis is connected			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaMachax (C, SEMA)		$AA_MACHAX		
Assignment of physical axis The NCU and machine axis are output for an axis, representing the physical image of the axis. The machine axis must be assigned to at least one channel on this NCU, otherwise 0 will be returned. Without an NCU link, i.e. if there is only one NCU, only the number of the machine axis will be output. In this case, the NCU ID is equal to zero.				
Return value	The NCU ID is output as from position 100, e.g. 20005: NCU 2 axis 5.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaIpoNcChanax (C, SEMA)		$AA_IPO_NC_CHANAX		
NC, channel and channel axis no. of interpolator. If the axis is currently interpolated to this NCU, the channel and channel axis number which define the interpolator of the axis are output. If the axis is currently interpolated to a different NCU, the NCU identifier of the interpolated NCU and the global axis number of the machine axis are output. This global axis number can then be used to transfer the interpolated channel and the channel axis number to the other NCU, with NCU ID 2, with anIpoChanAx[203]. The axis must be assigned to at least one channel on this NCU, otherwise 0 will be returned.				
Return value	The channel is output as from position 100, and the channel axis number is output as from position 1, e.g. 1005 - channel 10 channel axis 5. These values are always lower than 10000. The NCU is output as from position 10000, e.g. 20203: NCU 2 and the global axis number is 203.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaIpoNcChanax (C, SEMA)		$VA_IPO_NC_CHANAX		
Machine axis to interpolator. If the machine axis is currently interpolated to this NCU, the channel and channel number which define the interpolator of the axis are output. If the machine axis is currently interpolated to a different NCU, the NCU identifier of the interpolated NCU and the global axis number of the machine axis are output. This global axis number can then be used to transfer the interpolated channel and the channel axis number to the other NCU, with NCU ID 2, with anIpoChanAx[203]. The axis must be assigned to at least one channel on this NCU, otherwise 0 will be returned.				
Return value	The channel is output as from position 100, and the channel axis number is output as from position 1, e.g. 1005 - channel 10 channel axis 5. These values are always lower than 10000. The NCU is output as from position 10000, e.g. 20203: NCU 2 and the global axis number is 203.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaMotSensorDigi (C, SEMA)		$VA_MOT_SENSOR_DIGI		
Digital sensors on the motor This variable determines the states of the digital sensors S4 and S5. The variable is bit-coded, and has the following meanings: Bit0 = 0: Bit1 = 0: Bit2 = 0: Bit3 = 0: Bit4 = 1: Sensor S4 piston end position Bit5 = 1: Sensor S5 angular position of the shaft				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaMotSensorAna (C, SEMA)		$VA_MOT_SENSOR_ANA		
Analog sensor on the motor This variable determines the analog measured value of sensor S1. At a resolution of 1 mV, the analog value 0 - 10 V is mapped by a maximum of +10000 increments.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaMotSensorConf (C, SEMA)	$VA_MOT_SENSOR_CONF	
Configuration of motor sensors The configuration of the motor sensors can be queried with this variable. The variable is bit-coded, and has the following meanings: Bit0 = 1: Sensor system present. Bit1 = 1: Sensor S1 present. Analog measured value for position of the draw-bar. Bit2 = 0: Bit3 = 0: Bit4 = 1: Sensor S4 present. Digital value for the piston end position Bit5 = 1: Sensor S5 present. Digital value for the angular position of the shaft.		

vaMotSensorConf (C, SEMA)		$VA_MOT_SENSOR_CONF		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaMotClampingState (C, SEMA)		$VA_MOT_CLAMPING_STATE		
State of the clamping system Starting from the position of the draw-bar (value of the S1), this variable determines the clamping state. A maximum speed is assigned to each state. These are stored in the drive parameters p5043[0..6]. The following values are possible: 0: Sensor not present 1: Initial state, speed limit 0 rpm 2: Alarm, speed limit 0 rpm 3: Tool released / ejected, speed limit see drive parameter p5043[0] 4: Clamping (by spring force), speed limit see drive parameter p5043[1] 5: Releasing (by compressed air), speed limit see drive parameter p5043[2] 6: Releasing (by compressed air), speed limit see drive parameter p5043[3] 7: Clamped with tool, speed limit see drive parameter p5043[4] 8: Clamped with tool, speed limit see drive parameter p5043[4] 9: Further clamping (by spring force), speed limit see drive parameter p5043[5] 10: Clamped without tool, speed limit see drive parameter p5043[6] 11: Alarm, speed limit 0 rpm				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCpSync2 (C, SEMA)		$VA_CPSYNC2[a]		
Status synchronism(2) of the foll. axis / spindle Second synchronism monitoring of the following axis / spindle				
Return value	0: Monitoring not active Bit 0 = 1: Monitoring 'Synchronism(2) coarse' active Bit 1 = 1: Synchronism(2) coarse available Bit 2 = 1: Monitoring 'Synchronism(2) fine' active Bit 3 = 1: Synchronism(2) fine available			
Units and value ranges				
Physical unit:	Data type:			

vaCpSync2 (C, SEMA)		$VA_CPSYNC2[a]		
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaPosRes (C, SEMA)		$AA_POSRES		
Axis status "Position restored" Axis status "Position restored". The value TRUE shows that the position of the axis has been restored after the voltage breakdown. ($MA_ENC_REFP_STATE[] = 3). After referencing of the axis, the value goes to FALSE.				
Return value	1 = TRUE: Axis position not restored 0 = FALSE: Axis position restored			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaCollPos (C, SEMA)		$AA_COLLPOS[Achse]		
Collision position Position of a machine axis with risk of collision				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

driveDdsPerMds (C, SEMA)				
Number of drive data sets Number of drive data sets assigned to a motor data set. Refer to the SINAMICS S120 Function Manual for more information about drive and motor data sets.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

driveDdsPerMds (C, SEMA)			
		$$numMachAxes	
		Read access	

spindleModePiState (C, SEMA)				
Status of the spindle mode switchover Status of the spindle mode switchover for this machine axis by PI service _N_SPIMOD				
Return value	0 = PI service not selected 10 = PI service active 50 = PI service ended successfully 101 = PI service rejected because axis/spindle is not known in the channel 102 = PI service rejected because axis/spindle is not available in the channel 104 = PI service rejected because axis/spindle is not defined as a spindle. 105 = PI service rejected because axis/spindle is a permanently assigned PLC axis/spindle 106 = PI service rejected because axis/spindle is an active following axis/spindle 107 = PI service rejected because axis/spindle is a transformed spindle/axis 108 = PI service rejected because axis/spindle is not available as a command axis 200 = PI service rejected because of an internal error				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	999
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaLoadSmooth (C, SEMA)		$AA_LOAD_SMOOTH[Achse]		
Smoothed drive load Smoothed drive load in %				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaPowerSmooth (C, SEMA)		$AA_POWER_SMOOTH[Achse]		
Smoothed drive power Smoothed drive power in W (only available for PROFIdrive drives)				
Units and value ranges				
Physical unit:	Data type:			
W	TYPE_DOUBLE			

aaPowerSmooth (C, SEMA)	$AA_POWER_SMOOTH[Achse]	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes	
	Read access	

safeAxisType (C, SEMA)				
Safety axis mode Type of axial safety monitoring 0 = No SINUMERIK Safety Integrated active 1 = SINUMERIK Safety Integrated (SPL) active 2 = SINUMERIK Safety Integrated plus (F-PLC) active 2 = SINUMERIK Safety Integrated (F-PLC) active				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:		Max. line index:		
		$$numMachAxes		
		Read access		

enc1IsOn (C, SEMA)				
Position measuring system 1 is enabled Operating status of position measuring system 1				
Return value	0 = Position measuring system 1 parked (or is not configured), may be removed 1 = Position measuring system 1 is passive 2 = Position measuring system 1 is active (e.g. position control)			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	2
Addressing				
Line index:		Max. line index:		
		$$numMachAxes		
		Read access		

enc2IsOn (C, SEMA)				
Position measuring system 2 is enabled Operating status of position measuring system 2				
Return value	0 = Position measuring system 2 parked (or is not configured), may be removed 1 = Position measuring system 2 is passive 2 = Position measuring system 2 is active (e.g. position control)			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	2

enc2IsOn (C, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaAxForce (C, SEMA)		$VA_AX_FORCE[x] x = Axis		
Feedrate force				
Units and value ranges				
Physical unit:	Data type:	Initial value		
N	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaDesValFiltersDeltaPos (C, SEMA)		$VA_DESVAL_FILTERS_DEL-TA_POS[Achse]		
Position difference between setpoint filter chains Position difference between the setpoint filter chains				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user de-fined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

aaDesValFiltersSelect (C, SEMA)		$VA_DESVAL_FILTERS_ACTIVE[Achse]		
Selection of the setpoint filter chains				
Return value	0 = FALSE = Selection of first setpoint filter chain 1 = TRUE = Selection of second setpoint filter chain			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_BOOLEAN	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaDesValFiltersActive (C, SEMA)		$VA_DESVAL_FILTERS_ACTIVE[Achse]		
Status of the setpoint filter chains				
Return value	0 = FALSE = The first setpoint filter chain is active 1 = TRUE = The second setpoint filter chain is active			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_BOOLEAN	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaDesValFiltersDelay1 (C, SEMA)		$VA_DESVAL_FILTERS_DE-LAY_1[Achse]		
eff. delay time of the 1st setpoint filter chain effective delay time of the 1st setpoint filter chain				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaDesValFiltersDelay2 (C, SEMA)		$VA_DESVAL_FILTERS_DE-LAY_2[Achse]		
eff. delay time of the 2st setpoint filter chain effective delay time of the 2st setpoint filter chain				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptAccDynNorm (C, SEMA)				
Adaptation factor for acceleration DYNNORM Active adaptation factor for acceleration value DYNNORM				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	

vaCAdaptAccDynNorm (C, SEMA)				
			$$numMachAxes	
			Read access	

vaCAdaptAccDynPos (C, SEMA)				
Adaptation factor for acceleration DYNPOS Active adaptation factor for acceleration value DYNPOS				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptAccDynRough (C, SEMA)				
Adaptation factor for acceleration DYNROUGH Active adaptation factor for acceleration value DYNROUGH				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptAccDynSemiFin (C, SEMA)				
Adaptation factor for acceleration DYNSEMIFIN Active adaptation factor for acceleration value DYNSEMIFIN				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptAccDynFinish (C, SEMA)				
Adaptation factor for acceleration DYNFINISH Active adaptation factor for acceleration value DYNFINISH				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		

vaCAdaptAccDynFinish (C, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptPosCtrlGain (C, SEMA)				
Adaptation factor for position control gain Active adaptation factor for position control gain factor				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptFfwTorqueFact (C, SEMA)				
Adaptation factor for torque feedforward control Active adaptation factor for inertia and/or mass of the torque feedforward control				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptDrvAdapt1 (C, SEMA)				
Adaptation factor ADAPT1 for drive parameters Active adaptation factor ADAPT1 for drive parameters				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptDrvAdapt2 (C, SEMA)				
Adaptation factor ADAPT2 for drive parameters Active adaptation factor ADAPT2 for drive parameters				
Units and value ranges				
Physical unit:	Data type:	Initial value		

vaCAdaptDrvAdapt2 (C, SEMA)				
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptDrvAdapt3 (C, SEMA)				
Adaptation factor ADAPT3 for drive parameters Active adaptation factor ADAPT3 for drive parameters				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptDrvAdapt4 (C, SEMA)				
Adaptation factor ADAPT4 for drive parameters Active adaptation factor ADAPT4 for drive parameters				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptInputLoad (C, SEMA)				
Input value for load adaptation Input value (inertia and/or mass) for load adaptation				
Units and value ranges				
Physical unit:	Data type:	Initial value		
kgm2	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptIsActive (C, SEMA)				
Adaptation is active At least one adaptation factor is active				
Return value	0 = adaptation not active 1 = adaptation active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptJerkDynNorm (C, SEMA)				
Adaptation factor for jerk DYNNORM Active adaptation factor for jerk value DYNNORM				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptJerkDynPos (C, SEMA)				
Adaptation factor for jerk DYNPOS Active adaptation factor for jerk value DYNPOS				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptJerkDynRough (C, SEMA)				
Adaptation factor for jerk DYNROUGH Active adaptation factor for jerk value DYNROUGH				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	

vaCAdaptJerkDynRough (C, SEMA)				
			$$numMachAxes	
			Read access	

vaCAdaptJerkDynSemiFin (C, SEMA)				
Adaptation factor for jerk DYNSEMIFIN Active adaptation factor for jerk value DYNSEMIFIN				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptJerkDynFinish (C, SEMA)				
Adaptation factor for jerk DYNFINISH Active adaptation factor for jerk value DYNFINISH				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptAccDynPrec (C, SEMA)				
Adaptation factor for acceleration DYNPREC Active adaptation factor for acceleration value DYNPREC				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

vaCAdaptJerkDynPrec (C, SEMA)				
Adaptation factor for jerk DYNREC Active adaptation factor for jerk value DYNPREC				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		

vaCAdaptJerkDynPrec (C, SEMA)		
Addressing		
Line index:		Max. line index:
		$$numMachAxes
		Read access

3.2.3.22 SGA: State data in the WCS

All status data that are dependent on machine movement and specified in the workpiece coordinate system are included in module SGA. Supplementary information can be found in module SEGA. The individual variables are defined as arrays where the line index is the number of the axis (assigned to the current channel).The variable "name" in module SGA with the line index in question identifies the axis.

The assignment of the line indices in modules SGA and SEGA is identical.

With SW 5.2 and later, OPI modules SGA and SEGA can be addressed via the geo-axis no. instead of via the channel axis no.:

Line index 1001: 1st geo-axis

Line index 1002: 2nd geo-axis

Line index 1003: 3rd geo-axis

The number of channel axes (geometry, special axes and spindles) can be found in "$$numMachAxes" in module Y in area C.

actIncrVal (C, SGA)		DB31-61, DBB5		
Active INC weighting of the axis				
Return value	0 = INC_10000 1 = INC_1000 2 = INC_100 3 = INC_10 4 = INC_1 5 = INC_VAR 6 = INC_JOG_CONT 7 = no increment mode has been set			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

actProgPos (C, SGA)		
Current position of axis Returns the current actual value for the axis.		

actProgPos (C, SGA)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

actToolBasePos (C, SGA)				
Current setpoint of the axis position in the MCS Determines the current setpoint of axis "x" in the machine coordinate system (MCS) for the corresponding axis. This value contains all axis-specific override components (e.g. Differential Resolver Function, $AA_OFF, external work offset)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

actToolEdgeCenterPos (C, SGA)	$AA_IW[x] x = Axis			
Center point of a cutting edge Center point of a cutting edge. Physical unit is defined in the variable extUnit (from this module)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdProgPos (C, SGA)				
Target value of programmed position Returns the target value programmed in the part program for the axis.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

cmdProgPos (C, SGA)				
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdToolBasePos (C, SGA)				
Pos. setpoint in machine coordinate system (MCS) The axial variable determines the current setpoint in the machine coordinate system (MCS) for the corresponding machine axis. The setpoint is equivalent to the interpolator output value for the current interpolation cycle. The MCS value contains no axial override components (DRF, AA_OFF, ext. work offset, etc.).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdToolEdgeCenterPos (C, SGA)				
Position of the cutting edge center point. Position of the cutting edge center point. Physical unit is defined in variable extUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

extUnit (C, SGA)				
Current physical unit Current physical unit of the related geometry axis or auxiliary axis				
Return value	0 = mm 1 = inch 2 = degree 3 = indexing position 4 = userdef			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

name (C, SGA)				
Axis name				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

progDistToGo (C, SGA)				
Difference between current and programmed position Returns the difference between the current actual value and the value (distance-to-go) programmed in the part program for the axis.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

progREPOS (C, SGA)				
Programmed position, REPOS Programmed position, REPOS. The physical unit is defined in the variable extUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

status (C, SGA)				
Axis status				
Return value	0 = travel command in plus direction 1 = travel command in minus direction 2 = exact position coarse reached 3 = exact position fine reached			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

status (C, SGA)				
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

subType (C, SGA)				
Geometry or auxiliary axis Axis type geometry or auxiliary axis				
Return value	0 = auxiliary axis 1 = geometry axis 2 = orientation axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

toolBaseDistToGo (C, SGA)				
Tool base distance-to-go Tool base distance-to-go. Physical unit is defined in the variable extUnit (in this module)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

toolBaseREPOS (C, SGA)				
Tool base REPOS Tool base REPOS. Physical unit is defined in the variable extUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

toolEdgeCenterDistToGo (C, SGA)				
Center point of cutting edge distance-to-go Center point of cutting edge distance-to-go. Physical unit results from the variable extUnit (in this module)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

toolEdgeCenterREPOS (C, SGA)				
Center point of the cutting edge REPOS Center point of the cutting edge REPOS. Physical unit is defined in the variable extUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

varIncrVal (C, SGA)				
Settable value for INC_VAR Setable value for INC_VAR. The physical unit depends on whether the axis is rotary or linear. Linear axes: 1mm rotary axes: 1/1000 degrees				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access and write access	

3.2.3.23 SINF: Part program-specific state data

During automatic execution of a part program different parameters can influence the type of machining. The current status data for the selected part program are combined in module SINF. The status data must only be changed via the PLC. interface.

DRFActive (C, SINF)				
DRF active				
Return value	0 = not active 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

cfgStopActive (C, SINF)		DB21-30, DBX7.6		
Configured stop Configured stop (CFG-STOP)				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

feedStopActive (C, SINF)				
Feed disable				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

ipoBlocksOnly (C, SINF)				
Display traversing blocks				
Return value	0 = normal block transfer 1 = exclusively traversing blocks			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

optAssStopActive (C, SINF)				
Associated M01 selected				
Return value	0: Not selected 1: Selected			
Units and value ranges				
Physical unit:	Data type:	Initial value		

optAssStopActive (C, SINF)				
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

optStopActive (C, SINF)				
M01 selected				
Return value	0 = not selected 1 = selected			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

progTestActive (C, SINF)		DB21-30, DBX1.7		K1
Program test				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

rapFeedRateOvrActive (C, SINF)				
ROV rapid traverse override				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

singleBlockActive (C, SINF)				
Single block The variable returns the current status for the current mode of the Single block function.				
Return value	Value range: o =0: No single block active o =1: Single block mode 1 (SBL 1, Main run, single block) o =2: Single block mode 2 (SBL 2, Decoding run, single block) o =3: Single block mode 3 (SBL 3, Stop in the cycle)			
Units and value ranges				
Physical unit:	Data type:			

singleBlockActive (C, SINF)				
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

singleBlockType (C, SINF)				
Single block mode Single block mode can be set with this variable. o =1: Single block mode 1 (main run, single block) o =2: Single block mode 2 (decoder, single block) o =3: Single block mode 3 (stop in the cycle) Single block mode is activated by the interface signal DB21, ... DBX0.4 = 1				
Return value	1 = interpolation single block 2 = decoder single block			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access and write access	

skipLevel0Active (C, SINF)				
Skip level /0 Info whether skip level /0 is activated.				
Return value	0: Skip level /0 not active 1: Skip level /0 active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Validity:	From NCK version 3.6		Read access	

skipLevel1Active (C, SINF)				
Skip level /1 Info whether skip level /1 is activated.				
Return value	0: Skip level /1 not active 1: Skip level /1 active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Validity:	From NCK version 3.6		Read access	

skipLevel2Active (C, SINF)				
Skip level /2 Info whether skip level /2 is activated.				
Return value	0: Skip level /2 not active 1: Skip level /2 active			

skipLevel2Active (C, SINF)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Validity:	From NCK version 3.6		Read access	

skipLevel3Active (C, SINF)				
Skip level /3 Info whether skip level /3 is activated.				
Return value	0: Skip level /3 not active 1: Skip level /3 active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Validity:	From NCK version 3.6		Read access	

skipLevel4Active (C, SINF)				
Skip level /4 Info whether skip level /4 is activated.				
Return value	0: Skip level /4 not active 1: Skip level /4 active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Validity:	From NCK version 3.6		Read access	

skipLevel5Active (C, SINF)				
Skip level /5 Info whether skip level /5 is activated.				
Return value	0: Skip level /5 not active 1: Skip level /5 active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Validity:	From NCK version 3.6		Read access	

skipLevel6Active (C, SINF)				
Skip level /6 Info whether skip level /6 is activated.				
Return value	0: Skip level /6 not active 1: Skip level /6 active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

skipLevel6Active (C, SINF)				
-	TYPE_UWORD	0	0	1
Validity:	From NCK version 3.6		Read access	

skipLevel7Active (C, SINF)				
Skip level /7 Info whether skip level /7 is activated.				
Return value	0: Skip level /7 not active 1: Skip level /7 active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Validity:	From NCK version 3.6		Read access	

skipLevel8Active (C, SINF)				
Skip level /8 Info whether skip level /8 is activated.				
Return value	0: Skip level /8 not active 1: Skip level /8 active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
			Read access	

skipLevel9Active (C, SINF)				
Skip level /9 Info whether skip level /9 is activated.				
Return value	0: Skip level /9 not active 1: Skip level /9 active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
			Read access	

trialRunActive (C, SINF)		DB21-30, DBX0.6		V1
Dry run feedrate				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

3.2.3.24 SMA: State data in the MCS

All state data that are dependent on machine movement and are defined specifically for machine axes (geometry and special axes) are combined in module SMA. Supplementary information is to be found in module SEMA. The individual variables are defined as fields where the line index is the number of the machine axis (assigned to the current channel). The variable "name" in module SMA with the line index in question identifies the axis.

The assignment of the line indices in modules SMA and SEMA is identical.

actIncrVal (C, SMA)		DB31-61, DBB5		H1
Active INC weighting of the axis				
Return value	0 = INC_10000 1 = INC_1000 2 = INC_100 3 = INC_10 4 = INC_1 5 = INC_VAR 6 = INC_JOG_CONT 7 = no incremental mode set			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

actToolBasePos (C, SMA)		$AA_IM[x] x = Ax is		
Current setpoint of the axis position in the MCS Determines the current setpoint of axis "x" in the machine coordinate system (MCS) for the corresponding axis. This value contains all axis-specific override components (e.g. Differential Resolver Function, $AA_OFF, external work offset)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdToolBasePos (C, SMA)				
Pos. setpoint in machine coordinate system (MCS). The axial variable determines the current setpoint in the machine coordinate system (MCS) for the corresponding machine axis. The setpoint is equivalent to the interpolator output value for the current interpolation cycle. The MCS value contains no axial override components (DRF, AA_OFF, ext. work offset, etc.).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

extUnit (C, SMA)				
Current physical unit Current physical unit of the axis position				
Return value	0 = mm 1 = inch 2 = degree 3 = indexing position 4 = userdef			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

name (C, SMA)				
Axis name				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

status (C, SMA)				
Axis status				
Return value	0 = travel command in plus direction 1 = travel command in minus direction 2 = exact position coarse reached 3 = exact position fine reached			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

toolBaseDistToGo (C, SMA)				
Tool base distance-to-go Tool base distance-to-go. Physical unit is defined in the variable extUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

toolBaseREPOS (C, SMA)				
Tool base REPOS Tool base REPOS. Physical unit is defined in the variable extUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

varIncrVal (C, SMA)				
Settable value for INC_VAR Settable value for INC_VAR. The physical value depends on whether the axis is linear or rotary. Linear axis: unit is 1 mm Rotary axis: unit is 1/1000 degrees				

varIncrVal (C, SMA)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

3.2.3.25 SNCF: Active G-functions

All G functions are organized in G groups. Only one function of each G group can be active at a time.

The module SNCF consists of a single variable that is organized as an array. The row index corresponds to the G group number.

ncFkt (C, SNCF)				
Active G function Active G-function of relevant group G <No>. If there is no function active within the corresponding G-group, the variable returns an empty string "\0".				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numGCodeGroups	
Validity:	From NCK version 3.6		Read access	

ncFktAct (C, SNCF)				
Active G function (current language) Active G function of relevant current group in current language mode. Depending on whether function has been programmed in Siemens or ISO Dialect mode, this is identical to $$ncFkt or $$ncFktFanuc.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numGCodeGroups bzw. $$numGCode-GroupsFanuc	
			Read access	

ncFktBin (C, SNCF)				
Active G-function of the correponding group				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numGCodeGroups	
			Read access	

ncFktBinAct (C, SNCF)				
Index act. G function (current language) Active G function of relevant current group in current language mode. Depending on whether function has been programmed in Siemens or ISO Dialect mode, this is identical to $$ncFktBin or $$ncFktBinFanuc. (The value is the index of the active G function within the group)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numGCodeGroups bzw. $$numGCode-GroupsFanuc	
			Read access	

ncFktBinFanuc (C, SNCF)				
Index of active G function (ISO Dialect) Active G function of relevant ISO Dialect group (the value is the index of the active G function within the group)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numGCodeGroupsFanuc	
			Read access	

ncFktBinS (C, SNCF)				
Index of the active G function for block search Index of the active G function of the corresponding group for block search with calculation Notice: This variable is available only for logging block search events, but not for the Variable Service.				

ncFktBinS (C, SNCF)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numGCodeGroups	
			Read access	

ncFktFanuc (C, SNCF)				
Active G function (ISO Dialect) Active G function of relevant ISO Dialect group				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numGCodeGroupsFanuc	
			Read access	

ncFktS (C, SNCF)				
Name of the active G function for block search Name of the active G function of the corresponding group for block search with calculation Notice: This variable is available only for logging block search events, but not for the Variable Service.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numGCodeGroups	
			Read access	

3.2.3.26 SPARP: Part program information

This module contains information on the currently active part programm in the respective channel.

absoluteBlockBufferName (C, SPARP)				
Upload buffer for display blocks File name with path of upload buffer in which display blocks are stored				
Return value	Empty string: Function is deactivated			

absoluteBlockBufferName (C, SPARP)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

absoluteBlockBufferPreview (C, SPARP)				
Part of $$absoluteBlockBufferName Part of content of file $$absoluteBlockBufferName. The desired content of the variables is set by $MC_MM_ABSBLOCK_BUFFER_CONF. In principle, only complete parts program blocks are entered. If the desired number of previous blocks are not present, then an empty block ("LF") is entered in that place. If there is insufficient space for all parts program blocks, then the previous blocks are first replaced by empty blocks ("LF"), if this is still insufficient, the blocks at the end are also omitted.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

absoluteBlockCounter (C, SPARP)				
Modification counter for upload buffer Modification counter for display information in the upload buffer				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actBlock (C, SPARP)				
Current block (DISPLOF active) Current part program block. With DISPLOF the subroutine call is displayed.				

actBlock (C, SPARP)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actBlockA (C, SPARP)				
Current block (DISPLOF inactive) Current part program block. If search run is active, then search run block is displayed. Display is always made irrespective of DISPLOF.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actBlockI (C, SPARP)				
Current interpreter block Current part program in the interpreter. Display is always made irrespective of DISPLOF.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actLineNumber (C, SPARP)				
Line number of current NC instruction (start:1) Line number of the current NC instruction (starting at 1) 0: before program start -1: not available due to an error -2: not available because of DISPLOF				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			

actLineNumber (C, SPARP)				
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actPartProgram (C, SPARP)				
Content of the current part program Content of the current part program starting with the previous block. Blocks may be cut off at the end of the string. The line index determines the section within the program. An efficient current block display can be achieved with the aid of a cyclic variable service. If multiple lines are required, then it must be ensured that the client reads line 1 first, so that an NCK internal buffer is filled, which ensures that the following lines are returned accordingly. Note: In EES mode, only the current part program block is ever returned!				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

block (C, SPARP)				
Part program blocks from the current operation To display the currently active part program, NCK supplies up to 3 blocks of the part program in one single variable job (last, current and next block). Line index 1: string of the last block Line index 2: string of the current block Line index 3: string of the next block To gain consistent information, all 3 array elements must be processed in one variable request. This is why the maximum string length of each array element is limited to 66 characters.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			3	
Validity:	From NCK version 3.6		Read access	

blockNoStr (C, SPARP)				
Block number				

blockNoStr (C, SPARP)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

circleCenter (C, SPARP)				
Center of the circle Center of the circle (WCS)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

circleCenterS (C, SPARP)				
Corr. to circleCenter for block search with calc. Corresponds to circleCenter for search with calculation Attention: This variable is available for protocolling the block search events only, not for the Variable Service!				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

circlePlane (C, SPARP)				
Vector of circular plane The vector perpendicular to the circular plane (axial) is output to enable identification of the position of a circle in space				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

circlePlaneData (C, SPARP)				
circlePlaneData To identify the position of a circle in space, the vector which is vertical to the plane of the circle is output (vector) The variable consists of three values of the type DOUBLE, that is, it is 24 bytes long. Note: The variable is released for use only with the Simultaneous recording function.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

circlePlaneDataNorm (C, SPARP)				
circlePlaneDataNorm To identify the position of a circle in space, the vector which is vertical to the plane of the circle is output (normalized vector) The variable consists of three values of the UWORD type, that is, it is 6 bytes long. Note: The variable is released for use only with the Simultaneous recording function.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

circlePlaneDataNormS (C, SPARP)				
circlePlaneDataNormS To identify the position of a circle in space during the search, the vector which is vertical to the plane of the circle is output (normalized vector) The variable consists of three values of the UWORD type, that is, it is 6 bytes long. Note: The variable is released for use only with the Simultaneous recording function.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

circlePlaneS (C, SPARP)				
Vector circular plane block search The vector perpendicular to the circular plane (axial) is output to enable identification of the position of a circle in space				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

circleRadius (C, SPARP)				
Radius of the circle (only effective for G02/G03)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

circleRadiusS (C, SPARP)				
circleRadius for block search. Corresponds to $$circleRadius for block search with calculation. Note: This variable is not available for the variable service, but only for logging in connection with block search events!				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

circleTurn (C, SPARP)				
Programmed circular passes Progr. number of additional circular passes with helical interpolation in curr. program				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

circleTurnS (C, SPARP)				
Programmed circular passes search Programmed number of additional circular passes with helical interpolation in the current program for search with calculation. Note: This variable is not available for the Variable Service, but only for logging of block search events				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

cmdToolEdgeCenterCircleCenterEns (C, SPARP)				
Arc center in relation to WOS frame Arc center in relation to WOS frame, i.e. with tool length but without tool radius				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

cmdToolEdgeCenterCircleCenterEnsS (C, SPARP)				
Corresponds to circleCenterWos Corresponds to circleCenterWos for block search with calculation in relation to the WOS frame, i.e. with tool length but without tool radius Note: This variable is not available for the variable service, but only for logging in connection with block search events!				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

cmdToolEdgeCenterCircleDataEns (C, SPARP)				
cmdToolEdgeCenterCircleDataEns Corresponds to $$cmdToolEdgeCenterCircleCenterEns for the 3 geometry axes as well as $$cmdToolEdgeCenterCircleRadiusEns The variable consists of four values of the DOUBLE type, i.e. is 32 bytes long. Note: The variable is released for use only with the Simultaneous recording function.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

cmdToolEdgeCenterCircleRadiusEns (C, SPARP)				
Arc radius in relation to WOS frame Arc radius in relation to WOS frame as center-point path, i.e. with tool length but without tool radius				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

cmdToolEdgeCenterCircleRadiusEnsS (C, SPARP)				
Corresponds to circleRadiusWos Corresponds to circleRadiusWos for block search with calculation in relation to WOS frame as center-point path. i.e. with tool length but without tool radius Note: This variable is not available for the variable service, but only for logging in connection with block search events!				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

lastBlockNoStr (C, SPARP)				
Last programmed block number Indicates the last programmed block number, if $MN_DISPLAY_FUNCTION_MASK bit 0 is set. A block number is shown until either a new block number is programmed or the subroutine level which generated the block number has been left. Block numbers of masked blocks are not displayed. There is also no display if DISPLOF is active.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

msg (C, SPARP)				PG
Message from the part program Messages from a part program can be programmed with the instruction 'MSG (...)'. The variable 'msg' contains the text of the current 'MSG(...)'-instruction until a new instrucion is processed or until the message is deleted with the instruction 'MSG ()'.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

progName (C, SPARP)				
Program name Progam name of the currently active program (or subroutine)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

seekOffset (C, SPARP)				
Line number of current NC block Line number of the current NC block in the program $$/C/SPARP/workPandProgName				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

seekw (C, SPARP)				
Line enabled for modification First line enabled for modification in part program				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

selectedWorkPProg (C, SPARP)				
Currently selected program Currently selected program, i.e. the program that has been selected with "Select". The variable also displays the program in the JOG and MDI modes. During the simulation, the simulation search temporarily deselects the selected program and selects the program to be simulated. This is hidden by selectedWorkPProg, i.e. during the simulation search, selectedWorkPProg remains unchanged.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

singleBlock (C, SPARP)				
Single block read in quasi-stop state In most cases the variable 'block' is used to read the currently active blocks of the part program. Because this variable is limited to 66 characters per string, it might be necessary (for long blocks) to read longer strings. The variable 'singleBlock' can read complete blocks (up to strings with 198 charecters) . 3 lines can be addressed: Line index 1: last block Line index 2: current block Line index 3: next block It is not guaranteed for rapid block changes, that the information of 3 successive blocks is always consistent, because each block is read with a single variable request. This method is only safe, if the part program has stopped.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			3	
Validity:	From NCK version 3.6		Read access	

stepEditorFormName (C, SPARP)				
Current module name for step editor Current module name for step editor is stored				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

workPName (C, SPARP)				
Workpiece name abbreviated Name of the active workpiece				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

workPNameLong (C, SPARP)				
Workpiece name Name of the active workpiece				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

workPandProgName (C, SPARP)				
Workpiece and program name Workpiece name and name of current program.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

byteOffset (C, SPARP)				
Byte offset of the current NC block Byte offset of the current NC block in the program $$/C/SPARP/workPandProgName				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

extProgFlag (C, SPARP)				
External execution Indicates whether programs are being executed externally				
Return value	0: Program is being processed from NCK program memory 1: Program is being executed externally 2: Program is being executed in EES mode			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

displProgLevel (C, SPARP)				
Program level to be displayed. Lowest program level to be displayed. The value 1 corresponds to the main program level.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

displProgLevelVL (C, SPARP)				
Program level in the VL, which shall be displayed. Lowest preprocessing program level that shall be displayed. The value 1 corresponds to the main program level.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

workPandProgNameVL (C, SPARP)				
Workpiece name and program name VL Workpiece name and program name of the current program in the preprocessing.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

byteOffsetVL (C, SPARP)				
Byte offset of NC block VL Byte offset of the current NC block in the preprocessing				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	

byteOffsetVL (C, SPARP)		
	1	
	Read access	

eesBufferStart (C, SPARP)				
Byte offset of the first block in the EES buffer The value is only relevant with EES, when the part program is partially loaded into a buffer NCK. It indicates wthich NC block is entered first in the buffer.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

eesBufferEnd (C, SPARP)				
Byte offset following the last block in the buffer The value is only relevant with EED, when the part program is partially loaded into a buffer of the NCK. It indicates which NC block has been entered last in the buffer. The comparison with the value $$byteOffsetVL shows whether the reloading process is performed quickly enough in order to perform the preprocessing run without delay.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

eesBufferFilling (C, SPARP)				
Number of bytes for interpreter The value is only relevant with EES, when the part program is partially loaded into a buffer of the NCK. It indicates who many bytes are provided to the interpreter for processing in a the buffer ($$eesBufferEnd - $$byteOffsetVL). If the value approaches 0, this indicates that the reloading process is not performed quickly enough in order to supply the interpreter quickly enough with NC blocks.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

eesBufferStatus (C, SPARP)				
Status of the EES buffer				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

eesProgLevel (C, SPARP)				
Program level in the EES to be displayed. Lowest program level of the EES reloading mode, which shall be displayed. The value 1 corresponds to the main program level.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

3.2.3.27 SPARPF: Search pointer for block search

To look for a particular block within a parts program the user can enter search criteria and start a block search. The variables to be entered are combined in the module SPARPF and must be written by the HMI (or another component on the MPI bus).

One main program level and 11 subroutine levels can be processed. These levels are the row indices of the individual variables. The search targets (seek pointer and search string) can only be used mutually exclusively in one level. If a collision occurs, a negative acknowledgment results when the block search is started.

Depending on the search type, the search string is either a block label, block number or any string.

If no path name is specified, the default search strategy for subroutine calls is used. The main program entered in the first program level must be selected for the block search; otherwise the search request is acknowledged negatively.

The array indices (row indices) mean:

1 = main program level for search run
2-18 = subroutine levels for search run

101 = main program level for stop run
102-118 = subroutine levels for stop run

forward (C, SPARPF)				
Search direction Search direction Search direction "backwards" is only possible in the mode without calculation				
Return value	1 = backwards (without calculation) 2 = forwards			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access and write access	

invocCount (C, SPARPF)				
Actual value of subroutine call counter. Actual value of the subroutine call counter. Is always 1 for the main program.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access and write access	

plcStartReason (C, SPARPF)				
Specification of channel for SERUPRO Specifies for the SERUPRO function which channel has to be started by the PLC so that the current channel starts.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			112	
			Read access and write access	

progName (C, SPARPF)				
Program name Progam name of the currently active program (or subroutine)				
Units and value ranges				
Physical unit:	Data type:			

progName (C, SPARPF)				
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access and write access	

searchString (C, SPARPF)				
Search string Search string (the first 64 characters of the NC block - corresponding to search pointer). Contents of the search string depends on the search type and are either: block label block number any string				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access and write access	

searchType (C, SPARPF)				
Search type				
Return value	1 = block number 2 = label 3 = string 4 = program level 5 = search pointer line-oriented (searching for line feeds)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access and write access	

seekOffset (C, SPARPF)				
Search pointer Search pointer (line-oriented, searching for line feeds). If the search pointer is used, a program name (progName) must always have been defined. The search pointer refers to this program.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			

seekOffset (C, SPARPF)				
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access and write access	

status (C, SPARPF)				
Without function This variable is without function in block SPARPF. It has only been introduced to achieve the same structure of SPARPI and SPARPF.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

workPName (C, SPARPF)				
Workpiece name abbreviated Workpiece name = path name in the NC file structure. If no path name is specified, the default search strategy for subroutine calls is used.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access and write access	

workPNameLong (C, SPARPF)				
Workpiece name Workpiece name of the active workpiece Note: This variable is ignored when lines are accessed.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access and write access	

workPNameL (C, SPARPF)				
Workpiece name Workpiece name = path name in the NCK file structure. If no path name is specified, the default search strategy for subroutine calls is used. Note: This variable is ignored when lines are accessed!				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access and write access	

haltBlock (C, SPARPF)				
Hold block The following applies to the SPARPI: The interrupt pointer does not mark the block where the program was cancelled but a previous block (hold block), which enables a better resumption. The hold block is explicitly set with the part program commands IPTRLOCK and IPTRUNLOCK, or implicitly manipulated via $MC_AUTO_IPTR_LOCK. The following applies to the SPARPF: If SPARPI is copied completely, the value of the hold block which was set by the NCK is retained. This enables the NCK to recognize the situation, and it responds with the suppressed message alarm 16950. NOTE: For SPARPI and SPARPF this value is ONLY defined for program level 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

displayState (C, SPARPF)				
Display status for block display. Display state for block display. (Blocks should not be displayed automatically for program levels for which DISPLAY OFF has been programmed in the PROC instruction. This is valid also for the subroutine levels below).				
Return value	Value Meaning 0 = DISPLAY OFF for the program level 1 = DISPLAY ON for the program level			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		

displayState (C, SPARPF)		
Addressing		
Line index:	Max. line index:	
	18	
	Read access	

byteOffset (C, SPARPF)				
Search pointer (byte-oriented)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:		Max. line index:		
		18		
		Read access		

3.2.3.28 SPARPI: Program pointer at interruption

In order to be able to continue at the point of interruption in a program, the current states of the main program and any subroutines must be stored. On a program interrupt the information is immediately updated in the NCK and reamins valid even after RESET.

This makes it possible to read the states of the main program level and the 11 subroutine levels (incl. ASUB levels).

The array indices (row indices) mean:

1 = main program level
2-18 = subroutine levels

forward (C, SPARPI)				
Search direction				
Return value	2 = forwards			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:		Max. line index:		
		18		
Validity:	From NCK version 3.6	Read access		

invocCount (C, SPARPI)		
Actual value of subroutine call counter. Actual value of the subroutine call counter. Is always 1 for the main program.		

invocCount (C, SPARPI)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

plcStartReason (C, SPARPI)				
Specification of channel for SERUPRO Specifies for the SERUPRO function which channel has to be started by the PLC so that the current channel starts.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

progName (C, SPARPI)				
Program name Progam name of the currently active program (or subroutine)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

searchString (C, SPARPI)				
Search string Search string (the first 64 characters of the NC block - corresponding to the search pointer)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

searchType (C, SPARPI)				
Search type				
Return value	5 = search pointer line-oriented (searching for line feed characters)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

seekOffset (C, SPARPI)				
Search pointer Search pointer (line-oriented, searching for linefeed characters) 1fffffff HEX is returned if the value is invalid.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

status (C, SPARPI)				
Status of block SPARPI Informs about whether block SPARPI includes currently valid values, and provides the reason for the last update of the block, if available. Note: If an interruption occurs in a program range between the command IPTRLOCK and IPTRUNLOCK, the first block after IPTRLOCK will be provided in the SPARPI instead of the current block. The first interruption between IPTRLOCK and IPTRUNLOCK will set "status" and any additional interruption prior to IPTRUNLOCK will neither change "status" nor SPARPI.				
Return value	0: Program is running, i.e. SPARPI variables are not up-to-date 1: Program selection, i.e. SPARPI has been reset 2: Block selection through PI service _N_SEL_BL 3: Reset (program abort) 4: Stop after program instruction, e.g. M0 5: Stop with STOP key 6: Stop caused by alarm			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	0	6

status (C, SPARPI)				
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

workPName (C, SPARPI)				
Workpiece name abbreviated Workpiece name = path name in the NC file structure				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

workPNameLong (C, SPARPI)				
Workpiece name Name of the active workpiece				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

workPNameL (C, SPARPI)				
Workpiece name Workpiece name = path name in the NCK file structure Note: This variable is ignored when lines are accessed!				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

haltBlock (C, SPARPI)				
Hold block The following applies to the SPARPI: The interrupt pointer does not mark the block where the program was cancelled but a previous block (hold block), which enables a better resumption. The hold block is explicitly set with the part program commands IPTRLOCK and IPTRUNLOCK, or implicitly manipulated via $MC_AUTO_IPTR_LOCK. The following applies to the SPARPF: If SPARPI is copied completely, the value of the hold block which was set by the NCK is retained. This enables the NCK to recognize the situation, and it responds with the suppressed message alarm 16950. NOTE: For SPARPI and SPARPF this value is ONLY defined for program level 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

displayState (C, SPARPI)				
Display status for block display. Display state for block display. (Blocks should not be displayed automatically for program levels for which DISPLAY OFF has been programmed in the PROC instruction. This is valid also for the subroutine levels below).				
Return value	Value Meaning 0 = DISPLAY OFF for the program level 1 = DISPLAY ON for the program level			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

byteOffset (C, SPARPI)				
Search pointer (byte-oriented)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

3.2.3.29 SPARPP: Program pointer in automatic operation mode

In automatic mode it is possible to branch to several subroutine levels from the main program level. The state of the program can be determined for every program level. Each variable of the module consists of 18 rows. This makes it possible to address the main program level and 17 subroutine levels (incl. ASUB levels).

The array indices (row indices) mean:

1 = main program level
2-18 = subroutine levels

actInvocCount (C, SPARPP)				
Actual value of subroutine call counter. Subroutine call counter, actual value. Specifies the number of subroutine passes. Is always set 1 for the main program and for asynchronous subroutines.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

blockLabel (C, SPARPP)				
Block label				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

blockNoStr (C, SPARPP)				
Block number				
Return value	[:][N]<No>			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

cmdInvocCount (C, SPARPP)				
Subroutine call counter, desired value Subroutine call counter, desired value. Specifies the number of subroutine passes. Is always set to 1 for the main program and for asynchronous subroutines.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

displayState (C, SPARPP)				
Display state for block display Display state for block display. (Blocks should not be displayed automatically for program levels for which DISPLAY OFF has been programmed in the PROC instruction. This is valid also for the subroutine levels below).				
Return value	Value Meaning 0 = DISPLAY OFF for the program level 1 = DISPLAY ON for the program level			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

extProgBufferName (C, SPARPP)				
FIFO buffer for execution from external source Name of FIFO buffer for execution from external source				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access and write access	

extProgFlag (C, SPARPP)				
External execution Indicates whether programs are being executed externally				
Return value	0: Program is being processed from NCK program memory 1: Program is being executed externally 2: Program is being executed in EES mode			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

lastBlockNoStr (C, SPARPP)				
Last programmed block number Returns the last programmed block number for each program level when $MN_DISPLAY_FUNCTION_MASK bit 0 is set. A block number is shown until either a new block number is programmed or the subroutine level which generated the block number has been left. Block numbers of masked blocks are not displayed. There is also no display if DISPLOF is active.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

progName (C, SPARPP)				
Program name Progam name of the currently active program (or subroutine)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

seekOffset (C, SPARPP)				
Search pointer Search pointer (block offset, each block consists of a string that ends with a line feed)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

seekw (C, SPARPP)				
Line enabled for modification First line enabled for modification in part program				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

workPName (C, SPARPP)				
Workpiece name abbreviated Workpiece name = path name in the NC file structure				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
Validity:	From NCK version 3.6		Read access	

workPNameLong (C, SPARPP)				
Workpiece name Workpiece name of the active workpiece Note: This variable is ignored when lines are accessed.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	

workPNameLong (C, SPARPP)				
			18	
			Read access	

workPandProgName (C, SPARPP)				
Workpiece and program name Workpiece name and name of current program.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

byteOffset (C, SPARPP)				
Byte offset of NC block Byte offset of the current NC block				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

progNameVL (C, SPARPP)				
Program name VL Program name in the preprocessing				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

workPandProgNameVL (C, SPARPP)				
Workpiece name and program name VL Workpiece name and program name of the current program in the preprocessing.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

workPandProgNameVL (C, SPARPP)				
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

blockNoStrVL (C, SPARPP)				
Block number VL Block number in the preprocessing				
Return value	[:][N]<No>			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

seekOffsetVL (C, SPARPP)				
Search pointer VL Search pointer in the preprocessing (block offset, each block consists of a string that ends with a line feed)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

byteOffsetVL (C, SPARPP)				
Byte offset of NC block VL Byte offset of the current NC block in the preprocessing				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

actInvocCountVL (C, SPARPP)				
Subroutine call counter, actual value VL Subroutine call counter, actual value in the preprocessing. Specifies the number of subroutine passes. It is always set to 1 for the main program and for asynchronous subroutines.				

actInvocCountVL (C, SPARPP)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

programAsciiBlock (C, SPARPP)				
getProgramAsciiBlock Value of the CC binding getProgramAsciiBlock. Use released only for internal development test purposes!				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

eesBufferStart (C, SPARPP)				
Byte offset of the first block in the EES buffer The value is only relevant with EES, when the part program is partially loaded into a buffer NCK. It indicates wthich NC block is entered first in the buffer.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

eesBufferEnd (C, SPARPP)				
Byte offset following the last block in the buffer The value is only relevant with EED, when the part program is partially loaded into a buffer of the NCK. It indicates which NC block has been entered last in the buffer. The comparison with the value $$byteOffsetVL shows whether the reloading process is performed quickly enough in order to perform the preprocessing run without delay.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	

eesBufferEnd (C, SPARPP)				
			18	
			Read access	

eesBufferFilling (C, SPARPP)				
Number of bytes for interpreter The value is only relevant with EES, when the part program is partially loaded into a buffer of the NCK. It indicates who many bytes are provided to the interpreter for processing in a the buffer ($$eesBufferEnd - $$byteOff-setVL). If the value approaches 0, this indicates that the reloading process is not performed quickly enough in order to supply the interpreter quickly enough with NC blocks.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

eesBufferStatus (C, SPARPP)				
Status of the EES buffer				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			18	
			Read access	

3.2.3.30 SSP: State data spindle

All status data that refer to the spindle are combined in the module SSP. The individual variables are defined as arrays where the row index is the number of the spindle (assigned to the current channel). The spindle can be identified by reading the variables "name" or "index" in the same module with the respective row index.

The number of spindles can be read from "$$numSpindles" in the module Y in the area C.

acConstCutS (C, SSP)		$AC_CONSTCUT_S[n]		
Current const. cutting rate Current constant cutting rate				
Units and value ranges				
Physical unit:	Data type:	Initial value		
m/min, ft/min, user defined	TYPE_DOUBLE	0		

acConstCutS (C, SSP)	$AC_CONSTCUT_S[n]	
Addressing		
Line index:	Max. line index:	
	$$numSpindles	
	Read access	

acSMode (C, SSP)		$AC_SMODE[x]		
Mode according to $AC_SMODE Spindle mode				
Return value	0: No spindle present in channel or spindle is active in another channel or is being used by PLC (FC18) or by synchronized actions. 1: Open-loop speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	0	4
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

actGearStage (C, SSP)				
Actual gear stage of spindle				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

actSpeed (C, SSP)		$AA_S[x] x = SpindleNo		
Actual value of the spindle speed Current speed of the spindle [n]. The sign indicates the direction of rotation. The direction of rotation can vary from machine to machine, and is specified by the machine manufacturer.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

actSpeed (C, SSP)		$AA_S[x] x = SpindleNo		
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

channelNo (C, SSP)				
Channel number of spindle Number of channel in which spindle is configured				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

cmdAngPos (C, SSP)				
Spindle position Spindle position (SPOS)				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

cmdConstCutSpeed (C, SSP)				
Constant cutting speed of the master spindle Constant cutting rate of the master spindle. The requested value for the master spindle differs from SSP:cmdSpeed only if G96 is active. (For a certain OEM customer this variable is now available retroactively in software version 3.2)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

cmdGearStage (C, SSP)				
Requested gear stage				

cmdGearStage (C, SSP)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

cmdGwps (C, SSP)				
Programmed SUG desired value Programmed SUG desired value (SUG is the function "constant perimeter speed of grinding wheel")				
Units and value ranges				
Physical unit:	Data type:			
m/s, ft/s	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

cmdSpeed (C, SSP)		$P_S[x] x = SpindleNo		
Spindle speed, desired value Spindle speed desired value				
Units and value ranges				
Physical unit:	Data type:			
rev/min , m/min	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

driveLoad (C, SSP)				
Load				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

gwpsActive (C, SSP)		{$GWPS}		
SUG programming active SUG programming active (SUG is the function "constant perimeter speed of grinding wheel)				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

index (C, SSP)				
Absolute axis index Absolute axis index referred to MD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

name (C, SSP)				
Spindle name Spindle name Note: If several logical spindles are referred to one physical spindle with active spindle conversion and access is made via area N of module SSP2, then the name of the first suitable logical spindle is output.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

namePhys (C, SSP)				
Name of physical spindle. Name of assigned physical spindle, identical to "name" variable.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

namePhys (C, SSP)		
Addressing		
Line index:	Max. line index:	
	$$numSpindles	
	Read access	

opMode (C, SSP)				
Spindle mode				
Return value	0 = spindle mode 1 = oscillation mode (gear step changeover) 2 = positioning mode 3 = synchronous mode 4 = axis mode			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

pSMode (C, SSP)		$P_SMODE		
Prog. spindle mode Last programmed spindle mode				
Return value	0: No spindle configured in channel or spindle is active in another channel or in use by the PLC (FC18) or by synchronized actions. 1: Speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	4
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

pSModeS (C, SSP)				
Prog. spindle mode block search Last programmed spindle mode with block search				
Return value	0: No spindle configured in channel or spindle is active in another channel or in use by the PLC (FC18) or by synchronized actions. 1: Speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	4
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

psModePos (C, SSP)				
Corresponds to $$/C/SEGA/actToolEdgeCenterPosEns If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$/C/SEGA/actToolEdgeCenterPosEns is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

psModePosBKS (C, SSP)				
Corresponds to $$/C/SEGA/actProgPosBKS If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$/C/SEGA/actProgPosBKS is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

psModePosS (C, SSP)				
Corresponds to $$cmdToolEdgeCenterPosEnsS If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$cmdToolEdgeCenterPosEnsS is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

speedLimit (C, SSP)				
Current speed limitation for spindle				
Units and value ranges				
Physical unit:	Data type:			
rev/min , m/min	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

speedOvr (C, SSP)				
Spindle override				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

spindleType (C, SSP)				
Spindle type				
Return value	0 = master spindle 1 = no master spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

status (C, SSP)				
Spindle state Spindle status Bit0 = following spindle Bit1 = leading spindle Bit2 = master spindle Bit3 = constant cutting rate (G96) active				
Return value	Bit0 = following spindle Bit1 = leading spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

turnState (C, SSP)				
State of spindle rotation				
Return value	value range to be read via BTSS variable 0 = clockwise 1 = counter-clockwise 2 = stop value range to be read via $ variable 3 = clockwise 4 = counter-clockwise 5 = stop			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
Validity:	From NCK version 3.6		Read access	

acSDir (C, SSP)		$AC_SDIR[x] x = SpindleNo		
Current dir. of spindle rot. in sense of M3/M4/M5 Current direction of spindle rotation in the sense of M3/M4/M5 in the part program, synchronized actions, PLC FC18, PLC DBB30. 3: Clockwise spindle rotation, 4: Counterclockwise spindle rotation, 5: Spindle stop				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

acSDir (C, SSP)	$AC_SDIR[x] x = SpindleNo	
Addressing		
Line index:		Max. line index:
		$$numSpindles
		Read access

acSmaxVelo (C, SSP)	$AC_SMAXVELO[]			
Maximum spindle speed Maximum spindle speed This variable returns the maximum spindle speed for spindle mode. This is formed from the smallest active speed limitation, and cannot be exceeded by speed programming or override > 100%. A speed limitation is indicated by the VDI interface signal DB31..,DBX83.1 'Setpoint speed limited' and by $AC_SPIND_STATE, bit 10 (setpoint speed limited). The cause of the speed limitation (machine, setting data, G code, VDI interface signal etc.) can also be determined with the system variable $AC_SMAXVELO_INFO. If the spindle is in axis mode, then the speed is not limited by $AC_SMAXVELO but instead the machine data (MAX_AX_VELO, ...) typical for axis mode are active.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

acSmaxVeloInfo (C, SSP)	$AC_SMAXVELO_INFO[]	

Max. identifier for speed-limiting data

Identifier (index) for the speed-limiting data (machine/setting data, etc.)

The system variable provides additional information about $AC_SMAXVELO and returns the definitive data

(machine, setting data, G code, VDI interface etc.) as identifier/index. The index can be used to determine the speed-limiting data

on the basis of the following table of existing spindle speed limitations.

0 No limitation (SERUPRO)

1 Maximum speed (chuck speed) of spindle MD 35100 SPIND_VELO_LIMIT

2 Speed limited to maximum speed in the current gear stage MD 35130 GEAR_STEP_MAX_VELO_LIMIT

3 Speed limited due to position control to 90% of the minimum from MD 35100 and MD 35130 (SPCON, SPOS, poss. with COUPON,..)

4 Speed limited due to position control to MD 35132 GEAR_STEP_PC_MAX_VELO_LIMIT

5 Speed limited to SD 43220 SPIND_MAX_VELO_G26 (G26 S.. or specification from HMI)

6 Speed limited to MD 35160 SPIND_EXTERN_VELO_LIMIT based on the set VDI interface signal DB31,...DBX3.6

7 Speed limited to SD 43230 SPIND_MAX_VELO_LIMS at constant cutting speed (G96, G961, G962, G97, LIMS)

8 Speed limited to safe speed (SG) by Safety Integrated

9 Speed limited by preparation calculations

10 Limitation by drive parameter SINAMICS p1082 to maximum speed of the drive

11 Speed limitation to MD 36300 ENC_FREQ_LIMIT with functions that require a functioning measuring system, e.g. position control and G95, G96, G97, G973, G33, G34, G35 for the master spindle. The limitation takes into account the encoder speed, the MS arrangement (direct/indirect), MS limiting frequency and the current parameter set

12 Speed limited by axis mode. In the case of a synchronous spindle, axis mode is enforced by the leading spindle.

13 Speed of the superimposed motion of the following spindle limited to the residual dynamics remaining following the coupling. A larger proportion of the superimposed motion can be achieved by reducing the speed of the leading spindle, e.g. by programming G26 S, VELOLIM for the leading spindle or VELOLIMA for the following spindle. The coupling factor must be taken into account.

14 Speed of the leading spindle limited due to missing following spindle dynamics or a high transformation ratio

15 Speed of the master spindle limited to MD 35550 DRILL_VELO_LIMIT in the case of tapping with G331, G332

16 Speed limitation due to the programming of VELOLIM

17 Speed limitation by tool parameter $TC_TP_MAX_VELO

18 Not used

19 Not used

20 Speed limited due to NCU link

21 Speed limited by SD43235 SD_SPIND_USER_VELO_LIMIT, user-controlled speed limitation, e.g. tensioning device, chuck speed

22 Speed limited by the programming of VELOLIMA

23 Speed limited by the clamping state of the tool. In the case of a Weiss spindle, the clamping state can be read from $VA_MOT_CLAMPING_STATE[axn].

In oscillation mode (gear stage change), the variable returns the value for spindle mode (speed control mode).

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			

acSmaxVeloInfo (C, SSP)	$AC_SMAXVELO_INFO[]	
Addressing		
Line index:	Max. line index:	
	$$numSpindles	
	Read access	

acSminVelo (C, SSP)	$AC_SMINVELO[]			
Minimum spindle speed Minimum spindle speed This variable returns the minimum spindle speed for speed control mode. This is formed from the highest active speed increase, and cannot be undershot by speed programming or override < 100%. A speed increase is indicated by the VDI interface signal DB31..,DBX83.2 'Setpoint speed increased' and by $AC_SPIND_STATE, bit 11 (setpoint speed increased). The cause of the speed increase (machine, setting data, G code, VDI interface signal etc.) can also be determined with the system variable $AC_SMINVELO_INFO. If the spindle is in axis or positioning mode, then the speed is not increased by $AC_SMINVELO.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

acSminVeloInfo (C, SSP)	$AC_SMINVELO_INFO[]	
Min. identifier for speed-limiting data Identifier (index) for the speed-limiting data (machine/setting data, etc.) The system variable provides additional information about $AC_SMAXVELO, and returns the definitive data (machine, setting data, G code, VDI interface, etc.) as identifier/index. The speed-limiting data can be determined with the index from the following table of existing spindle speed limitations. The system variable provides additional information about $AC_SMINVELO, and returns the speed increasing data (machine, setting data) as identifier/index. The speed-increasing data can be determined with the index from the following table of existing spindle speed increases. 0 Not used 1 Not used 2 Lower speed limit (minimum speed) of the current gear stage MD 35140 GEAR_STEP_MIN_VELO_LIMIT 3 Not used 4 Not used 5 Lower speed limit (minimum speed) from SD 43210 SPIND_MIN_VELO_G25 (G25 S.. or specification from HMI) In oscillation mode (gear stage change) and axis mode, the variable returns the values from spindle mode.		

acSminVeloInfo (C, SSP)		$AC_SMINVELO_INFO[]		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

acSmaxAcc (C, SSP)		$AC_SMAXACC[]		
Active acceleration of the spindle Active acceleration of the spindle This variable returns the active acceleration of the spindle for spindle mode. Bit 14 of $AC_SPIND_STATE (spindle accelerating) is set for the duration of the acceleration to the defined setpoint speed. Bit 15 of $AC_SPIND_STATE (spindle braking) is set for the duration of the braking to the defined setpoint speed. Apart from that, the acceleration-determining machine and setting data can be determined with the system variable $AC_SMAXACC_INFO. If the spindle is in axis mode, then $AC_SMAXACC does not return the current acceleration, instead the machine data (MAX_AX_VELO, MAX_AX_ACCEL, ...) typical for axis mode are active.				
Units and value ranges				
Physical unit:	Data type:			
Rev/s2, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindles	
			Read access	

acSmaxAccInfo (C, SSP)	$AC_SMAXACC_INFO[]	

Identifier for active spindle acceleration data

Identifier for the active spindle acceleration data

The system variable provides additional information about $AC_SMAXACC and returns the definitive machine data as identifier/index. The index can be used to determine the active acceleration data on the basis of the following table of existing spindle accelerations.

The number range is oriented to the system variable $AC_SMAXVELO_INFO:

0 No acceleration limitation (SERUPRO)

1 Not used

2 Acceleration in speed control mode without position control in the current gear stage MD 35200 GEAR_STEP_SPEEDCTRL_ACCEL

3 Not used

4 Acceleration in the current gear stage based on position control MD 35210 GEAR_STEP_POSCTRL_ACCEL (SPCON, SPOS, poss. with COUPON,..)

5 Not used

6 Not used

7 Not used

8 Not used

9 Acceleration limited by preparation calculations

10 Not used

11 Not used

12 Acceleration limited by axis mode. In the case of a synchronous spindle, the axis mode is enforced by the leading spindle.

13 Acceleration of the superimposed motion of the following spindle limited to the residual dynamics remaining following the coupling

14 Acceleration of the leading spindle due to missing following spindle dynamics or a high transformation ratio

15 Acceleration of the master spindle MD 35212 GEAR_STEP_POSCTRL_ACCEL2 in the case of tapping with G331, G332 (only when the second data set is configured accordingly)

16 Acceleration limited by the configuration of ACC or ACCFXS (synchronized action)

17 Acceleration limited by tool parameter $TC_TP_MAX_ACCEL

18 Not used

19 Acceleration limited in JOG mode by MD 32301 MA_JOG_MAX_ACCEL

20 Acceleration limited due to NCU link

21 Not used

22 Acceleration limited by programming ACCLIMA

23 Not used

In oscillation mode (gear stage change), the variable returns the value for spindle mode (speed control mode).

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			

Addressing

Line index:	Max. line index:

acSmaxAccInfo (C, SSP)	$AC_SMAXACC_INFO[]	
	$$numSpindles	
	Read access	

acSpindState (C, SSP)	$AC_SPIND_STATE[]	
Spindle status in speed control mode		
This variable returns the selected states of the spindle. For positioning and axis modes, the variable $AA_IN-POS_STATE[Sn] can also be read.		
Bit 0: "Constant cutting speed active" (VDI interface signal DB31..,DBX84.0)		
Bit 1: "SUG active" (VDI interface signal DB31..,DBX84.1)		
Bit 2: "CLGON active" (VDI interface signal DB31..,DBX84.2)		
Bit 3: "Tapping without compensating chuck" (VDI interface signal DB31..,DBX84.3)		
Bit 4: "Synchronous mode" (following spindle with synchronous spindle coupling) (VDI interface signal DB31..,DBX84.4)		
Bit 5: "Positioning mode" (VDI interface signal DB31..,DBX84.5)		
Bit 6: "Oscillating mode" (gear stage change) (VDI interface signal DB31..,DBX84.6)		
Bit 7: "Speed control mode" (VDI interface signal DB31..,DBX84.7)		
Bit 8: "Spindle programmed" (e.g. M3, M4 S.., FC18, ..) (VDI interface signal DB31..,DBX64.4/5 or 6/7)		
Bit 9: "Speed limit exceeded" (VDI interface signal DB31..,DBX83.0)		
Bit 10: "Setpoint speed limited" (VDI interface signal DB31..,DBX83.1), active if the speed would be greater than the maximum speed as a result of programming or override ($AC_SMAXVELO)		
Bit 11: "Setpoint speed increased" (VDI interface signal DB31..,DBX83.2) active if the speed would be less than the minimum speed as a result of programming or override (system variable $AC_SMINVELO)		
Bit 12: "Spindle in setpoint range" (VDI interface signal DB31..,DBX83.5)		
Bit 13: "Actual direction of rotation right" (VDI interface signal DB31..,DBX83.7)		
Bit 14: "Spindle accelerating" remains active as long as the spindle is accelerating to the defined setpoint speed on the setpoint side.		
Bit 15: "Spindle braking" remains active as long as the spindle is braking to the defined setpoint speed or comes to a standstill on the setpoint side.		
Bit 16: "Spindle stopped" (VDI interface signal DB31..,DBX61.4)		
Bit 17: "Tool with dynamic limitation active" (VDI interface signal DB31..,DBX85.0)		
Bit 18: Reserved		
Bit 19: "Spindle in position" (VDI interface signal DB31..,DBX85.5)		
Bit 20: "Position control active" (VDI interface signal DB31..,DBX61.5)		
Bit 21: "Referenced/synchronized 1" (VDI interface signal DB31..,DBX60.4)		
Bit 22: "Referenced/synchronized 2" (VDI interface signal DB31..,DBX60.5)		
Bit 23: Direction of spindle rotation inverted by interface signal "Invert M3/M4" (DB31..,DBX17.6)		

Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			

Addressing	
Line index:	Max. line index:
	$$numSpindles
	Read access

acSVC (C, SSP)		$AC_SVC[x]		
Programmed, active cutting rate				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
			Read access	

acSType (C, SSP)		$AC_S_TYPE[x]		
Type of spindle programming Type of spindle programming Spindle programmed as: 0 Spindle not programmed 1 Spindle speed, S in rev/min 2 Cutting rate, SVC in m/min or ft/min 3 Constant cutting rate, S in m/min or ft/min 4 Constant grinding wheel peripheral speed, S in m/s or ft/s				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
			Read access	

vcSGear (C, SSP)		$VC_SGEAR[spino]		
Currently active spindle gear stage Variable $VC_SGEAR[spino] determines the currently active spindle gear stage. $AC_SGEAR[spino] determines the defined gear stage in the main run. During search the actual gear stage may differ from the defined gear stage, because during search the gear stages are not changed. Using $VC_SGEAR[spino] and $AC_SGEAR[spino] it can be checked whether a gear stage change is to be performed after a search. The following values are possible: 1: 1st gear stage active .... 5: 5th gear stage active				
Return value	1: 1st gear stage active .... 5: 5th gear stage active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_WORD	0	0	5
			Read access	

acSGear (C, SSP)	$AC_SGEAR[spino]			
Active spindle gear stage Variable $VC_SGEAR[spino] determines the currently active spindle gear stage. $AC_SGEAR[spino] determines the defined gear stage in the main run. During search the actual gear stage may differ from the defined gear stage, because during search the gear stages are not changed. Using $VC_SGEAR[spino] and $AC_SGEAR[spino] it can be checked whether a gear stage change is to be performed after a search. The following values are possible: 1: 1st gear stage active .... 5: 5th gear stage active				
Return value	1: 1st gear stage active .... 5: 5th gear stage active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_WORD	0	0	5
			Read access	

3.2.3.31 SSP2: Spindle status data for spindle conversion

All state data that refer to a spindle, if a spindle converter (logical spindles) is active

acConstCutS (C, SSP2)				
Current const. cutting rate Current constant cutting rate				
Units and value ranges				
Physical unit:	Data type:	Initial value		
m/min, ft/min, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

acSMode (C, SSP2)				
Mode according to $AC_SMODE Spindle mode				
Return value	0: No spindle present in channel or spindle is active in another channel or is being used by PLC (FC18) or by synchronized actions. 1: Open-loop speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			

acSMode (C, SSP2)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	0	4
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

actGearStage (C, SSP2)				
Actual gear stage of spindle				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

actSpeed (C, SSP2)				
Actual value of the spindle speed Current speed of the spindle [n]. The sign indicates the direction of rotation. The direction of rotation can vary from machine to machine, and is specified by the machine manufacturer.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

channelNo (C, SSP2)				
Channel number of spindle Number of channel in which spindle is configured				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

cmdAngPos (C, SSP2)				
Spindle position Spindle position (SPOS)				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

cmdConstCutSpeed (C, SSP2)				
Constant cutting speed of the master spindle Constant cutting rate of the master spindle. The requested value for the master spindle differs from SSP:cmdSpeed only if G96 is active. (For a certain OEM customer this variable is now available retroactively in software version 3.2)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

cmdGearStage (C, SSP2)				
Requested gear stage				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

cmdGwps (C, SSP2)				
Programmed SUG desired value Programmed SUG desired value (SUG is the function "constant perimeter speed of grinding wheel")				
Units and value ranges				
Physical unit:	Data type:			
m/s, ft/s	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

cmdGwps (C, SSP2)				
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

cmdSpeed (C, SSP2)				
Spindle speed, desired value Spindle speed desired value				
Units and value ranges				
Physical unit:	Data type:			
rev/min , m/min	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

driveLoad (C, SSP2)				
Load				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

gwpsActive (C, SSP2)				
SUG programming active SUG programming active (SUG is the function "constant perimeter speed of grinding wheel)				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

index (C, SSP2)				
Absolute axis index Absolute axis index referred to MD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

index (C, SSP2)				
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

name (C, SSP2)				
Spindle name Spindle name Note: If several logical spindles are referred to one physical spindle with active spindle conversion and access is made via area N of module SSP2, then the name of the first suitable logical spindle is output.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

namePhys (C, SSP2)				
Physical spindle name Name of assigned physical spindle, identical to "name" variable.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

opMode (C, SSP2)				
Spindle mode				
Return value	0 = spindle mode 1 = oscillation mode (gear step changeover) 2 = positioning mode 3 = synchronous mode 4 = axis mode			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

pSMode (C, SSP2)				
Prog. spindle mode Last programmed spindle mode				
Return value	0: No spindle configured in channel or spindle is active in another channel or in use by the PLC (FC18) or by synchronized actions. 1: Speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	4
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

pSModeS (C, SSP2)				
Prog. spindle mode block search Last programmed spindle mode with block search				
Return value	0: No spindle configured in channel or spindle is active in another channel or in use by the PLC (FC18) or by synchronized actions. 1: Speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	4
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

psModePos (C, SSP2)				
Corresponds to $$/C/SEGA/actToolEdgeCenterPosEns If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$/C/SEGA/actToolEdgeCenterPosEns is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	

psModePos (C, SSP2)				
			$$numSpindlesLog	
			Read access	

psModePosBKS (C, SSP2)				
Corresponds to $$/C/SEGA/actProgPosBKS If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$/C/SEGA/actProgPosBKS is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

psModePosS (C, SSP2)				
Corresponds to $$cmdToolEdgeCenterPosEnsS If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$cmdToolEdgeCenterPosEnsS is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

speedLimit (C, SSP2)				
Current speed limitation for spindle				
Units and value ranges				
Physical unit:	Data type:			
rev/min , m/min	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

speedOvr (C, SSP2)				
Spindle override				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			

speedOvr (C, SSP2)				
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

spindleType (C, SSP2)				
Spindle type				
Return value	0 = master spindle 1 = no master spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

status (C, SSP2)				
Spindle state Spindle status Bit0 = following spindle Bit1 = leading spindle Bit2 = master spindle Bit3 = constant cutting rate (G96) active				
Return value	Bit0 = following spindle Bit1 = leading spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

turnState (C, SSP2)				
State of spindle rotation				
Return value	value range to be read via BTSS variable 0 = clockwise 1 = counter-clockwise 2 = stop value range to be read via $ variable 3 = clockwise 4 = counter-clockwise 5 = stop			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
Validity:	From NCK version 3.6		Read access	

acSDir (C, SSP2)				
Programmed direction of spindle rotation Programmed direction of spindle rotation in part program, synchronized actions, PLC FC18, PLC DBB30. 3: Clockwise spindle rotation, 4: Counterclockwise spindle rotation, 5: Spindle stop				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

acSmaxVelo (C, SSP2)				
Maximum spindle speed Maximum spindle speed This variable returns the maximum spindle speed for spindle mode. This is formed from the smallest active speed limitation, and cannot be exceeded by speed programming or override > 100%. A speed limitation is indicated by the VDI interface signal DB31..,DBX83.1 'Setpoint speed limited' and by $AC_SPIND_STATE, bit 10 (setpoint speed limited). The cause of the speed limitation (machine, setting data, G code, VDI interface signal etc.) can also be determined with the system variable $AC_SMAXVELO_INFO. If the spindle is in axis mode, then the speed is not limited by $AC_SMAXVELO but instead the machine data (MAX_AX_VELO, ...) typical for axis mode are active.				

acSmaxVelo (C, SSP2)				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

acSmaxVeloInfo (C, SSP2)		

Max. identifier for speed-limiting data

Identifier (index) for the speed-limiting data (machine/setting data, etc.)

The system variable provides additional information about $AC_SMAXVELO and returns the definitive data

(machine, setting data, G code, VDI interface etc.) as identifier/index. The index can be used to determine the speed-limiting data

on the basis of the following table of existing spindle speed limitations.

0 No limitation (SERUPRO)

1 Maximum speed (chuck speed) of spindle MD 35100 SPIND_VELO_LIMIT

2 Speed limited to maximum speed in the current gear stage MD 35130 GEAR_STEP_MAX_VELO_LIMIT

3 Speed limited due to position control to 90% of the minimum from MD 35100 and MD 35130 (SPCON, SPOS, poss. with COUPON,..)

4 Speed limited due to position control to MD 35132 GEAR_STEP_PC_MAX_VELO_LIMIT

5 Speed limited to SD 43220 SPIND_MAX_VELO_G26 (G26 S.. or specification from HMI)

6 Speed limited to MD 35160 SPIND_EXTERN_VELO_LIMIT based on the set VDI interface signal DB31,...DBX3.6

7 Speed limited to SD 43230 SPIND_MAX_VELO_LIMS at constant cutting speed (G96, G961, G962, G97, LIMS)

8 Speed limited to safe speed (SG) by Safety Integrated

9 Speed limited by preparation calculations

10 Limitation by drive parameter SINAMICS p1082 to maximum speed of the drive

11 Speed limitation to MD 36300 ENC_FREQ_LIMIT with functions that require a functioning measuring system, e.g. position control and G95, G96, G97, G973, G33, G34, G35 for the master spindle. The limitation takes into account the encoder speed, the MS arrangement (direct/indirect), MS limiting frequency and the current parameter set

12 Speed limited by axis mode. In the case of a synchronous spindle, axis mode is enforced by the leading spindle.

13 Speed of the superimposed motion of the following spindle limited to the residual dynamics remaining following the coupling. A larger proportion of the superimposed motion can be achieved by reducing the speed of the leading spindle, e.g. by programming G26 S, VELOLIM for the leading spindle or VELOLIMA for the following spindle. The coupling factor must be taken into account.

14 Speed of the leading spindle limited due to missing following spindle dynamics or a high transformation ratio

15 Speed of the master spindle limited to MD 35550 DRILL_VELO_LIMIT in the case of tapping with G331, G332

16 Speed limitation due to the programming of VELOLIM

17 Speed limitation by tool parameter $TC_TP_MAX_VELO

18 Not used

19 Not used

20 Speed limited due to NCU link

21 Speed limited by SD43235 SD_SPIND_USER_VELO_LIMIT, user-controlled speed limitation, e.g. tensioning device, chuck speed

22 Speed limited by the programming of VELOLIMA

23 Speed limited by the clamping state of the tool. In the case of a Weiss spindle, the clamping state can be read from $VA_MOT_CLAMPING_STATE[axn].

In oscillation mode (gear stage change), the variable returns the value for spindle mode (speed control mode).

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			

acSmaxVeloInfo (C, SSP2)		
Addressing		
Line index:	Max. line index:	
	$$numSpindlesLog	
	Read access	

acSminVelo (C, SSP2)				
Minimum spindle speed Minimum spindle speed This variable returns the minimum spindle speed for speed control mode. This is formed from the highest active speed increase, and cannot be undershot by speed programming or override < 100%. A speed increase is indicated by the VDI interface signal DB31..,DBX83.2 'Setpoint speed increased' and by $AC_SPIND_STATE, bit 11 (setpoint speed increased). The cause of the speed increase (machine, setting data, G code, VDI interface signal etc.) can also be determined with the system variable $AC_SMINVELO_INFO. If the spindle is in axis or positioning mode, then the speed is not increased by $AC_SMINVELO.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

acSminVeloInfo (C, SSP2)		
Min. identifier for speed-limiting data Identifier (index) for the speed-limiting data (machine/setting data, etc.) The system variable provides additional information about $AC_SMAXVELO, and returns the definitive data (machine, setting data, G code, VDI interface, etc.) as identifier/index. The speed-limiting data can be determined with the index from the following table of existing spindle speed limitations. The system variable provides additional information about $AC_SMINVELO, and returns the speed increasing data (machine, setting data) as identifier/index. The speed-increasing data can be determined with the index from the following table of existing spindle speed increases. 0 Not used 1 Not used 2 Lower speed limit (minimum speed) of the current gear stage MD 35140 GEAR_STEP_MIN_VELO_LIMIT 3 Not used 4 Not used 5 Lower speed limit (minimum speed) from SD 43210 SPIND_MIN_VELO_G25 (G25 S.. or specification from HMI) In oscillation mode (gear stage change) and axis mode, the variable returns the values from spindle mode.		

acSminVeloInfo (C, SSP2)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

acSmaxAcc (C, SSP2)				
Active acceleration of the spindle Active acceleration of the spindle This variable returns the active acceleration of the spindle for spindle mode. Bit 14 of $AC_SPIND_STATE (spindle accelerating) is set for the duration of the acceleration to the defined setpoint speed. Bit 15 of $AC_SPIND_STATE (spindle braking) is set for the duration of the braking to the defined setpoint speed. Apart from that, the acceleration-determining machine and setting data can be determined with the system variable $AC_SMAXACC_INFO. If the spindle is in axis mode, then $AC_SMAXACC does not return the current acceleration, instead the machine data (MAX_AX_VELO, MAX_AX_ACCEL, ...) typical for axis mode are active.				
Units and value ranges				
Physical unit:	Data type:			
Rev/s2, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numSpindlesLog	
			Read access	

acSmaxAccInfo (C, SSP2)		

Identifier for active spindle acceleration data

Identifier for the active spindle acceleration data

The system variable provides additional information about $AC_SMAXACC and returns the definitive machine data as identifier/index. The index can be used to determine the active acceleration data on the basis of the following table of existing spindle accelerations.

The number range is oriented to the system variable $AC_SMAXVELO_IDX:

0 No acceleration limitation (SERUPRO)

1 Not used

2 Acceleration in speed control mode without position control in the current gear stage MD 35200 GEAR_STEP_SPEEDCTRL_ACCEL

3 Not used

4 Acceleration in the current gear stage based on position control MD 35210 GEAR_STEP_POSCTRL_ACCEL (SPCON, SPOS, poss. with COUPON,..)

5 Not used

6 Not used

7 Not used

8 Not used

9 Acceleration limited by preparation calculations

10 Not used

11 Not used

12 Acceleration limited by axis mode. In the case of a synchronous spindle, the axis mode is enforced by the leading spindle.

13 Acceleration of the superimposed motion of the following spindle limited to the residual dynamics remaining following the coupling

14 Acceleration of the leading spindle due to missing following spindle dynamics or a high transformation ratio

15 Acceleration of the master spindle MD 35212 GEAR_STEP_POSCTRL_ACCEL2 in the case of tapping with G331, G332 (only when the second data set is configured accordingly)

16 Acceleration limited by the configuration of ACC or ACCFXS (synchronized action)

17 Acceleration limited by tool parameter $TC_TP_MAX_ACCEL

18 Not used

19 Acceleration limited in JOG mode by MD 32301 MA_JOG_MAX_ACCEL

20 Acceleration limited due to NCU link

21 Not used

22 Acceleration limited by programming ACCLIMA

23 Not used

In oscillation mode (gear stage change), the variable returns the value for spindle mode (speed control mode).

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			

Addressing

Line index:	Max. line index:

acSmaxAccInfo (C, SSP2)		
	$$numSpindlesLog	
	Read access	

acSpindState (C, SSP2)		
Spindle status in speed control mode		
This variable returns the selected states of the spindle. For positioning and axis modes, the variable $AA_IN-POS_STATE[Sn] can also be read.		
Bit 0: "Constant cutting speed active" (VDI interface signal DB31..,DBX84.0)		
Bit 1: "SUG active" (VDI interface signal DB31..,DBX84.1)		
Bit 2: "CLGON active" (VDI interface signal DB31..,DBX84.2)		
Bit 3: "Tapping without compensating chuck" (VDI interface signal DB31..,DBX84.3)		
Bit 4: "Synchronous mode" (following spindle with synchronous spindle coupling) (VDI interface signal DB31..,DBX84.4)		
Bit 5: "Positioning mode" (VDI interface signal DB31..,DBX84.5)		
Bit 6: "Oscillating mode" (gear stage change) (VDI interface signal DB31..,DBX84.6)		
Bit 7: "Speed control mode" (VDI interface signal DB31..,DBX84.7)		
Bit 8: "Spindle programmed" (e.g. M3, M4 S.., FC18, ..) (VDI interface signal DB31..,DBX64.4/5 or 6/7)		
Bit 9: "Speed limit exceeded" (VDI interface signal DB31..,DBX83.0)		
Bit 10: "Setpoint speed limited" (VDI interface signal DB31..,DBX83.1), active if the speed would be greater than the maximum speed as a result of programming or override ($AC_SMAXVELO)		
Bit 11: "Setpoint speed increased" (VDI interface signal DB31..,DBX83.2) active if the speed would be less than the minimum speed as a result of programming or override (system variable $AC_SMINVELO)		
Bit 12: "Spindle in setpoint range" (VDI interface signal DB31..,DBX83.5)		
Bit 13: "Actual direction of rotation right" (VDI interface signal DB31..,DBX83.7)		
Bit 14: "Spindle accelerating" remains active as long as the spindle is accelerating to the defined setpoint speed on the setpoint side.		
Bit 15: "Spindle braking" remains active as long as the spindle is braking to the defined setpoint speed or comes to a standstill on the setpoint side.		
Bit 16: "Spindle stopped" (VDI interface signal DB31..,DBX61.4)		
Bit 17: "Tool with dynamic limitation active" (VDI interface signal DB31..,DBX85.0)		
Bit 18: Reserved		
Bit 19: "Spindle in position" (VDI interface signal DB31..,DBX85.5)		
Bit 20: "Position control active" (VDI interface signal DB31..,DBX61.5)		
Bit 21: "Referenced/synchronized 1" (VDI interface signal DB31..,DBX60.4)		
Bit 22: "Referenced/synchronized 2" (VDI interface signal DB31..,DBX60.5)		
Bit 23: Direction of spindle rotation inverted by interface signal "Invert M3/M4" (DB31..,DBX17.6)		

Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			

Addressing	
Line index:	Max. line index:
	$$numSpindlesLog
	Read access

acSVC (C, SSP2)				
Programmed, active cutting rate				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
			Read access	

acSType (C, SSP2)				
Type of spindle programming Type of spindle programming Spindle programmed as: 0 Spindle not programmed 1 Spindle speed, S in rev/min 2 Cutting rate, SVC in m/min or ft/min 3 Constant cutting rate, S in m/min or ft/min 4 Constant grinding wheel peripheral speed, S in m/s or ft/s				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
			Read access	

vcSGear (C, SSP2)	$VC_SGEAR[spino]			
Currently active spindle gear stage Variable $VC_SGEAR[spino] determines the currently active spindle gear stage. $AC_SGEAR[spino] determines the defined gear stage in the main run. During search the actual gear stage may differ from the defined gear stage, because during search the gear stages are not changed. Using $VC_SGEAR[spino] and $AC_SGEAR[spino] it can be checked whether a gear stage change is to be performed after a search. The following values are possible: 1: 1st gear stage active .... 5: 5th gear stage active				
Return value	1: 1st gear stage active .... 5: 5th gear stage active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_WORD	0	0	5
			Read access	

acSGear (C, SSP2)		$AC_SGEAR[spino]		
Active spindle gear stage Variable $VC_SGEAR[spino] determines the currently active spindle gear stage. $AC_SGEAR[spino] determines the defined gear stage in the main run. During search the actual gear stage may differ from the defined gear stage, because during search the gear stages are not changed. Using $VC_SGEAR[spino] and $AC_SGEAR[spino] it can be checked whether a gear stage change is to be performed after a search. The following values are possible: 1: 1st gear stage active .... 5: 5th gear stage active				
Return value	1: 1st gear stage active .... 5: 5th gear stage active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_WORD	0	0	5
			Read access	

3.2.3.32 SSYNAC: Synchronous actions

Several synchronous actions (M, H, S, E, F, T, D) can be active simultaneously in one channel. The module SSYNAC contains a list of all the synchronous actions programmed in the current block. This module consists of arrays of varying length because some types of synchronous actions might be programmed several times in a block. A synchronous action that is not assigned produces a negative number for the respecitive index.

For each synchronous action there is an address variable and a variable in which the value of the address is entered.

5 M functions
3 S functions
3 H functions
1 T function
1 D function
6 F functions
1 E function

can be programmed in each part program block, but no more than 10 synchronous actions must be programmed in a single block.

Dadr (C, SSYNAC)				
D-number D-number. There is only one active D-number per channel.				
Units and value ranges				
Physical unit:	Data type:			

Dadr (C, SSYNAC)				
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

Dval (C, SSYNAC)				
Value of the current D-number				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

Eadr (C, SSYNAC)				S5
Number of active E-function				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

Eval (C, SSYNAC)				S5
Value of the E-function				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

Hadr (C, SSYNAC)				S5
Number of active auxiliary functions(H-functions) Number of active auxiliary functions (H-functions). Up to three H-functions can be active simultaneously.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	99

Hadr (C, SSYNAC)				S5
Addressing				
Line index:			Max. line index:	
			3	
Validity:	From NCK version 3.6		Read access	

Hval (C, SSYNAC)				S5
Value of the H-function				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DOUBLE		-99999,9999	99999,9999
Addressing				
Line index:			Max. line index:	
			3	
Validity:	From NCK version 3.6		Read access	

Madr (C, SSYNAC)				S5
Number of the active M-function Number of the active M-function. Up to 5 M-functions can be active simultaneously.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	99
Addressing				
Line index:			Max. line index:	
			5	
Validity:	From NCK version 3.6		Read access	

Mval (C, SSYNAC)				S5
Value of the M-function				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	99999999
Addressing				
Line index:			Max. line index:	
			5	
Validity:	From NCK version 3.6		Read access	

Sadr (C, SSYNAC)				S5
Number of active S-functions Number of active S-functions. Up to three S-functions can be active simultaneously.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	6

Sadr (C, SSYNAC)				S5
Addressing				
Line index:			Max. line index:	
			3	
Validity:	From NCK version 3.6		Read access	

Sval (C, SSYNAC)				S5
Value of the S-function. Value of the S-function. Specifies the spindle speed.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
rev/min , m/min	TYPE_DOUBLE		0	999999,999
Addressing				
Line index:			Max. line index:	
			3	
Validity:	From NCK version 3.6		Read access	

TPreSelAdr (C, SSYNAC)				
Number of the preselected T-function				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

TPreSelVal (C, SSYNAC)				
Value of the preselected T-function				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

Tadr (C, SSYNAC)				
Active T-number Active T-number. Only one T-number can be active at any a time.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

Tadr (C, SSYNAC)				
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

Tval (C, SSYNAC)				
T-function value				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

3.2.3.33 SYNACT: Channel-specific synchronous actions

This module contains information on the synchronous actions. The 1000 digit of the cell contains the user protection level (0-7) needed for displaying the corresponding synchronous action.

blockNoStrAct (C, SYNACT)				
For active tech.cycle: block no. of current action If a technology cycle is active: block number of the current action				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			7 * 1000 + $$/C/SYNACT/numSynAct	
			Read access	

blockNoStrProg (C, SYNACT)				
Block no. where the syn. act. is programmed Number of the block where the synchronous action has been programmed				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			7 * 1000 + $$/C/SYNACT/numSynAct	
			Read access	

id (C, SYNACT)				
ID of the synchronous action ID of the synchronous action; value 0 means that there is no ID defined (block-by-block)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			7 * 1000 + $$/C/SYNACT/numSynAct	
			Read access	

numSynAct (C, SYNACT)				
Number of synchronous actions				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			7 * 1000 + 1	
			Read access	

typStatus (C, SYNACT)				
Type and state of the synchronous action Type and status of the synchronized action Bit 0-7 describe the status: Bit 0: Active, i.e. condition fulfilled, action is being executed Bit 1: Lock, i.e. action is locked by PLC or Synact Bit 2: Lock nc, i.e. locked by another Synact Bit 3: Lock plc, i.e. locked by PLC Bit 4: Fire, i.e. condition is fulfilled Bit 5: Check condition, i.e. condition is checked Bit 6: Waiting, i.e. action is waiting to be executed Bit 7: Done, i.e. synchronized action has been completed Bit 8-15 describe the type: Bit 8: Static Bit 9: Modal Bit 10: Group-by-group (recognized by id=0)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

typStatus (C, SYNACT)				
			7 * 1000 + $$/C/SYNACT/numSynAct	
			Read access	

synactBlock (C, SYNACT)				
Current synchronized action block (short)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

synactBlockL (C, SYNACT)				
Current synchronized action block (long)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

progPathName (C, SYNACT)				
Synchronized action file				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

progLineOffset (C, SYNACT)				
SYNACT offset SYNACT offset within the $$progPathName file				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

numElem (C, SYNACT)				
Number of occupied SYNACT elements				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

synActInfo (C, SYNACT)				
Classification of the SYNACT Information on classification of the SYNACT				
Return value	Bit0: Area: User Bit1: Area: Manufacturer Bit2: Area: System Bit3: Area: Safety Bit8: Type: Static Bit9: Type: Modal			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

selectIndex (C, SYNACT)				
Consistent backup of a line The HMI writes the corresponding ID of the synchronized action into lines 8000 or 10000. The variables are then read from this synchronized action and can be viewed via line 8000 in the case of a modal/static synchronized action or line 10000.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

synActCounter (C, SYNACT)				
Modification counter of the SYNACTs Modification counter for the SYNACT entries in the relevant list.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

selectMask (C, SYNACT)				
Masking of the SYNACTs Masks some of the entries in the relevant SYNACT list. Only those SYNACTS are entered in the lists for which (selectMask-lowByte AND synActInfo-lowByte) AND (selectMask-higByte AND synActInfo-highByte) apply. The default value 0xFFFF generates completely unfiltered lists.				
Return value	Bit0: Area: User Bit1: Area: Manufacturer Bit2: Area: System Bit3: Area: Safety Bit8: Type: Static Bit9: Type: Modal			

selectMask (C, SYNACT)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

numVars (C, SYNACT)				
Number of SYNACT variables				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

varName (C, SYNACT)				
Name of SYNACT variable				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

varTyp (C, SYNACT)				
Data type Data type of SYNACT variable. Coding according to ACX.				
Return value	0: BOOL (2 bytes) 3: LONG 10: DOUBLE 12: CHAR[32]			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

varValue (C, SYNACT)				
Variable value Value of SYNACT variable				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

3.2.3.34 TO: Tool offsets

Tool data of the active tool

cuttEdgeParam (C, TO)				
Cutting edge number of the offset data block Cutting edge number CE of the offset data block of the tool with T no. = "t" and the offset number D = "d". No longer exists as it was replaced by edgeData				
Return value	The maximum number of cutting edge numbers (MD18105 $MN_MM_MAX_CUTTING_EDGE_NO) must be greater than the maximum number of cutting edges per tool (MD 18106 $MN_MM_MAX_CUTTING_EDGE_PERTOOL).			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			24	
			Read access	

cuttEdgeParamMod (C, TO)				
Modified parameter of the active tool Modified parameter of the active tool cutting edge. The rotation has been included in the calculation, so the value may not correspond with the original tool data.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			24	
			Read access	

3.2.3.35 VSYN: NCK-specific user variable for synchronous actions

This module contains channel-specific user variables for synchronous actions

acFifoN (C, VSYN)		$AC_FIFOx[y] , x = FIFONo (1-10) y = ParameterNo		
FIFO variable for synchronous actions FIFO variable for synchronous actions (Note: SYNACT only) The number of columns depends on the number of FIFOs				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

acFifoN (C, VSYN)	$AC_FIFOx[y] , x = FIFONo (1-10) y = ParameterNo	
	MD $MC_MM_LEN_AC_FIFO+6	
	Read access	

acMarker (C, VSYN)				
replaced by acMarkerL replaced by $$acMarkerL)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			MD $MC_MM_NUM_AC_MARKER	
			Read access	

acMarkerL (C, VSYN)	$AC_MARKER[n]			
Counter for motion synchronous actions 32 bit Flag variable, counter for synchronous actions (Note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			MD $MC_MM_NUM_AC_MARKER	
			Read access and write access	

acParam (C, VSYN)	$AC_PARAM[x] x = ParameterNo			
Dynamic parameters for motion-synchronous actions Dynamic parameters for motion-synchronous actions (Note: only with SYNACT)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			MD $MC_MM_NUM_AC_PARAM	
			Read access and write access	

acSystemMarkerL (C, VSYN)				
Counter for motion-synch. actions. 32-bit (system) Flag variable, counter for motion-synchronous actions (Note: only with SYNACT) Reserved for system.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			MD $MC_MM_NUM_AC_MARKER	
			Read access and write access	

acSystemParam (C, VSYN)				
Dyn. parameters for motion-synch. actions (system) Dynamic parameters for motion-synchronous actions (Note: only with SYNACT) Reserved for system.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			MD $MC_MM_NUM_AC_PARAM	
			Read access and write access	

3.2.3.36 WAL: Working area limitation

This block contains the working area limitation data.

waCSCoordSys (C, WAL)		$P_WORKAREA_CS_COORD_SYSTEM		
Coordinate system for working area limitation				
Return value	Identifier for the coordinate system in which the working area limitation is to apply. The following are valid: 0: Working area limitation in the WCS 3: Working area limitation in the SZS Special feature of line addressing: Any channel axis index can be selected as the channel axis index. Within a working area limitation group, the values are identical.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3

waCSCoordSys (C, WAL)	$P_WORKAREA_CS_COORD_SYSTEM	
Addressing		
Line index:	Max. line index:	
	$$numMachAxes * $MC_MM_NUM_WORKAREA_CS_GROUPS	
	Read access and write access	

waCSPlusEnable (C, WAL)		$P_WORKAREA_CS_PLUS_ENABLE		
TRUE: The limitation $$waCSLimitPlus is valid. Coord.-specific working area limitation, positive valid				
Return value	TRUE: The limitation of $$waCSLimitPlus is valid.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $MC_MM_NUM_WORKAREA_CS_GROUPS	
			Read access and write access	

waCSMinusEnable (C, WAL)		$P_WORKAREA_CS_MINUS_ENABLE		
TRUE: The limitation $$waCSLimitMinus is valid. Coord.-specific working area limitation, negative valid				
Return value	TRUE: The limitation of $$waCSLimitMinus is valid.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $MC_MM_NUM_WORKAREA_CS_GROUPS	
			Read access and write access	

waCSLimitPlus (C, WAL)		$P_WORKAREA_CS_LIMIT_PLUS		
Coord.-specific working area limitation, positive Position of the coordinate system-specific working area limitation in the positive direction for the addressed axis and working area group.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $MC_MM_NUM_WORKAREA_CS_GROUPS	
			Read access and write access	

waCSLimitMinus (C, WAL)		$P_WORKAREA_CS_LIMIT_MINUS		
Coord.-specific working area limitation, negative Position of the coordinate system-specific working area limitation in the negative direction for the addressed axis and working area group.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes * $MC_MM_NUM_WORKAREA_CS_GROUPS	
			Read access and write access	

3.2.3.37 Y: System data

The machine tool builder or user configures the control with the help of the machine data. Configuration can only be performed with certain access rights. The configuration of the NC can be read in the system data regardless of current access rights.

channelName (C, Y)		MD 20000: CHAN_NAME		K1
Channel name				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Validity:	From NCK version 3.6		Read access	

maskToolManagement (C, Y)	$MC_TOOL_MANAGEMENT_MASK	

Channel-specific setting for tool management

Channel-specific settings for NCK tool management function

Activation of TM memory by "0" means: The set tool management data do not use any memory space.

Value=0: TM deactivated

Bit 0=1: TM active: The tool management functions are enabled for the current channel.

Bit 1=1: TM monitoring function active: Functions required to monitor tools (tool life and number of workpieces) are enabled.

Bit 2=1: OEM functions active: The memory for user data can be utilized.

Bit 3=1: Consideration of adjacent location active

Bits 0 to 3 must be set identically to machine data MM_TOOL_MANAGEMENT_MASK (18080).

Bit 4=1: The PLC has the possibility of issuing another request for tool change preparation with modified parameters.

Part program is halted in response to T selection or M06 until it has been acknowledged by the PLC program.

Bit 5=1: The main run/PLC synchronization in response to a tool change for the main spindle is executed simultaneously with the transport acknowledgment.

Bit 6=1: The main run/PLC synchronization in response to a tool change for the auxiliary spindle is executed simultaneously with the transport acknowledgment.

Bit 7=1: The main run/PLC synchronization in response to a tool change for the main spindle is not executed until the PLC acknowledgment confirms that the tool change is complete.

Bit 8=1: The main run/PLC synchronization in response to a tool change for the auxiliary spindle is not executed until the PLC acknowledgment confirms that the tool change is complete..

Bit 9: Reserved

Bit 10=1: M06 is delayed until the preparation acknowledgment has been output by the PLC. The change signal (e.g. M06) is not output until the tool selection (DBX [n+0].2) has been acknowledged. The part program is halted in response to M06 until the T selection has been acknowledged.

Bit 11=1: The preparation command is output even if a preparation command has already been output for the same tool. This setting is useful, for example, if the chain is to be positioned when "Tx" is first called and if the second call is to initiate a check as to whether the tool is in the correct location for a tool change (e.g. in front of tool-change station).

Bit 12=1: The preparation command is executed even if the tool is already loaded in the spindle, i.e. the T selection signal (DB72.DBXn.2) is set even if it has already been set for the same tool. (Tx...Tx)

Bit 13=1: Only on systems with sufficient memory space (NCU572, NCU573): Recording of tool sequences in a diagnostics buffer. The commands are fetched from the diagnostics buffer in response to Reset and stored in a file in the passive file system, NCATR xx.MPF under part program. The trace file is useful for the Hotline in the event of errors and is not described in detail here.

Bit 14=1: Automatic tool change in response to Reset and Start according to machine data MD20120 TOOL_RESET_NAME MD20110 RESET_MODE_MASK MD20124 TOOL_MANAGEMENT_TOOLHOLDER. If machine data RESET_MODE_MASK is in use, then this bit must be set as well. If RESET_MODE_MASK is set such that the tool stored in TOOL_RESET_NAME must be loaded in response to RESET, then the select and change command is output to the user interface (DB 72) in response to RESET or Start. If machine data RESET_MODE_MASK is set such that the active tool must remain active after M30 or RESET and if the active tool is disabled in the spindle (by user), then a change command for a replacement tool is output to the user interface in response to RESET. If no replacement tool is available, then an error message is output.

Bit 15=1: No return transport of tool when several preparation commands are output. (Tx->Tx)

Bit 16=1: T location number is active

Bit 17=1: Tool life decrementation can be started/stopped via the PLC.

Units and value ranges

Physical unit:	Data type:	Initial value		

maskToolManagement (C, Y)		$MC_TOOL_MANAGEMENT_MASK		
-	TYPE_DWORD	0		
			Read access	

mmcCmd (C, Y)				
Command from NCK to HMI Command from NCK to HMI The string is made up of the following characters: 1st Character acknowledgment mode: "N" no acknowledgment "S" synchronous acknowledgment "A" asynchronous acknowledgment 2. - 6th character: five-digit sequence number in ASCII that is generated by the NCK 7. - 207th character: Command string which ends with "\0"				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Validity:	From NCK version 3.6		Read access	

mmcCmdPrep (C, Y)				
Command from NCK-preparation to the HMI Command from the NCK-preparation task to the HMI (e.g. for calling external subprograms)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

mmcCmdQuit (C, Y)				
Acknowl. from HMI for command from NCK to HMI acknowledgment from HMI for command from NCK to HMI The string is made up of the following characters: 1st Character acknowledgment code: "P" programmed "B" busy "F" failed "E" executed 2. - 6th character: five-digit sequence number in ASCII for acknowledgment code "B", "F" or "E", generated by NCK 7. - 201th character: additional communication-specific information for acknowledgment code "B", "F" or "E", ends with "\0"				
Units and value ranges				
Physical unit:	Data type:			

mmcCmdQuit (C, Y)				
-	TYPE_STRING			
Validity:	From NCK version 3.6		Write access	

mmcCmdQuitPrep (C, Y)				
Ackn. by HMI for NCK-preparation command Acknowledgemnt by HMI for an NCK-preparation command to the HMI (e.g. for calling external subprograms)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

numActAxes (C, Y)				
Number of axes in channel. Number of active tools in channel. Channel axis gaps are not included in count which means that value might be lower than numMachAxes. The following applies: $$numMachAxes >= $$numGeoAxes + $$numAuxAxes $$numActAxes = $$numGeoAxes + $$numAuxAxes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$numMachAxes
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numAuxAxes (C, Y)				
Number of auxiliary axes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numBasisFrames (C, Y)		$MC_MM_NUM_BASE_FRAMES		
Number of basic frames in channel				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	

numBasisFrames (C, Y)	$MC_MM_NUM_BASE_FRAMES	
	1	
	Read access	

numContourInProtArea (C, Y)				
Number of polygon elements / protection zone Maximum number of polygon elements per protection zone				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numGFrames (C, Y)	MD 28079: MM_NUM_G_FRAMES			
Number of G frames Number of G frames in this channel				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

numGeoAxes (C, Y)				
Number of geometry axes Number of geometry axes and orientation axes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numMachAxes (C, Y)				
Number of highest channel axis. No. of highest channel axis. This also corresponds to the number of axes in the channel provided there are no gaps in the axis sequence.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	1	
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

numOriAxes (C, Y)				
Number of orientation axes in channel				

numOriAxes (C, Y)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numProtArea (C, Y)		MD 28200: MM_NUM_PRO-TECT_AREA_CHAN		S7
Maximum number of protection zones				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numRParams (C, Y)		MD 28050: MM_NUM_R_PARAM		S7
Number of channel-specific R variables				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numSpindles (C, Y)				
Number of spindles				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numSpindlesLog (C, Y)				
Number of log. spindles Number of logical spindles. Specifies the number of lines in module SSP2.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numToolEdges (C, Y)		MD 18100: MM_NUM_CUT-TING_EDGES_IN_TOA		S7
Number of cutting edges Number of tool edges in this channel				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numUserFrames (C, Y)		MD 28080: MM_NUM_USER_FRAMES		S7
Number of user frames Number of user frames in this channel				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

oemProtText (C, Y)				
OEM text for logging buffer OEM text to be entered next in the logging buffer.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

progProtText (C, Y)				
Text for logging buffer Programmable text to be entered next in the logging buffer				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

punchNibActivation (C, Y)		MD 26012: PUNCHNIB_ACTIVATION		N4
Activation of punching and nibbling functions				
Return value	0 = option not available 1 = option available			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

systemFrameMask (C, Y)		$MC_MM_SYSTEM_FRAME_MASK		
Screenform for system frames Configuring screenform for channel-specific system frames Indicates in bit-coded form which system frames are available				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

toNo (C, Y)		MD 28085: MM_LINK_TOA_UNIT		W1
Number of T area Number of T area that is assigned to the channel				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

toolDataChangeBufferSize (C, Y)		$MC_MM_TOOL_DATA_CHANGE_BUF-FER_SIZE		
Size of the tool data modification memory Size of the effective ring buffer for the tool data modifications in the OPI block TDC (0x56). This value is the maximum column number in the OPI block TDC. If a number of channels works with a TO unit, the setting with the lowest channel number applies. The value = 0 is returned if the the ring buffer is not active ($MN_TOOL_DATA_CHANGE_COUNTER, Bit2=0 and Bit3=0).				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	

toolDataChangeBufferSize (C, Y)	$MC_MM_TOOL_DATA_CHANGE_BUF-FER_SIZE	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

stringIsMeaning (C, Y)				
Meaning of the string As soon as PI _N_STRGIS is executed, the result from interpretation of the transferred string is stored as a code in this variable. For example, the code for $P_TOOL is value 207. Also refer to NC command STRINGIS.				
Return value	000 = itemName string is unknown in the NCK 100 = itemName string is a language construct, but is not programmable (option/function is not active) 2xx = itemName string is a permissible language construct (option/function is active) 2xx = is defined by: 200 = no interpretation possible 201 = DIN address / NC address (e.g. MEAS) 202 = G code (e.g. G04, INVCW) 203 = NC language function (= command with return value, parameter Passing) (e.g. GETM-DACT) 204 = NC language procedure (= command without return value, with parameter Passing) (e.g. SBLOF) 205 = NC key word (e.g. DEFINE) 206 = machine/setting/option data (= parameter starting with $M / $S / $O) 207 = NC system parameter (= parameter starting with R and $) 208 = cycle name (name created by cycle) 209 = GUD variable (name created by GUD definitions 210 = macro name (name created by macro definition file) 211 = LUD variable (name created by active program) 212 = no Siemens G code, but an ISO G code 400 = NC address, which is not xx=01 or xx=10, and which is also not: G or R (e.g. T, D, F, H, L, M)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	4000
			Read access	

stringIsFileId (C, Y)		
Numerical file ID for the string As soon as PI _N_STRGIS is executed, the result from interpretation of the transferred string is stored as file ID in this variable. moduleId may be used instead of fileId. Also refer to NC command STRINGIS.		

stringIsFileId (C, Y)			
Return value	Meaning	Name of definition file in NCK (explanation)	OPI area (domain name)	OPI modules
	0	String is unknown		
	1	GCODE	(G code - Siemens and/or ISO)	
	2	NCADDRES	(NCK NC address character)	
	3	NCADDRES_CHAN	(channel NC address character)	
	4	NCNAM	(named NCK NC address)	
	5	NCNAM_CHAN	(named channel NC address)	
	6	FRAME	(frame variable)	
	7	TOOLCORR	(tool parameter)	TO (=4)
	8	MACHDAT_NCK	(machine data)	NCK (=0)	M (= 0x1A)
	9	MACHDAT_CHAN	(machine data)	CHAN (=2)	M (=-0x1A)
	10	MACHDAT_AXIS	(machine data)	AXIS (=3)	M (=-0x1A)
	11	R_PARAM	(R variable)	CHAN (=2)	RP (= 0x15)
	12	AC_MARKER	(synchronized action marker)	
	13	AC_PARAM	(synchronized action parameter)	
	14	PRED_FUNC	(NC language function)	
	15	SYSDAT_NCK	(status variable)	
	16	SYSDAT_CHAN	(status variable)	
	17	SYSDAT_AXIS	(status variable)	
	18	USER_NCK	_N_SGUD_DEF	NCK (=0)	GD1 (= 0x36)
	19	USER_CHAN	_N_SGUD_DEF	CHAN (=2)	GD1 (= 0x36)
	20	USER_AXIS	_N_SGUD_DEF	AXIS (=3)	GD1 (= 0x36)
	21	USERMACRO	_N_SMAC_DEF _N_MMAC_DEF _N_UMAC_DEF	
	22	EEC	(leadscrew parameter)	
	23	QEC	(quadrant error parameter)	
	24	CEC	(cross error compensation parameter)	
	25	TOOLMAGAZINE	(magazine parameter)	TO (=4)
	26	PROTAREA	(protection area parameter)	
	27	PROTAREA_CHAN	(protection area parameter)	
	28	USER_NCK2	_N_MGUD_DEF	NCK (=0)	GD2 (= 0x2D)
	29	USER_NCK3	_N_UGUD_DEF	NCK (=0)	GD3 (= 0x2E)
	30	USER_NCK4	_N_GUD4_DEF	NCK (=0)	GD4 (= 0x2F)
	31	USER_NCK5	_N_GUD5_DEF	NCK (=0)	GD5 (= 0x30)
	32	USER_NCK5	_N_GUD6_DEF	NCK (=0)	GD6 (= 0x31)
	33	USER_NCK5	_N_GUD7_DEF	NCK (=0)	GD7 (= 0x32)
	34	USER_NCK5	_N_GUD8_DEF	NCK (=0)	GD8 (= 0x33)
	35	USER_NCK5	_N_GUD9_DEF	NCK (=0)	GD9 (= 0x34)
	36	USER_CHAN2	_N_MGUD_DEF	CHAN (=2)	GD2 (= 0x2D)
	37	USER_CHAN3	_N_UGUD_DEF	CHAN (=2)	GD3 (= 0x2E)
	38	USER_CHAN4	_N_GUD4_DEF	CHAN (=2)	GD4 (= 0x2F)

stringIsFileId (C, Y)				
	39 USER_CHAN5	_N_GUD5_DEF	CHAN (=2)	GD5 (= 0x30)
	40 USER_CHAN6	_N_GUD6_DEF	CHAN (=2)	GD6 (= 0x31)
	41 USER_CHAN7	_N_GUD7_DEF	CHAN (=2)	GD7 (= 0x32)
	42 USER_CHAN8	_N_GUD8_DEF	CHAN (=2)	GD8 (= 0x33)
	43 USER_CHAN9	_N_GUD9_DEF	CHAN (=2)	GD9 (= 0x34)
	44 reserved			
	45 reserved			
	46 reserved			
	47 reserved			
	48 reserved			
	49 reserved			
	50 reserved			
	51 reserved			
	52 TOOLCARRIER (orientable tool carrier parameter TO (=4)			
	53 GCODESEXT (G code)	NCK (=0)	(G codes FANUC)	
	54 FRAME_NCK (frame)	NCK (=0)		
	55 CYC_PARAM_CHAN (global cycle transfer parameter)r			
	56 TOOLENVMOD (tool environment parameter)			
	57 SYNAGUD_CHAN (synchronized action capable GUD	CHAN (=2)	GD1 (= 0x36)	
	58 SYNAGUD_CHAN2 (synchronized action capable GUD	CHAN (=2)	GD2 (= 0x2D)	
	59 SYNAGUD_CHAN3 (synchronized action capable GUD	CHAN (=2)	GD3 (= 0x2E)	
	60 SYNAGUD_CHAN4 (synchronized action capable GUD	CHAN (=2)	GD4 (= 0x2F)	
	61 SYNAGUD_CHAN5 (synchronized action capable GUD	CHAN (=2)	GD5 (= 0x30)	
	62 SYNAGUD_CHAN6 (synchronized action capable GUD	CHAN (=2)	GD6 (= 0x31)	
	63 SYNAGUD_CHAN7 (synchronized action capable GUD	CHAN (=2)	GD7 (= 0x32)	
	64 SYNAGUD_CHAN8 (synchronized action capable GUD	CHAN (=2)	GD8 (= 0x33)	
	65 SYNAGUD_CHAN9 (synchronized action capable GUD	CHAN (=2)	GD9 (= 0x34)	
	66 NKIN (kinematic chain parameter)			
	67 NPA (3D protection area parameter)			
	68 WAL_CS (work area in specific coordinate system)			
	69 TOOLISO22CORR (tool correction parameter with ISO2.2)			
	70 TOOLISO32CORR (tool correction parameter with ISO3.2)			
	71 EPS_PARAM (ePS services parameters (reserved for ePS use only !!)			
	>= 200 LUD (LUD / PUD - program local variable)			
	Note: via OPI normally only a subset of the NCK data blocks is made known.			
	Note: missing column entries means that no definition file has been defined for this line, or that in OPI no domain name has been defined. Various NCK data may be accessed through OPI variable blocks instead of through domain names. Example: tool data, frame data, etc. It is possible that several OPI variable blocks exist for a value of stringIsFileId.			

Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

stringIsSymbolId (C, Y)				
Numerical symbol ID As soon as PI _N_STRGIS is executed, the result from interpretation of the transferred string is stored as symbol ID in this variable. The symbol ID is found in the NCK module, which is specified in variable $$stringIsFileId. This value can be found also in the corresponding ACC and ACX file.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

3.2.4 DriveHsa: Main spindle drive data

3.2.4.1 S: State data

During NC operation different internal states occur and system-specific data may change during operation. To distinguish those from system variables, they are classified as state data.

A distinction is made between:
- NCK-specific state data
- Mode-group-specific state data
- Channel-specific state data
- Drive-specific state data (FDD)
- Drive-specific state data (MSD)

Attention: The HS module cannot be addressed with MMC100/EBF/OP030

load611U (H, S)				
Load 611U				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Validity:	From NCK version 3.6		Read access	

3.2.5 Drive: HMI data

3.2.5.1 S: State data

Some internal status data of the HMI can be accessed via this module.

_Nck_Nck_ActApplication (M, S)				
Current application Current application for display in HMI				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access and write access	

_Nck_Nck_ActBag (M, S)				
Current operating mode HMI Current operating mode for display in HMI				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access and write access	

_Nck_Nck_Channel (M, S)				
Current channel Current channel for display in HMI				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access and write access	

_Nck_Nck_CoordSystem (M, S)				
Coordinate system Coordinate system for display in HMI				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access and write access	

3.2.6 Nck: NC data

3.2.6.1 CP: Generic coupling

The CP block contains the status data for the generic coupling.
The status of the axis couplings is structured in an NCK-specific and channel-specific area.

cpCtabExists (N, CP)				
Status of the specified curve table Not zero, if the specified curve table exists				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			Read access	

cpCtabId (N, CP)				
ID of the nth curve table in the specif. mem. type ID no. of the nth curve table in the specified memory type				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			Read access	

cpCtabIdNumLinSegDef (N, CP)				
Number of linear segments in the curve table Number of the linear segments defined for the specified curve table				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			Read access	

cpCtabIdNumPolDef (N, CP)				
Number of polynomials for the spec. curve table Number of the polynomials defined for the specified curve table				

cpCtabIdNumPolDef (N, CP)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			Read access	

cpCtabIdNumPolySegDef (N, CP)				
Number of polynomial segments in the curve table Number of the polynomial segments defined for the specified curve table				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			Read access	

cpCtabIdNumSegDef (N, CP)				
Number of segments for the specified curve table Number of segments defined for the specified curve table				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			Read access	

cpCtabLocked (N, CP)				
Locking status of the specified curve table Locking status, value > 0, if curve table is locked				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		-1	3
Addressing				
Line index:			Max. line index:	
			Read access	

cpCtabMemType (N, CP)				
Memory type of the curve table Memory type in which the curve table is stored				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		-1	2
Addressing				
Line index:			Max. line index:	
			Read access	

cpCtabNumDef (N, CP)				
Number of defined curve tables Total number of curve tables defined for the specified memory type				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

cpCtabNumFree (N, CP)				
Number of available curve tables Number of additional curve tables which can be defined in the specified memory type				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

cpCtabNumPolDef (N, CP)				
Number of defined curve table polynomials Total number of curve table polynomials defined in the specified memory type				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

cpCtabNumPolFree (N, CP)				
Number of available curve table polynomials Number of additional curve table polynomials which can be defined in the specified memory type				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

cpCtabNumPolMax (N, CP)				
Max. number of permissible curve table polynomials Maximum number of curve table polynomials permissible in the specified memory type				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

cpCtabNumSegDef (N, CP)				
Number of defined curve table segments Total number of curve table segments of the specified segment type defined in the specified memory type				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			23	
			Read access	

cpCtabNumSegFree (N, CP)				
Number of available curve table segments Number of additional curve table segments of the specified segment type which can be defined in the specified memory type				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	

cpCtabNumSegFree (N, CP)				
			23	
			Read access	

cpCtabNumSegMax (N, CP)				
Maximum number of permissible curve table segments Maximum number of curve table segments of the specified segment type permissible in the specified memory type				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			23	
			Read access	

cpCtabPeriodic (N, CP)				
Periodicity of the specified curve table Periodicity, value > 0, if curve table is periodic				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		-1	2
Addressing				
Line index:			Max. line index:	
			Read access	

3.2.6.2 DIAG: Diagnostics data, available only for internal development purposes

This block contains NCK-specific diagnostic data.

ipoClockNr (N, DIAG)				
Number of the IPO cycle Number of the IPO cycle since system ramp-up. This value may only be read out on the target in the IPO time level (via logging function).				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

ipoRemainTime (N, DIAG)				
Remaining IPO time Returns the remaining IPO time in ms. This value may only be read out on the target in the IPO time level (via logging function).				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
ms	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckAliveAndWellAddr (N, DIAG)				
Address of $$nckAliveAndWell Memory address of $$nckAliveAndWell				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckAliveAndWellAddrPtr (N, DIAG)				
References to $$nckAliveAndWell Memory address of a variable that contains the address of $$nckAliveAndWell				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

protocTimeMeasurement (N, DIAG)				
Parameterization of the time measurement Parameterization of the logging time measurement				
Return value	O: Delivery version 1: Test version			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

protocTimeMeasurement (N, DIAG)				
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

createFile (N, DIAG)				
Serves for creating internal NCK files				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0
Addressing				
Line index:			Max. line index:	
			5	
			Read access	

bootStatus (N, DIAG)				
Boot status				
Return value	= 0: No active boot Bit0: actTime < rebootTime, that is boot has been initiated Bit1: actTime < rebootTimeout, that is no timeout yet Bit2: File system is currently inconsistent - boot delayed Bit3: Backup on CF card is active - boot delayed Bit5: Only with 840D-powerline: FFS is currently inconsistent - boot delayed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

exTime (N, DIAG)				
System time in seconds since power up System time in seconds since power up The time is advanced in the IPO cycle by software. This time is used to control the variable service cycle and the delayed boot, among other things.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	

exTime (N, DIAG)				
			1	
			Read access	

asyncFlushDelayTime (N, DIAG)				
Flush delay time Number of seconds delay of the asynchronous flush processes in the file system. Negative values result in a synchronous instead of an asynchronous flush.				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access and write access	

systemTime (N, DIAG)				
System time				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access and write access	

lastTimeSeqSi (N, DIAG)				
Time measurement of the last variable access Time measurement of the last read/write variable via OPI in ms				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

maxPduLength (N, DIAG)				
Length of a PDU				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

eesParameter (N, DIAG)				
Parameterization of EES				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			4	
			Read access and write access	

eesSimulateFileAccessProblem (N, DIAG)				
Simulate file access problem with EES				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

plcVersCount (N, DIAG)				
Number of PLC components Number of PLC components returned when calling BSVdiPlcVersionData::getPlcVersCount().				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

btrace (N, DIAG)				
Text for BTrace Text to be accepted into the BTrace.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access and write access	

createMaxPfsPath (N, DIAG)				
Generate a maximum path in the PFS A path is being generated with maximum length in the PFS				

createMaxPfsPath (N, DIAG)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckFreezeMode (N, DIAG)				
Freeze mode Information about whether the NCK is in freeze mode state. Value 0 : NCK runs normally. Value 1 : NCK has stopped (this means program execution and communication with PLC have stopped). Only OPI communication is possible (for diagnostics).				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dsfPathIsRelativeCfCard (N, DIAG)				
DSF path mode DSF path mode: 0 : HOSTF_DSF_FILE==3; the path of the DSF archives can lie anywhere; NCROOTDIR is not taken into account 1 : HOSTF_DSF_FILE==0; the path of the DSF archives is relative to the CF card; NCROOTDIR is not taken into account by the NCK				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

3.2.6.3 DIAGN: Diagnostic module

This module contains information about global NC diagnostic data.

Net times: Time without interrupts by higher priority time levels.

Gross times: Time with interrupts by higher priority time levels.

Time levels in order of their priority: position controller, interpolator, block preparation.

actCycleTimeBrut (N, DIAGN)				
Sum of current gross runtime Total of the current gross run times of all channels in ms.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
ms	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

actCycleTimeNet (N, DIAGN)				
Sum of current net runtime Total of the current net run times of all channels in ms.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
ms	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

nckCapabilities (N, DIAGN)				
Alternative name:	compressAbility			
Compressed data transfer Describes the NCK functionality				
Return value	Bit0=1: With Huffman algorithm compressed files can be transferred (this corresponds to instruction ";$COMPR=HUFFMAN1" during download) Bit1=1: The protocol of optimized upload is supported			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dp611USpecAccChangeCnt (N, DIAGN)				
Modification counter for ACC information The counter is incremented if the NCK changes the available ACC information				
Units and value ranges				
Physical unit:	Data type:	Initial value		

dp611USpecAccChangeCnt (N, DIAGN)				
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dp611USpecAccKey (N, DIAGN)				
Info about available ACC contents Version and type information about available ACC contents				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumDrives	
			Read access	

dp611USpecAccMask (N, DIAGN)				
Screenform: ACC files available Bit-coded screenform indicating the drives for which special ACC files are available				
Return value	Bit 0 == 1 -> A special ACC is available for drive with log. drive number 1.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dp611USpecAccPath (N, DIAGN)				
Path for ACC files in the NCK file system Path in which the ACC files are stored in the NCK file system. This path might be empty later on if the files are to be supplied from the active file system. Current equivalent value: /_N_VS_DIR				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	

dp611USpecAccPath (N, DIAGN)				
			1	
			Read access	

dpAxisCfgMachAxisNr (N, DIAGN)				
Machine axis !!CAUTION NCU LINK!!				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	INT32_MAX
Addressing				
Line index:			Max. line index:	
			$$dpAxisCfgNumAxes	
			Read access	

dpAxisCfgNumAxes (N, DIAGN)				
Number of axis entries Number of axes entered in the system				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	INT32_MAX
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpAxisCfgValid (N, DIAGN)				
Axis info is available				
Return value	0=Information is not available 1=Information is available			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpAxisStateCtrlout (N, DIAGN)				
Status of output drivers.				
Return value	0=no axis status assigned 1=axis status assigned 2=axis status is cyclical 3=axis status assigned and cyclical			

dpAxisStateCtrlout (N, DIAGN)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			$$dpAxisCfgNumAxes	
			Read access	

dpAxisStateEnc1 (N, DIAGN)				
Status encoder 1 driver				
Return value	0=no axis status assigned 1=axis status assigned 2=axis status is cyclical 3=axis status assigned and cyclical			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpAxisCfgNumAxes	
			Read access	

dpAxisStateEnc2 (N, DIAGN)				
Status encoder 2 driver				
Return value	0=no axis status assigned 1=axis status assigned 2=axis status is cyclical 3=axis status assigned and cyclical			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpAxisCfgNumAxes	
			Read access	

dpAxisStateLifeCntErrCtrlout (N, DIAGN)				
Sign-of-life error counter output This data counts the number of position control cycles since failure of the sign-of-life signal				
Return value	0 to n= number of position control cycles since failure of the sign-of-life signal			

dpAxisStateLifeCntErrCtrlout (N, DIAGN)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	INT32_MAX
Addressing				
Line index:			Max. line index:	
			$$dpAxisCfgNumAxes	
			Read access	

dpAxisStateLifeCntErrEnc1 (N, DIAGN)				
Sign-of-life error counter encoder 1 This data counts the number of position control cycles since failure of the sign-of-life signal				
Return value	0 to n= number of position control cycles since failure of the sign-of-life signal			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpAxisCfgNumAxes	
			Read access	

dpAxisStateLifeCntErrEnc2 (N, DIAGN)				
Sign-of-life error counter encoder 2 This data counts the number of position control cycles since failure of the sign-of-life signal				
Return value	0 to n= number of position control cycles since failure of the sign-of-life signal			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpAxisCfgNumAxes	
			Read access	

dpBusCfgBaudrate (N, DIAGN)				
Baud rate on DP bus (bit/s)				
Return value	The permissible baud rates are determined by the Profibus standard (DIN19245 EN50170)			
Units and value ranges				
Physical unit:	Data type:	Initial value		
Hz	TYPE_DOUBLE	0		

dpBusCfgBaudrate (N, DIAGN)				
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusCfgBusNo (N, DIAGN)				
Bus number of the bus Bus number of the bus; used for conversion of "Bus index"=1...$$dpBusCfgNumBuses to "Bus number"				
Return value	All permissible bus numbers are possible: 1 = 1.DP bus on the PLC 2 = 2.DP / MPI bus on the PLC 3 = Virtual PROFIBUS 4 = Isochronous real-time Ethernet (reserved)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	4
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpBusCfgCycleTime (N, DIAGN)				
DP cycle time in [s,s,userdef] The time required by the master to scan all slaves once (request, response), until the cycle starts from the beginning again.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
s, user defined	TYPE_DOUBLE	0	0	DOUBLE_MAX
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusCfgDataExTime (N, DIAGN)				
Data exchange time in [s,s,userdef]				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
s, user defined	TYPE_DOUBLE	0	0	DOUBLE_MAX
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusCfgNumBuses (N, DIAGN)				
Number of DP buses				
Return value	Currently only one bus standardized acc. to Profibus DP standard			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpBusCfgValid (N, DIAGN)				
Bus configuration data are available				
Return value	TRUE= data exist and are initialized FALSE= no data exist			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpBusStateAccessDurationAct (N, DIAGN)				
Current access time to bus Current access time to communications buffer for DP master				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateAccessDurationMax (N, DIAGN)				
Maximum access time to bus Maximum access time to communications buffer for DP master				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	

dpBusStateAccessDurationMax (N, DIAGN)				
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateAccessDurationMin (N, DIAGN)				
Minimum access time to bus Minimum access time to communications buffer for DP master				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateAccessErrCnt1 (N, DIAGN)				
Number of bus access errors type 1 Number of bus access errors of type 1 since NCK Start				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateAccessErrCnt2 (N, DIAGN)				
Number of bus access errors type 2 Number of bus access errors of type 2 since NCK Start				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateAvgCycleBetweenErr1 (N, DIAGN)				
Cycles between bus access errors type 1 Average number of cycles between two bus access errors of type 1				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		

dpBusStateAvgCycleBetweenErr1 (N, DIAGN)				
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateAvgCycleBetweenErr2 (N, DIAGN)				
Cycles between bus access errors type 2 Average number of cycles between two bus access errors of type 2				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateCycleCnt (N, DIAGN)				
Number of bus cycles Number of bus cycles since NCK Start				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateDpmAction (N, DIAGN)				
Indic. for operating progr. of DP M Indicator for operating progress of DP M				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateDpmActual (N, DIAGN)				
Curr. stat. of DP M bus - DP M contr. Current status of DP M bus - controlled by DP M				

dpBusStateDpmActual (N, DIAGN)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateDpmCtrl (N, DIAGN)				
Boot. stat. proc. f. DP Master Booting status of processor for DP Master dpcadmin				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateDpmError (N, DIAGN)				
Error on status transitions				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateDpmPrjCnt (N, DIAGN)	
Modification counter for new CP configs. Modification counter for new DP configurations. Suggested use: *) Read modification counter (1) *) Read out configuring data *) Read modification counter (2) *) If the modification counters in (1) and (2) are identical and both display "valid", the data read from HW-Config will be consistent.	
Return value	even values -> configuration invalid uneven values -> configuration valid

dpBusStateDpmPrjCnt (N, DIAGN)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateDpmRequest (N, DIAGN)				
Des. status of DP M bus - HOST req. Desired status of DP M bus - request from HOST				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpBusStateNumActiveSlaves (N, DIAGN)				
Number of slaves connected to bus This data indicates how many slaves can currently be accessed via the bus. This value is updated in online operation.				
Return value	The number of slaves on the bus is determined by the Profibus standard (DIN19245 EN50170)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	125
Addressing				
Line index:			Max. line index:	
			$$dpBusCfgNumBuses	
			Read access	

dpClientCfgId (N, DIAGN)				
Identification client NCK/PLC/3RD				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpClientCfgNumClnt	
			Read access	

dpClientCfgNumClnt (N, DIAGN)				
Number of clients				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	INT32_MAX
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpClientCfgValid (N, DIAGN)				
Client information is available				
Return value	0=no client information available 1=client information is available			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpClientStateComm (N, DIAGN)				
Client status incl. output release				
Return value	0=None output enable 1=Client state output enable			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpClientCfgNumClnt	
			Read access	

dpSlaveCfgAssignBus (N, DIAGN)				
Bus number of the slave				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveCfgBusAddr (N, DIAGN)				
Broadcast bus address of slave The address of the slave on the bus. In addition to its own address, every slave has a broadcast address via which all salves can be addressed.				
Return value	The broadcast address is not available for individually addressing a single slave. 127: Broadcast address			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	127
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveCfgDataExchangeTime (N, DIAGN)				
Tdx [s,s,userdef] Time for the end of cyclical data transfer				
Return value	See $$dpSlaveCfgMasterAppCycTime			
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveCfgInputTime (N, DIAGN)				
Ti [s,s,userdef] Time for actual-value sensing				
Return value	See $$dpSlaveCfgMasterAppCycTime			
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveCfgIsochronModeSupport (N, DIAGN)				
Isochronous mode configured for slave Gives information whether the slave has been configured for isochronous mode on the PROFIBUS.				
Return value	0: Isochronous mode not configured 1: Isochronous mode configured			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveCfgMasterAppCycTime (N, DIAGN)				
T_mapc [s,s,userdef] Position controller cycle. For a detailed description, please refer to PROFIDRIVE PROFIL ANTRIEBSTECHNIK (Edition: V1.2 Draft, April 1999) Section 7				
Return value	See PROFIDRIVE PROFIL ANTRIEBSTECHNIK (Edition: V1.2 Draft, April 1999) Section 7			
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveCfgNumSlaves (N, DIAGN)				
Number of configured slaves Number of slaves configured in SDB1xxx. This value may not match the actual number of slaves connected to the bus.				
Return value	The number of slaves which can be configured for bus connection is determined by Profibus standard (DIN19245 EN50170).			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	125
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpSlaveCfgProfibusCycleTime (N, DIAGN)				
Tdp [s,s,userdef] Bus cycle time				
Return value	See $$dpSlaveCfgMasterAppCycTime			
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveCfgOutputTime (N, DIAGN)				
To [s,s,userdef] Time for setpoint acceptance				
Return value	See $$dpSlaveCfgMasterAppCycTime			
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveCfgValid (N, DIAGN)				
Slave data are available This data indicates whether the slave data structure has already been initialized. The structure is initialized when a slave configuration or status data is accessed. Scanning dpSlaveCfgValid also activates initialization of the structure.				
Return value	True: Slave data are available False: Slave data are not available			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpSlaveIdentNo (N, DIAGN)				
Ident number of the slave				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveIdentNoEx (N, DIAGN)				
Extended ID no. of the slave The extended ID no. of the PROFIBUS slave helps to identify the PROFIBUS slaves not officially classified as such and therefore lack specificiation $$dpSlaveIdentNo.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveStateComm (N, DIAGN)				
Slave active on bus or not (LED green) The slave is active on the bus once the drive assigned to the slave has successfully logged on to the bus.				
Return value	True: Slave on bus False: Slave not on bus			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveStateIncCnt (N, DIAGN)				
Incarnation counter The incarnation counter of the slave is increased by one each time the slave is included in the bus. If the slave drops out of the bus, this counter is not changed. After the first time it has gone into the bus (that is the first operational status of the slave), the value is 1. In case of an area overflow, the count restarts at 0. This only functions with slaves which contain at least one assigned NC axis. In the case of other slaves (pure I/O slaves, or axes controlled by the PLC), this values remains at 0.				
Return value	From 0 (starting value after Restart) to a maximum of 2147483647 (2^31-1).			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	2147483647
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveStateSync (N, DIAGN)				
DpSync The drive linked to this slave is operating in cyclic mode. Slaves without a drive are defined as "non-cyclical".				
Return value	True: Cyclical False: Non-cyclical			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$dpSlaveCfgNumSlaves	
			Read access	

dpSlaveVendorId (N, DIAGN)				
Manufacturer's number of the device PROFIBUS: Always returns 0 PROFINET: Manufacturer's number of the device				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		

dpSlaveVendorId (N, DIAGN)		
Addressing		
Line index:	Max. line index:	
	$$dpSlaveCfgNumSlaves	
	Read access	

dpSlotCfgAssignAxis (N, DIAGN)				
Assignment [axis] This data supplies the axis indices of the drive, encoder 1 and encoder 2 for access in the Axis-Assign-Table.				
Return value	The 32-bit value consists of 4 bytes with the following meaning: Byte0(bits 0-7) = axis index of axis Byte1(bits 8-15) = axis index, encoder 1 Byte2(bits 16-23)= axis index, encoder 2 Byte3(bits 24-31)= provided for future extensions. A byte with the value 0xFF indicates that no axis index is defined for the relevant slot.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	255	0	32
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgAssignBus (N, DIAGN)				
Assignment [bus] Bus number assigned to this slot				
Return value	Since only one bus is currently supported by Profibus DP, there is only one bus to which all slots are assigned.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgAssignClient (N, DIAGN)		
Assignment [client] This data supplies the clientIndex for accessing the Client Assign table.		
Return value	0=no assignment possible (this applies to diagnostic and PKW slots) >0 assignment exists	

dpSlotCfgAssignClient (N, DIAGN)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgAssignMaster (N, DIAGN)				
Assignment [master] Number of master to which this slot is assigned				
Return value	Since only one bus is currently supported by Profibus DP and only one Class 1 Master exists per bus, there is only one master to which all slots are assigned.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgAssignSlave (N, DIAGN)				
Assignment [slave] This data contains the bus address of the slave belonging to the nth slot.				
Return value	All legal slave addresses can be specified			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	125
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgIoType (N, DIAGN)				
I/O identifier				
Return value	0 = input slot 1 = output slot 2 = diagnosis slot			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

dpSlotCfgIoType (N, DIAGN)				
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgLength (N, DIAGN)				
Length in number of bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	32
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgLogBaseAddress (N, DIAGN)				
Log. basic address of slot The logical basic address of the slot is assigned during configuration. Although it is not needed on the bus for data transfer purposes, this address is the only means by which a unique link can be created between the NCK and bus nodes.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	UINT16_MAX
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgNumSlots (N, DIAGN)				
Number of slots The total number of all slots configured in the system is stored in this data.				
Return value	0 (lower limit) up to INT32_MAX(upper limit); Note that a slave cannot support more than 256 slots.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	INT32_MAX
Addressing				
Line index:			Max. line index:	

dpSlotCfgNumSlots (N, DIAGN)				
			1	
			Read access	

dpSlotCfgPNSlotNr (N, DIAGN)				
PROFINET: Slot number within the IO device PROFIBUS: Not used PROFINET: Slot number within the IO device				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	255
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgSlaveAddress (N, DIAGN)				
Bus address of slave This data contains the bus address of the slave to which this slot is assigned. Several slots may have the same slave address.				
Return value	The number of available addresses on the bus is determined by the Profibus standard (DIN19245 EN50170).			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	125
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgSlotNr (N, DIAGN)				
(Sub-)slot number within the slave PROFIBUS: Slot number within the slave PROFINET: Subslot number within the IO device				
Return value	A maximum total of 256 slots can be assigned to each slave. 0: Diagnostic slot 2: Diagnostic slot 4: 1st data slot			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	255
Addressing				
Line index:			Max. line index:	

dpSlotCfgSlotNr (N, DIAGN)				
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotCfgValid (N, DIAGN)				
Slot information is available The slot data structure (CcIdent) exists and is initialized.				
Return value	True: Data are valid False: Data are invalid or not initialized			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpSlotStateComm (N, DIAGN)				
Status of slots Status of slots (ok, failed, not processed by the NCK)				
Return value	0= no sign of life 1= sign of life 2= not processed by NCK			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotStateRecvTelegram (N, DIAGN)				
Received message frame Bit pattern of this slot received by the master in the form of a hexadecimal string				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotStateSendTelegram (N, DIAGN)				
Transmitted message frame Bit pattern of this slot sent to the slave in the form of a hexadecimal string				
Return value	Transmitted message frame			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSlotStateTelegramType (N, DIAGN)				
Message frame type Message frame type of slot				
Return value	0 = Message frame type unknown			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	UINT16_MAX
Addressing				
Line index:			Max. line index:	
			$$dpSlotCfgNumSlots	
			Read access	

dpSysCfgAvailable (N, DIAGN)				
DPA/DPM available This data specifies whether the system has been generated with DP Adapter and/or DP Master				
Return value	0= Neither DPA nor DPM available 1= DPA available 2= DPM available 3= DPA and DPM available			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpSysCfgNumMaster (N, DIAGN)				
Number of masters				
Return value	There is only one master per bus with DP. Since only 1 bus is currently permitted by the bus standard, there can only be a maximum of one master.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpSysCfgValid (N, DIAGN)				
DP system configuration data are valid This data indicates whether the configuration data are valid and initialized.				
Return value	TRUE or FALSE			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

dpSysCfgVersionDpm (N, DIAGN)				
Version of DP Master software Version number of DP M SW as numerical value				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$dpSysCfgNumMaster	
			Read access	

dpSysCfgVersionDpr (N, DIAGN)				
Actual version Dpr Actual version Dpr (inaccessible in earlier SW)				
Units and value ranges				
Physical unit:	Data type:	Initial value		

dpSysCfgVersionDpr (N, DIAGN)				
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$dpSysCfgNumMaster	
			Read access	

dpSysCfgVersionDprEx (N, DIAGN)				
Expected Dpr version from DPR_SS_VERSION DPR_SS_VERSION is a version number stored in the NCK which can be read out via this variable.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$dpSysCfgNumMaster	
			Read access	

dpSysCfgVersionHost (N, DIAGN)				
Host SW version This data contains the version number of the host SW as a numerical value				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	0	UINT16_MAX
Addressing				
Line index:			Max. line index:	
			$$dpSysCfgNumMaster	
			Read access	

dpSysStateDpmInit (N, DIAGN)				
Startup synchronization status There are three different initialization states: REQUEST, ACKNOWLEDGE and ERROR				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$dpSysCfgNumMaster	
			Read access	

errCodeSetNrGen (N, DIAGN)				
Error code set for communication errors Selection of error code set to be used in the case of communication errors. The selection is client-specific, the client is identified by the sender address.				
Return value	1: P1-compatible codes (default) 0-4: As 1 5: P5-compatible codes 6: Current codes (from P6) 7-100: Reserved			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

errCodeSetNrPi (N, DIAGN)				
Error code set P1 for comm. errors Selection of error code set to be used by PI Services in the case of communication errors. The selection is client-specific, the client is identified by the sender address.				
Return value	0: P1-compatible code 5: P5-compatible code 6: P6-compatible code			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

maxCycleTimeBrut (N, DIAGN)				
Sum of maximum gross runtime Total of the maximum gross run times of all channels in ms.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
ms	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	

maxCycleTimeBrut (N, DIAGN)				
			13	
			Read access	

maxCycleTimeNet (N, DIAGN)				
Sum of maximum net runtime Total of the maximum net run times of all channels in ms.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
ms	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

minCycleTimeBrut (N, DIAGN)				
Sum of minimum gross runtime Total of the minimum gross run times of all channels in ms.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
ms	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

minCycleTimeNet (N, DIAGN)				
Sum of minimum net runtimes Total of the minimum net run times of all channels in ms.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
ms	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

nckCompileSwitches (N, DIAGN)				
NCK compiler switch Selected NCK compiler switches				
Return value	Bit0 (0x1): NDEBUG Bit1 (0x2): NOTRACES Bit2 (0x4): EMBARGO Bit3 (0x8): TARGET Bit4 (0x10): HOST_SIMULATION			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

poweronTime (N, DIAGN)		$AN_POWERON_TIME		
Time since normal boot Time since last normal boot (in minutes)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

setupTime (N, DIAGN)		$AN_SETUP_TIME		
Time since default boot Time since last "control system boot on default values" (in minutes). The timer is automatically set to zero on every "control system boot on default values".				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

noOfPersistencyReq (N, DIAGN)	$AN_PERSDIAG[row-1,0]	
Number of persistence operations		

noOfPersistencyReq (N, DIAGN)		$AN_PERSDIAG[row-1,0]		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			34	
			Read access	

noOfPersistencyReqFailed (N, DIAGN)		$AN_PERSDIAG[row-1,1]		
Number of failed persistence operations				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			34	
			Read access	

totalPersistencyTime (N, DIAGN)		$AN_PERSDIAG[row-1,2]		
Summated time for persistence Summated time for making data persistent				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			34	
			Read access	

persistencyTimeMinimal (N, DIAGN)		$AN_PERSDIAG[row-1,3]		
Minimum time for persistence Minimum time for making data persistent				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			34	
			Read access	

persistencyTimeAverage (N, DIAGN)	$AN_PERSDIAG[row-1,4]	
Average time for persistence Average time for making data persistent		

persistencyTimeAverage (N, DIAGN)		$AN_PERSDIAG[row-1,4]		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			34	
			Read access	

persistencyTimeMaximal (N, DIAGN)		$AN_PERSDIAG[row-1,5]		
Maximum time for persistence Maximum time for making data persistent				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			34	
			Read access	

optUploadDiag (N, DIAGN)				
Optimized upload test parameterization				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
			Read access and write access	

maxLockTimeDomSubtask (N, DIAGN)				
Max. lock time of the subtask EXCOM Max. net lock time of the subtask EXCOM (domain services)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
ms	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

noOfPersistencyOverflowPrep (N, DIAGN)	$AN_PERSDIAG[row-1,6]	
No. overflows of preprocessing power-fail buffer Number of overflows of the power-fail buffer in the preprocessing (Value > 0 indicates that the buffer is too small -> increase $MN_MM_ACTFILESYS_LOG_FILE_MEM[0] if possible)		

noOfPersistencyOverflowPrep (N, DIAGN)		$AN_PERSDIAG[row-1,6]		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

noOfPersistencyOverflowToolChange (N, DIAGN)		$AN_PERSDIAG[row-1,7]		
No. overflows of the tool change power-fail buffer Number of overflows of the power-fail buffer for the tool change data changes (Value > 0 indicates that the buffer is too small -> increase $MN_MM_ACTFILESYS_LOG_FILE_MEM[1] if possible)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

noOfPersistencyOverflowIpo (N, DIAGN)		$AN_PERSDIAG[row-1,8]		
No. overflows of synchro. action power-fail buffer Number of overflows of the power-fail buffer for the changes to persistent data in synchronous actions (Value > 0 indicates that the buffer is too small -> increase $MN_MM_ACTFILESYS_LOG_FILE_MEM[2] if possible)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

isPersistencyOverflowPrep (N, DIAGN)		$AN_PERSDIAG[row-1,9]		
Overflow of prepr. power-fail buffer at power-fail Value=1: At the time of power-fail/power off, overflow of the power-fail buffer was pending in the preprocessing. The last data change before power-fail/power off has been lost.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	

isPersistencyOverflowPrep (N, DIAGN)	$AN_PERSDIAG[row-1,9]	
	1	
	Read access	

isPersistencyOverflowToolChange (N, DIAGN)	$AN_PERSDIAG[row-1,10]			
Overfl. of tool ch. power-fail buff. at power-fail Value=1: At the time of power-fail/power off, overflow of the power-fail buffer was pending in the preprocessing. The last tool/magazine data change before power-fail/power off has been lost.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

isPersistencyOverflowIpo (N, DIAGN)	$AN_PERSDIAG[row-1,11]			
Overflow of synchro. action power-fail buffer at p Value=1: At the time of power-fail/power off, overflow of the power-fail buffer for changes of persistent data in synchronous actions was pending. The last data change before power-fail/power off has been lost.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

noOfPersistencyEntriesPrep (N, DIAGN)	$AN_PERSDIAG[row-1,12]			
No. data entries in preprocess. power-fail buffer Number of data entries in the power-fail buffer in the preprocessing				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

noOfPersistencyEntriesToolChange (N, DIAGN)	$AN_PERSDIAG[row-1,13]	
No. data entries in tool change power-fail buffer Number of data entries in the power-fail buffer for the tool change data changes in IPO		

noOfPersistencyEntriesToolChange (N, DIAGN)		$AN_PERSDIAG[row-1,13]		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

noOfPersistencyEntriesIpo (N, DIAGN)		$AN_PERSDIAG[row-1,14]		
No. data entries in synchronous actions pf buffer Number of data entries in the power-fail buffer for changes to persistent data in synchronous actions				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aveCycleTimeNet (N, DIAGN)				
Average net run time Average net run time in ms				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

sumCycleTimeNet (N, DIAGN)				
Sum of net run times Sum of net run times in s				
Units and value ranges				
Physical unit:	Data type:			
s	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

noOfPersistencyCollisions (N, DIAGN)				
Number of collisions during previous async flush If a persistence operation (= flush) is triggered, although an asynchronous persistence operation with the same ID has not yet been executed, then the value of this variable is incremented.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actNckLoad (N, DIAGN)				
Actual utilization of the NC in %. NC load imposed by cyclic tasks (position controller, interpolator and possibly soft PLC). The load must not be too high, so that even low priority tasks, such as communication of the data for the display, can be performed. The value is based on $$actCycleTimeNet with line=CYCLE and brings this in relation to the $$taskCycleTime with line=CYCLE. Continues to take into account that only part of the cycle is actually available to the NCK, corresponds to $NCK_PCOS_TIME_RATIO.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

minNckLoad (N, DIAGN)				
Minimum utilization of the NC in % Minimum NC load imposed by cyclic tasks (position controller, interpolator and possibly soft PLC). The value is based on $$minCycleTimeNet with line=CYCLE and brings this in relation to the $$taskCycleTime with line=CYCLE. Continues to take into account that only part of the cycle is actually available to the NCK, corresponds to $NCK_PCOS_TIME_RATIO.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

maxNckLoad (N, DIAGN)				
Maximum utilization of the NC in % Maximum NC load imposed by cyclic tasks (position controller, interpolator and possibly soft PLC). The value is based on $$maxCycleTimeNet with line=CYCLE and brings this in relation to the $$taskCycleTime with line=CYCLE. Continues to take into account that only part of the cycle is actually available to the NCK, corresponds to $NCK_PCOS_TIME_RATIO.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aveNckLoad (N, DIAGN)				
Average utilization of the NC in % Average NC load imposed by cyclic tasks (position controller, interpolator and possibly soft PLC). The value is based on $$aveCycleTimeNet with line=CYCLE and brings this in relation to the $$taskCycleTime with line=CYCLE. Continues to take into account that only part of the cycle is actually available to the NCK, corresponds to $NCK_PCOS_TIME_RATIO.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

taskCycleTime (N, DIAGN)				
Cycle time Cycle time of the task in ms				
Units and value ranges				
Physical unit:	Data type:			
ms	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			12	
			Read access	

mdChangeCounter (N, DIAGN)				
Modification counter for machine data				

mdChangeCounter (N, DIAGN)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			4	
			Read access	

taskName (N, DIAGN)				
Name of task				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

taskAvailable (N, DIAGN)				
Task is present				
Return value	0: Task not present 1: Task present			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_UWORD		0	
Addressing				
Line index:			Max. line index:	
			13	
			Read access	

numCores (N, DIAGN)				
Number of cores used by the NCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

3.2.6.4 ETPD: Data list for protocolling

Data lists for protocolling.This module allows to access several lines or rows at a time.

area (N, ETPD)				
nth data in list: area Variable specification of nth OPI data in the list: area				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			2 + 5 * ($$numData- 1)	
			Read access and write access	

col (N, ETPD)				
nth data in list: col Variable specification of nth OPI data in list: col				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			4 + 5 * ($$numData- 1)	
			Read access and write access	

numData (N, ETPD)				
Number of data in the list Number of data in the list.				
Return value	<= maxnumTraceProtData			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	$$maxnumTraceProtDa-ta
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

varSpecs (N, ETPD)				
Do not use this variable any more.				

varSpecs (N, ETPD)				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	$$maxnumTraceProtDa-ta
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

row (N, ETPD)				
nth data in list: row Variable specification of nth OPI data in list: row				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			5 + 5 * ($$numData- 1)	
			Read access and write access	

type (N, ETPD)				
nth data in list: type Low byte:Variable specification of nth OPI data in list: type (module type) High byte: Should be read more than a line so that the number of lines can be given here.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			6 + 5 * ($$numData- 1)	
			Read access and write access	

unit (N, ETPD)				
nth data in list: unit Variable specification of nth OPI data in list: unit				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

unit (N, ETPD)		
	3 + 5 * ($$numData- 1)	
	Read access and write access	

3.2.6.5 FA: Active Work Offset

There are the following frame indices:
2: IFRAME current settable work offset (only if $MN_MM_NUM_GLOBAL_USER_FRAMES > 0)
6: ACTBFRAME current total of base frames (only if $MN_MM_NUM_GLOBAL_BASE_FRAMES = 0)

The maximum frame index is: 6

linShift (N, FA)		diverse, siehe Bausteinbescheibung		PA
Translation of an active frame Translation of an active work offset (the physical unit is defined in basicLengthUnit in module Y in area N).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			20 * $$maxnumGlobMachAxes	
			Read access	

linShiftFine (N, FA)				
Fine offset for frames Fine offset for frames, extension of the basic frames and the settable frames.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			6 * $$maxnumGlobMachAxes	
			Read access and write access	

mirrorImgActive (N, FA)		diverse, siehe Bausteinbescheibung		PA
Mirroring enabled in an active frame Mirroring enabled in an active work offset				
Return value	0 = mirroring not active 1 = mirroring active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

mirrorImgActive (N, FA)	diverse, siehe Bausteinbescheibung	PA
Addressing		
Line index:	Max. line index:	
	20 * $$maxnumGlobMachAxes	
	Read access	

rotation (N, FA)		diverse, siehe Bausteinbescheibung		PA
Rotation of an active frame Rotation of an active work offset				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			20 * $$maxnumGlobMachAxes	
			Read access	

rotationCoordinate (N, FA)		diverse, siehe Bausteinbescheibung		
Coordinate rotation of active frames Rotation around a coordinate of an active work offset 1: Rotation around the first non-existing geometry axis.				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			20 * $$maxnumGlobMachAxes	
			Read access	

scaleFact (N, FA)		diverse, siehe Bausteinbescheibung		PA
Scaling factor of an active frame Scaling factor of an active work offset				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			6 * $$maxnumGlobMachAxes	
			Read access	

3.2.6.6 FB: Base Frame

This only applies if $MN_MM_NUM_GLOBAL_BASE_FRAMES > 0.

The maximum frame index is: $MN_MM_NUM_GLOBAL_BASE_FRAMES - 1

linShift (N, FB)		$P_NCBFR[x,TR] x=FrameNo, y=Axis		PA
Translation of a settable frame Translation of settable work offset (the physical unit is defined in basicLengthUnit in module Y in area N).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_BASE_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

linShiftFine (N, FB)		$P_NCBFR[x,SI] x=FrameNo, y=Axis		
Fine offset for frames Fine offset with frames, expansion of basic frames and settable frames				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_BASE_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

mirrorImgActive (N, FB)		$P_NCBFR[x ,MI] x=FrameNo, y=Axis		PA
Mirroring enabled in a settable frame Mirroring enabled in a settable work offset				
Return value	0: Mirroring not active 1: Mirroring active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_BASE_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

rotation (N, FB)		$P_NCBFR[x,y,RT] x=FrameNo, y=Axis		PA
Rotation of a settable frame Rotation of a settable work offset				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_BASE_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

rotationCoordinate (N, FB)		$P_NCBFR[x,y,RT] x=FrameNo, y=1		
Coordinate rotation of settable frames Rotation around a coordinate of a settable work offset 1: Rotation around the first non-existing geometry axis.				
Units and value ranges				
Physical unit:	Data type:			
Degree	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_BASE_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

scaleFact (N, FB)		$P_NCBFR[x,SC] x=FrameNo, y=Axis		PA
Scaling factor of a settable frame Scaling factor of a settable work offset				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_BASE_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

3.2.6.7 FG: Frames for grinding applications

These occur only if $MN_MM_NUM_GLOBAL_G_FRAMES > 0.

The following frame indices are possible:
0: GRAME1
1: GRAME2
2: GRAME3
3: GRAME4
...
n: GRAMEn
...
99: GRAME100

The maximum frame index is: $MN_MM_NUM_GLOBAL_G_FRAMES - 1

The PI service SETUFR has to be called to activate the grinding frames.

linShift (N, FG)				PA
Translation of grinding frame Translation (the physical unit is defined in basicLengthUnit in block Y in area N).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_G_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

linShiftFine (N, FG)				
Fine offset for frames Fine offset with frames, expansion of basic frames and settable frames				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_G_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

mirrorImgActive (N, FG)				PA
Mirroring of grinding frame Mirroring				
Return value	0 = mirroring not active 1 = mirroring active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_G_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

rotationCoordinate (N, FG)				
Rotations of grinding frame Dummy variable, do not use				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_G_FRAMES * $ $maxnumGlobMachAxes	
			Read access	

scaleFact (N, FG)				PA
Scaling factor of grinding frame Scaling factor				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_G_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

rotation (N, FG)				
Dummy variable, do not use				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

3.2.6.8 FU: Settable Work Offset

This only applies if $MN_MM_NUM_GLOBAL_USER_FRAMES > 0.

The following frame indices are possible:
0 = G500
1 = G54
2 = G55
3 = G56
4 = G57
5 = G505
6 = G506
:
n = G5n
:
99 = G599

The maximum frame index is: $MN_MM_NUM_GLOBAL_USER_FRAMES - 1

The PI service SETUFR has to be called in order to activate the settable frames.

linShift (N, FU)				PA
Translation of a settable frame Translation of settable work offset (the physical unit is defined in basicLengthUnit in module Y in area N).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_USER_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

linShiftFine (N, FU)				
Fine offset for frames Fine offset with frames, expansion of basic frames and settable frames				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

linShiftFine (N, FU)		
		$MN_MM_NUM_GLOBAL_USER_FRAMES * $ $maxnumGlobMachAxes
		Read access and write access

mirrorImgActive (N, FU)				PA
Mirroring enabled in a settable frame Mirroring enabled in a settable work offset				
Return value	0 = mirroring not active 1 = mirroring active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_USER_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

rotationCoordinate (N, FU)				
Coordinate rotation of a settable frame				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_USER_FRAMES * $ $maxnumGlobMachAxes	
			Read access	

scaleFact (N, FU)				PA
Scaling factor of a settable frame Scaling factor of a settable work offset				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_GLOBAL_USER_FRAMES * $ $maxnumGlobMachAxes	
			Read access and write access	

rotation (N, FU)		
Dummy variable, do not use		

rotation (N, FU)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

3.2.6.9 M: Machine data

Global machine data

ALARM_PAR_DISPLAY_TEXT (N, M)		MD 11413: $MN_ALARM_PAR_DISPLAY_TEXT		
MD 11413: $MN_ALARM_PAR_DISPLAY_TEXT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

ALARM_REACTION_CHAN_NOREADY (N, M)		MD 11412: $MN_ALARM_REACTION_CHAN_NOREADY		
MD 11412: $MN_ALARM_REACTION_CHAN_NOREADY				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

ASSIGN_CHAN_TO_MODE_GROUP (N, M)		MD 10010: $MN_ASSIGN_CHAN_TO_MODE_GROUP		
MD 10010: $MN_ASSIGN_CHAN_TO_MODE_GROUP				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

ASUP_EDITABLE (N, M)		MD 11610: $MN_ASUP_EDITABLE		
MD 11610: $MN_ASUP_EDITABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

ASUP_EDIT_PROTECTION_LEVEL (N, M)		MD 11612: $MN_ASUP_EDIT_PROTEC-TION_LEVEL		
MD 11612: $MN_ASUP_EDIT_PROTECTION_LEVEL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

ASUP_START_MASK (N, M)		MD 11602: $MN_ASUP_START_MASK		
MD 11602: $MN_ASUP_START_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

ASUP_START_PRIO_LEVEL (N, M)		MD 11604: $MN_AS-UP_START_PRIO_LEVEL		
MD 11604: $MN_ASUP_START_PRIO_LEVEL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

AUXFU_GROUP_SPEC (N, M)		MD 11110: $MN_AUXFU_GROUP_SPEC		
MD 11110: $MN_AUXFU_GROUP_SPEC				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

AUXFU_MAXNUM_GROUP_ASSIGN (N, M)		MD 11100: $MN_AUXFU_MAX-NUM_GROUP_ASSIGN		
MD 11100: $MN_AUXFU_MAXNUM_GROUP_ASSIGN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

BAG_MASK (N, M)		MD 11600: $MN_BAG_MASK		
MD 11600: $MN_BAG_MASK				

BAG_MASK (N, M)		MD 11600: $MN_BAG_MASK		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

CC_ASSIGN_FASTOUT_MASK (N, M)		MD 10420: $MN_CC_ASSIGN_FAST-OUT_MASK		
MD 10420: $MN_CC_ASSIGN_FASTOUT_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

CC_HW_DEBUG_MASK (N, M)		MD 10430: $MN_CC_HW_DEBUG_MASK		
MD 10430: $MN_CC_HW_DEBUG_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

CC_VDI_IN_DATA (N, M)		MD 10400: $MN_CC_VDI_IN_DATA		
MD 10400: $MN_CC_VDI_IN_DATA				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

CC_VDI_OUT_DATA (N, M)		MD 10410: $MN_CC_VDI_OUT_DATA		
MD 10410: $MN_CC_VDI_OUT_DATA				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

CHECK_DISABLE_TIME (N, M)		MD 10085: $MN_CHECK_DISABLE_TIME		
MD 10085: $MN_CHECK_DISABLE_TIME				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

COMPAR_ASSIGN_ANA_INPUT_1 (N, M)	MD 10530: $MN_COMPAR_AS-SIGN_ANA_INPUT_1	
MD 10530: $MN_COMPAR_ASSIGN_ANA_INPUT_1		

COMPAR_ASSIGN_ANA_INPUT_1 (N, M)		MD 10530: $MN_COMPAR_ASSIGN_ANA_INPUT_1		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

COMPAR_ASSIGN_ANA_INPUT_2 (N, M)		MD 10531: $MN_COMPAR_ASSIGN_ANA_INPUT_2		
MD 10531: $MN_COMPAR_ASSIGN_ANA_INPUT_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

COMPAR_TYPE_1 (N, M)		MD 10540: $MN_COMPAR_TYPE_1		
MD 10540: $MN_COMPAR_TYPE_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

COMPAR_TYPE_2 (N, M)		MD 10541: $MN_COMPAR_TYPE_2		
MD 10541: $MN_COMPAR_TYPE_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

COM_IPO_TIME_RATIO (N, M)		MD 10072: $MN_COM_IPO_TIME_RATIO		
MD 10072: $MN_COM_IPO_TIME_RATIO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

CTRLOUT_LEAD_TIME (N, M)	MD 10082: $MN_CTRLOUT_LEAD_TIME	
MD 10082: $MN_CTRLOUT_LEAD_TIME		

CTRLOUT_LEAD_TIME (N, M)		MD 10082: $MN_CTRLOUT_LEAD_TIME		
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
			Read access	

CTRLOUT_LEAD_TIME_MAX (N, M)		MD 10083: $MN_CTRLOUT_LEAD_TIME_MAX		
MD 10083: $MN_CTRLOUT_LEAD_TIME_MAX				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
			Read access	

DIG_ASSIGN_DIGITIZE_TO_CHAN (N, M)		MD 11430: $MN_DIG_ASSIGN_DIGITIZE_TO_CHAN		
MD 11430: $MN_DIG_ASSIGN_DIGITIZE_TO_CHAN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

DIR_VECTOR_NAME_TAB (N, M)		MD 10640: $MN_DIR_VECTOR_NAME_TAB		
MD 10640: $MN_DIR_VECTOR_NAME_TAB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

DRIVE_DIAGNOSIS (N, M)		MD 13100: $MN_DRIVE_DIAGNOSIS		
MD 13100: $MN_DRIVE_DIAGNOSIS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

DRYRUN_MASK (N, M)		MD 10704: $MN_DRYRUN_MASK		
MD 10704: $MN_DRYRUN_MASK				

DRYRUN_MASK (N, M)		MD 10704: $MN_DRYRUN_MASK		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

ENABLE_ALARM_MASK (N, M)		MD 11411: $MN_ENABLE_ALARM_MASK		
MD 11411: $MN_ENABLE_ALARM_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

ENC_HANDWHEEL_INPUT_NR (N, M)		MD 11344: $MN_ENC_HANDWHEEL_IN-PUT_NR		
MD 11344: $MN_ENC_HANDWHEEL_INPUT_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

ENC_HANDWHEEL_MODULE_NR (N, M)		MD 11342: $MN_ENC_HAND-WHEEL_MODULE_NR		
MD 11342: $MN_ENC_HANDWHEEL_MODULE_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

ENC_HANDWHEEL_SEGMENT_NR (N, M)		MD 11340: $MN_ENC_HAND-WHEEL_SEGMENT_NR		
MD 11340: $MN_ENC_HANDWHEEL_SEGMENT_NR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

ENC_SSI_BAUD_RATE (N, M)		MD 14000: $MN_ENC_SSI_BAUD_RATE		
MD 14000: $MN_ENC_SSI_BAUD_RATE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

EULER_ANGLE_NAME_TAB (N, M)		MD 10620: $MN_EULER_ANGLE_NAME_TAB		
MD 10620: $MN_EULER_ANGLE_NAME_TAB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

FASTIO_ANA_INPUT_WEIGHT (N, M)		MD 10320: $MN_FASTIO_ANA_INPUT_WEIGHT		
MD 10320: $MN_FASTIO_ANA_INPUT_WEIGHT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

FASTIO_ANA_NUM_INPUTS (N, M)		MD 10300: $MN_FASTIO_ANA_NUM_INPUTS		
MD 10300: $MN_FASTIO_ANA_NUM_INPUTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FASTIO_ANA_NUM_OUTPUTS (N, M)		MD 10310: $MN_FASTIO_ANA_NUM_OUTPUTS		
MD 10310: $MN_FASTIO_ANA_NUM_OUTPUTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FASTIO_ANA_OUTPUT_WEIGHT (N, M)		MD 10330: $MN_FASTIO_ANA_OUTPUT_WEIGHT		
MD 10330: $MN_FASTIO_ANA_OUTPUT_WEIGHT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			

FASTIO_ANA_OUTPUT_WEIGHT (N, M)	MD 10330: $MN_FASTIO_ANA_OUTPUT_WEIGHT	
Addressing		
Line index:	Max. line index:	
	2	
	Read access	

FASTIO_DIG_NUM_INPUTS (N, M)		MD 10350: $MN_FASTIO_DIG_NUM_INPUTS		
MD 10350: $MN_FASTIO_DIG_NUM_INPUTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FASTIO_DIG_NUM_OUTPUTS (N, M)		MD 10360: $MN_FASTIO_DIG_NUM_OUTPUTS		
MD 10360: $MN_FASTIO_DIG_NUM_OUTPUTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FASTIO_DIG_SHORT_CIRCUIT (N, M)		MD 10361: $MN_FASTIO_DIG_SHORT_CIRCUIT		
MD 10361: $MN_FASTIO_DIG_SHORT_CIRCUIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

FPU_CTRLWORD_INIT (N, M)		MD 18910: $MN_FPU_CTRLWORD_INIT		
MD 18910: $MN_FPU_CTRLWORD_INIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

FPU_ERROR_MODE (N, M)		MD 18900: $MN_FPU_ERROR_MODE		
MD 18900: $MN_FPU_ERROR_MODE				
Units and value ranges				
Physical unit:	Data type:			

FPU_ERROR_MODE (N, M)		MD 18900: $MN_FPU_ERROR_MODE		
-	TYPE_DWORD			
			Read access	

FPU_EXEPTION_MASK (N, M)		MD 18920: $MN_FPU_EXEPTION_MASK		
MD 18920: $MN_FPU_EXEPTION_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

FRAME_ANGLE_INPUT_MODE (N, M)		MD 10600: $MN_FRAME_ANGLE_IN-PUT_MODE		
MD 10600: $MN_FRAME_ANGLE_INPUT_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

HANDWH_IMP_PER_LATCH (N, M)		MD 11320: $MN_HANDWH_IMP_PER_LATCH		
MD 11320: $MN_HANDWH_IMP_PER_LATCH				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

HANDWH_REVERSE (N, M)		MD 11310: $MN_HANDWH_REVERSE		
MD 11310: $MN_HANDWH_REVERSE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

HANDWH_TRUE_DISTANCE (N, M)		MD 11346: $MN_HANDWH_TRUE_DIS-TANCE		
MD 11346: $MN_HANDWH_TRUE_DISTANCE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

HW_ASSIGN_ANA_FASTIN (N, M)		MD 10362: $MN_HW_ASSIGN_ANA_FASTIN		
MD 10362: $MN_HW_ASSIGN_ANA_FASTIN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

HW_ASSIGN_ANA_FASTOUT (N, M)		MD 10364: $MN_HW_ASSIGN_ANA_FASTOUT		
MD 10364: $MN_HW_ASSIGN_ANA_FASTOUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

HW_ASSIGN_DIG_FASTIN (N, M)		MD 10366: $MN_HW_ASSIGN_DIG_FASTIN		
MD 10366: $MN_HW_ASSIGN_DIG_FASTIN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

HW_ASSIGN_DIG_FASTOUT (N, M)		MD 10368: $MN_HW_ASSIGN_DIG_FASTOUT		
MD 10368: $MN_HW_ASSIGN_DIG_FASTOUT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

HW_CLOCKED_MODULE_MASK (N, M)		MD 10384: $MN_HW_CLOCKED_MODULE_MASK		
MD 10384: $MN_HW_CLOCKED_MODULE_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

HW_LEAD_TIME_FASTIO (N, M)		MD 10382: $MN_HW_LEAD_TIME_FASTIO		
MD 10382: $MN_HW_LEAD_TIME_FASTIO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

HW_UPDATE_RATE_FASTIO (N, M)		MD 10380: $MN_HW_UPDATE_RATE_FASTIO		
MD 10380: $MN_HW_UPDATE_RATE_FASTIO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

IGNORE_SINGLEBLOCK_MASK (N, M)		MD 10702: $MN_IGNORE_SINGLEBLOCK_MASK		
MD 10702: $MN_IGNORE_SINGLEBLOCK_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

INDEX_AX_LENGTH_POS_TAB_1 (N, M)	MD 10900: $MN_INDEX_AX_LENGTH_POS_TAB_1	
MD 10900: $MN_INDEX_AX_LENGTH_POS_TAB_1		

INDEX_AX_LENGTH_POS_TAB_1 (N, M)		MD 10900: $MN_INDEX_AX_LENGTH_POS_TAB_1		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

INDEX_AX_LENGTH_POS_TAB_2 (N, M)		MD 10920: $MN_INDEX_AX_LENGTH_POS_TAB_2		
MD 10920: $MN_INDEX_AX_LENGTH_POS_TAB_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

INDEX_AX_POS_TAB_1 (N, M)		MD 10910: $MN_INDEX_AX_POS_TAB_1		
MD 10910: $MN_INDEX_AX_POS_TAB_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

INDEX_AX_POS_TAB_2 (N, M)		MD 10930: $MN_INDEX_AX_POS_TAB_2		
MD 10930: $MN_INDEX_AX_POS_TAB_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

INFO_CROSSCHECK_CYCLE_TIME (N, M)		MD 10092: $MN_INFO_CROSSCHECK_CYCLE_TIME		
MD 10092: $MN_INFO_CROSSCHECK_CYCLE_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

INFO_FREE_MEM_DPR (N, M)		MD 18070: $MN_INFO_FREE_MEM_DPR		
MD 18070: $MN_INFO_FREE_MEM_DPR				

INFO_FREE_MEM_DPR (N, M)		MD 18070: $MN_INFO_FREE_MEM_DPR		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

INFO_FREE_MEM_DYNAMIC (N, M)		MD 18050: $MN_INFO_FREE_MEM_DYNAMIC		
MD 18050: $MN_INFO_FREE_MEM_DYNAMIC				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

INFO_FREE_MEM_STATIC (N, M)		MD 18060: $MN_INFO_FREE_MEM_STATIC		
MD 18060: $MN_INFO_FREE_MEM_STATIC				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

INFO_SAFETY_CYCLE_TIME (N, M)		MD 10091: $MN_INFO_SAFETY_CYCLE_TIME		
MD 10091: $MN_INFO_SAFETY_CYCLE_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

INIT_MD (N, M)		MD 11200: $MN_INIT_MD		
MD 11200: $MN_INIT_MD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

INI_FILE_MODE (N, M)		MD 11220: $MN_INI_FILE_MODE		
MD 11220: $MN_INI_FILE_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

INTCY_DELAY_TIME (N, M)		MD 14020: $MN_INTCY_DELAY_TIME		
MD 14020: $MN_INTCY_DELAY_TIME				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

INTERMEDIATE_POINT_NAME_TAB (N, M)		MD 10660: $MN_INTERMEDI-ATE_POINT_NAME_TAB		
MD 10660: $MN_INTERMEDIATE_POINT_NAME_TAB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

IPO_PARAM_NAME_TAB (N, M)		MD 10650: $MN_IPO_PAR-AM_NAME_TAB		
MD 10650: $MN_IPO_PARAM_NAME_TAB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

IPO_SYSCLOCK_TIME_RATIO (N, M)		MD 10070: $MN_IPO_SYS-CLOCK_TIME_RATIO		
MD 10070: $MN_IPO_SYSCLOCK_TIME_RATIO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

IPUCR_DELAY_TIME (N, M)		MD 14010: $MN_IPUCR_DELAY_TIME		
MD 14010: $MN_IPUCR_DELAY_TIME				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

JOG_INCR_SIZE_TAB (N, M)		MD 11330: $MN_JOG_INCR_SIZE_TAB		
MD 11330: $MN_JOG_INCR_SIZE_TAB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

JOG_INC_MODE_LEVELTRIGGRD (N, M)		MD 11300: $MN_JOG_INC_MODE_LEV-ELTRIGGRD		
MD 11300: $MN_JOG_INC_MODE_LEVELTRIGGRD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

JOG_MODE_KEYS_EDGETRIGGRD (N, M)		MD 10731: $MN_JOG_MODE_KEYS_EDGE-TRIGGRD		
MD 10731: $MN_JOG_MODE_KEYS_EDGETRIGGRD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

JOG_MODE_KEYS_LEVELTRIGGRD (N, M)		MD 10730: $MN_JOG_MODE_KEYS_LEVEL-TRIGGRD		
MD 10730: $MN_JOG_MODE_KEYS_LEVELTRIGGRD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

LEN_PROTOCOL_FILE (N, M)		MD 11420: $MN_LEN_PROTOCOL_FILE		
MD 11420: $MN_LEN_PROTOCOL_FILE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MAXNUM_USER_DATA_FLOAT (N, M)		MD 14508: $MN_MAXNUM_USER_DATA_FLOAT		
MD 14508: $MN_MAXNUM_USER_DATA_FLOAT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MAXNUM_USER_DATA_HEX (N, M)		MD 14506: $MN_MAXNUM_USER_DATA_HEX		
MD 14506: $MN_MAXNUM_USER_DATA_HEX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MAXNUM_USER_DATA_INT (N, M)		MD 14504: $MN_MAXNUM_USER_DATA_INT		
MD 14504: $MN_MAXNUM_USER_DATA_INT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MDCA_DRIVE_LOGIC_NR (N, M)		MD 13010: DRIVE_LOGIC_NR[x] x = PlugplaceNo		
Alternative name:	DRIVE_LOGIC_NR			
Logical drive number				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_CHAR		0	30
Addressing				
Line index:			Max. line index:	
			14	
Validity:	From NCK version 3.6		Read access and write access	

MDCA_DRIVE_MODULE_TYPE (N, M)		MD 13030: DRIVE_MODULE_TYPE[x] x = PlugplaceNo		
Alternative name:	DRIVE_MODULE_TYPE			
Module identifier Module identifier of relevant drive bus slot				

MDCA_DRIVE_MODULE_TYPE (N, M)		MD 13030: DRIVE_MODULE_TYPE[x] x = PlugplaceNo		
Return value	1 = single-axis module 2 = two-axis module 9 = terminal block for dig. I/Os 10 = bit bus interface			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			14	
Validity:	From NCK version 3.6		Read access and write access	

MDCA_DRIVE_TYPE (N, M)		MD 13040: DRIVE_TYPE[x] x = Plugpla-ceNo		
Alternative name:	DRIVE_TYPE			
Drive type Drive type identifier for each drive bus slot				
Return value	1 = FDD 2 = MSD			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			14	
Validity:	From NCK version 3.6		Read access and write access	

MDD_INT_INCR_PER_DEG (N, M)		MD 10210: INT_INCR_PER_DEG		
Alternative name:	INT_INCR_PER_DEG			
Calculation resolution for angular position				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DOUBLE		0,000001	1000
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDD_INT_INCR_PER_MM (N, M)		MD 10200: INT_INCR_PER_MM		
Alternative name:	INT_INCR_PER_MM			
Calculation resolution for linear positions				

MDD_INT_INCR_PER_MM (N, M)		MD 10200: INT_INCR_PER_MM		
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DOUBLE		0,000001	1000
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDD_SYSCLOCK_CYCLE_TIME (N, M)		MD 10050: SYSCLOCK_CYCLE_TIME		
Alternative name:	SYSCLOCK_CYCLE_TIME			
Basic system clock cycle Basic system clock cycle. For possible assignment of values, see description of machine data SYSCLOCK_CYCLE_TIME.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
s	TYPE_DOUBLE		0,000125 s	0,032 s
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDLA_DRIVE_INVERTER_CODE (N, M)		MD 13020: DRIVE_INVER-TER_CODE[x] x = PlugplaceNo		
Alternative name:	DRIVE_INVERTER_CODE			
Power section code of drive module				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			14	
Validity:	From NCK version 3.6		Read access and write access	

MDL_POSCTRL_SYSCLOCK_TIME_RATIO (N, M)		MD 10060: POSCTRL_SYS-CLOCK_TIME_RATIO		
Alternative name:	POSCTRL_SYSCLOCK_TIME_RATIO			
Position control cycle factor				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		1	100
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

MDSA_AXCONF_MACHAX_NAME_TAB (N, M)		MD 10000: AXCONF_MA-CHAX_NAME_TAB[x] x = Axis		
Alternative name:	AXCONF_MACHAX_NAME_TAB			
Machine axis name				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			7	
Validity:	From NCK version 3.6		Read access and write access	

MD_FILE_STYLE (N, M)		MD 11230: $MN_MD_FILE_STYLE		
MD 11230: $MN_MD_FILE_STYLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

MEAS_PROBE_LOW_ACTIVE (N, M)		MD 13200: $MN_MEAS_PROBE_LOW_ACTIVE		
MD 13200: $MN_MEAS_PROBE_LOW_ACTIVE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

MIRROR_REF_AX (N, M)		MD 10610: $MN_MIRROR_REF_AX		
MD 10610: $MN_MIRROR_REF_AX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

MMC_CMD_TIMEOUT (N, M)		MD 10132: $MN_MMC_CMD_TIMEOUT		
MD 10132: $MN_MMC_CMD_TIMEOUT				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

MM_CEC_MAX_POINTS (N, M)		MD 18342: $MN_MM_CEC_MAX_POINTS		
MD 18342: $MN_MM_CEC_MAX_POINTS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

MM_CHAN_HASH_TABLE_SIZE (N, M)		MD 18250: $MN_MM_CHAN_HASH_TA-BLE_SIZE		
MD 18250: $MN_MM_CHAN_HASH_TABLE_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_CHAR_LENGTH_OF_BLOCK (N, M)		MD 18330: $MN_MM_CHAR_LENGTH_OF_BLOCK		
MD 18330: $MN_MM_CHAR_LENGTH_OF_BLOCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_COM_TASK_STACK_SIZE (N, M)		MD 18502: $MN_MM_COM_TASK_STACK_SIZE		
MD 18502: $MN_MM_COM_TASK_STACK_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_DIR_HASH_TABLE_SIZE (N, M)		MD 18300: $MN_MM_DIR_HASH_TA-BLE_SIZE		
MD 18300: $MN_MM_DIR_HASH_TABLE_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_DRIVE_TASK_STACK_SIZE (N, M)		MD 18520: $MN_MM_DRIVE_TASK_STACK_SIZE		
MD 18520: $MN_MM_DRIVE_TASK_STACK_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_EXTCOM_TASK_STACK_SIZE (N, M)		MD 18500: $MN_MM_EXTCOM_TASK_STACK_SIZE		
MD 18500: $MN_MM_EXTCOM_TASK_STACK_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_EXT_PROG_BUFFER_SIZE (N, M)		MD 18360: $MN_MM_EXT_PROG_BUFFER_SIZE		
MD 18360: $MN_MM_EXT_PROG_BUFFER_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_EXT_PROG_NUM (N, M)		MD 18362: $MN_MM_EXT_PROG_NUM		
MD 18362: $MN_MM_EXT_PROG_NUM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

MM_FILE_HASH_TABLE_SIZE (N, M)		MD 18290: $MN_MM_FILE_HASH_TABLE_SIZE		
MD 18290: $MN_MM_FILE_HASH_TABLE_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_FRAME_FINE_TRANS (N, M)		MD 18600: $MN_MM_FRAME_FINE_TRANS		
MD 18600: $MN_MM_FRAME_FINE_TRANS				
Units and value ranges				
Physical unit:	Data type:			

MM_FRAME_FINE_TRANS (N, M)		MD 18600: $MN_MM_FRAME_FINE_TRANS		
-	TYPE_DWORD			
			Read access	

MM_GUD_VALUES_MEM (N, M)		MD 18150: $MN_MM_GUD_VALUES_MEM		
MD 18150: $MN_MM_GUD_VALUES_MEM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_LUD_HASH_TABLE_SIZE (N, M)		MD 18240: $MN_MM_LUD_HASH_TABLE_SIZE		
MD 18240: $MN_MM_LUD_HASH_TABLE_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_MAX_SIZE_OF_LUD_VALUE (N, M)		MD 18242: $MN_MM_MAX_SIZE_OF_LUD_VALUE		
MD 18242: $MN_MM_MAX_SIZE_OF_LUD_VALUE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NCK_HASH_TABLE_SIZE (N, M)		MD 18260: $MN_MM_NCK_HASH_TABLE_SIZE		
MD 18260: $MN_MM_NCK_HASH_TABLE_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_CC_MAGAZINE_PARAM (N, M)		MD 18090: $MN_MM_NUM_CC_MAGAZINE_PARAM		
MD 18090: $MN_MM_NUM_CC_MAGAZINE_PARAM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_CC_MAGLOC_PARAM (N, M)		MD 18092: $MN_MM_NUM_CC_MA-GLOC_PARAM		
MD 18092: $MN_MM_NUM_CC_MAGLOC_PARAM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_CC_MON_PARAM (N, M)		MD 18098: $MN_MM_NUM_CC_MON_PARAM		
MD 18098: $MN_MM_NUM_CC_MON_PARAM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_CC_TDA_PARAM (N, M)		MD 18094: $MN_MM_NUM_CC_TDA_PARAM		
MD 18094: $MN_MM_NUM_CC_TDA_PARAM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_CC_TOA_PARAM (N, M)		MD 18096: $MN_MM_NUM_CC_TOA_PARAM		
MD 18096: $MN_MM_NUM_CC_TOA_PARAM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_CURVE_POLYNOMS (N, M)		MD 18404: $MN_MM_NUM_CURVE_POLYNOMS		
MD 18404: $MN_MM_NUM_CURVE_POLYNOMS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_CURVE_SEGMENTS (N, M)		MD 18402: $MN_MM_NUM_CURVE_SEGMENTS		
MD 18402: $MN_MM_NUM_CURVE_SEGMENTS				
Units and value ranges				
Physical unit:	Data type:			

MM_NUM_CURVE_SEGMENTS (N, M)		MD 18402: $MN_MM_NUM_CURVE_SEGMENTS		
-	TYPE_DWORD			
			Read access	

MM_NUM_CURVE_TABS (N, M)		MD 18400: $MN_MM_NUM_CURVE_TABS		
MD 18400: $MN_MM_NUM_CURVE_TABS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_CUTTING_EDGES_IN_TOA (N, M)		MD 18100: $MN_MM_NUM_CUTTING_EDGES_IN_TOA		
MD 18100: $MN_MM_NUM_CUTTING_EDGES_IN_TOA				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_DIR_IN_FILESYSTEM (N, M)		MD 18310: $MN_MM_NUM_DIR_IN_FILESYSTEM		
MD 18310: $MN_MM_NUM_DIR_IN_FILESYSTEM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_FILES_IN_FILESYSTEM (N, M)		MD 18320: $MN_MM_NUM_FILES_IN_FILESYSTEM		
MD 18320: $MN_MM_NUM_FILES_IN_FILESYSTEM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_FILES_PER_DIR (N, M)		MD 18280: $MN_MM_NUM_FILES_PER_DIR		
MD 18280: $MN_MM_NUM_FILES_PER_DIR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_GUD_MODULES (N, M)		MD 18118: $MN_MM_NUM_GUD_MODULES		
MD 18118: $MN_MM_NUM_GUD_MODULES				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_GUD_NAMES_AXIS (N, M)		MD 18140: $MN_MM_NUM_GUD_NAMES_AXIS		
MD 18140: $MN_MM_NUM_GUD_NAMES_AXIS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_GUD_NAMES_CHAN (N, M)		MD 18130: $MN_MM_NUM_GUD_NAMES_CHAN		
MD 18130: $MN_MM_NUM_GUD_NAMES_CHAN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_GUD_NAMES_NCK (N, M)		MD 18120: $MN_MM_NUM_GUD_NAMES_NCK		
MD 18120: $MN_MM_NUM_GUD_NAMES_NCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_MAGAZINE (N, M)		MD 18084: $MN_MM_NUM_MAGAZINE		
MD 18084: $MN_MM_NUM_MAGAZINE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_MAGAZINE_LOCATION (N, M)		MD 18086: $MN_MM_NUM_MAGAZINE_LOCATION		
MD 18086: $MN_MM_NUM_MAGAZINE_LOCATION				
Units and value ranges				
Physical unit:	Data type:			

MM_NUM_MAGAZINE_LOCATION (N, M)		MD 18086: $MN_MM_NUM_MAGA-ZINE_LOCATION		
-	TYPE_DWORD			
			Read access	

MM_NUM_MAX_FUNC_NAMES (N, M)		MD 18170: $MN_MM_NUM_MAX_FUNC_NAMES		
MD 18170: $MN_MM_NUM_MAX_FUNC_NAMES				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_MAX_FUNC_PARAM (N, M)		MD 18180: $MN_MM_NUM_MAX_FUNC_PARAM		
MD 18180: $MN_MM_NUM_MAX_FUNC_PARAM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_MMC_UNITS (N, M)		MD 10134: $MN_MM_NUM_MMC_UNITS		
MD 10134: $MN_MM_NUM_MMC_UNITS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_PROTECT_AREA_NCK (N, M)		MD 18190: $MN_MM_NUM_PRO-TECT_AREA_NCK		
MD 18190: $MN_MM_NUM_PROTECT_AREA_NCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_SUBDIR_PER_DIR (N, M)		MD 18270: $MN_MM_NUM_SUB-DIR_PER_DIR		
MD 18270: $MN_MM_NUM_SUBDIR_PER_DIR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_TOOL (N, M)		MD 18082: $MN_MM_NUM_TOOL		
MD 18082: $MN_MM_NUM_TOOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_TOOL_CARRIER (N, M)		MD 18088: $MN_MM_NUM_TOOL_CARRIER		
MD 18088: $MN_MM_NUM_TOOL_CARRIER				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_NUM_USER_MACROS (N, M)		MD 18160: $MN_MM_NUM_USER_MACROS		
MD 18160: $MN_MM_NUM_USER_MACROS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_SERVO_TASK_STACK_SIZE (N, M)		MD 18510: $MN_MM_SERVO_TASK_STACK_SIZE		
MD 18510: $MN_MM_SERVO_TASK_STACK_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_TOOL_MANAGEMENT_MASK (N, M)		MD 18080: $MN_MM_TOOL_MANAGEMENT_MASK		
MD 18080: $MN_MM_TOOL_MANAGEMENT_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_TYPE_OF_CUTTING_EDGE (N, M)		MD 18102: $MN_MM_TYPE_OF_CUTTING_EDGE		
MD 18102: $MN_MM_TYPE_OF_CUTTING_EDGE				
Units and value ranges				
Physical unit:	Data type:			

MM_TYPE_OF_CUTTING_EDGE (N, M)		MD 18102: $MN_MM_TYPE_OF_CUTTING_EDGE		
-	TYPE_DWORD			
			Read access	

MM_USER_FILE_MEM_MINIMUM (N, M)		MD 18350: $MN_MM_USER_FILE_MEM_MINIMUM		
MD 18350: $MN_MM_USER_FILE_MEM_MINIMUM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_USER_MEM_BUFFERED (N, M)		MD 18230: $MN_MM_USER_MEM_BUFFERED		
MD 18230: $MN_MM_USER_MEM_BUFFERED				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_USER_MEM_DPR (N, M)		MD 18220: $MN_MM_USER_MEM_DPR		
MD 18220: $MN_MM_USER_MEM_DPR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MM_USER_MEM_DYNAMIC (N, M)		MD 18210: $MN_MM_USER_MEM_DYNAMIC		
MD 18210: $MN_MM_USER_MEM_DYNAMIC				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MONITOR_ADDRESS (N, M)		MD 11380: $MN_MONITOR_ADDRESS		
MD 11380: $MN_MONITOR_ADDRESS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MONITOR_DISPLAY_INT (N, M)		MD 11382: $MN_MONITOR_DISPLAY_INT		
MD 11382: $MN_MONITOR_DISPLAY_INT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MONITOR_DISPLAY_REAL (N, M)		MD 11384: $MN_MONITOR_DISPLAY_REAL		
MD 11384: $MN_MONITOR_DISPLAY_REAL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

MONITOR_INPUT_INT (N, M)		MD 11386: $MN_MONITOR_INPUT_INT		
MD 11386: $MN_MONITOR_INPUT_INT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

MONITOR_INPUT_REAL (N, M)		MD 11388: $MN_MONITOR_INPUT_REAL		
MD 11388: $MN_MONITOR_INPUT_REAL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

MONITOR_INPUT_STROBE (N, M)		MD 11390: $MN_MONITOR_INPUT_STROBE		
MD 11390: $MN_MONITOR_INPUT_STROBE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

NC_USER_CODE_CONF_NAME_TAB (N, M)		MD 10712: $MN_NC_USER_CODE_CONF_NAME_TAB		
MD 10712: $MN_NC_USER_CODE_CONF_NAME_TAB				
Units and value ranges				
Physical unit:	Data type:			

NC_USER_CODE_CONF_NAME_TAB (N, M)		MD 10712: $MN_NC_USER_CODE_CONF_NAME_TAB		
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

NORMAL_VECTOR_NAME_TAB (N, M)		MD 10630: $MN_NORMAL_VEC-TOR_NAME_TAB		
MD 10630: $MN_NORMAL_VECTOR_NAME_TAB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

OPERATING_MODE_DEFAULT (N, M)		MD 10720: $MN_OPERAT-ING_MODE_DEFAULT		
MD 10720: $MN_OPERATING_MODE_DEFAULT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

OSCILL_MODE_MASK (N, M)		MD 11460: $MN_OSCILL_MODE_MASK		
MD 11460: $MN_OSCILL_MODE_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

OVR_AX_IS_GRAY_CODE (N, M)		MD 12000: $MN_OVR_AX_IS_GRAY_CODE		
MD 12000: $MN_OVR_AX_IS_GRAY_CODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

OVR_FACTOR_AX_SPEED (N, M)		MD 12010: $MN_OVR_FACTOR_AX_SPEED		
MD 12010: $MN_OVR_FACTOR_AX_SPEED				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

OVR_FACTOR_FEEDRATE (N, M)		MD 12030: $MN_OVR_FACTOR_FEEDRATE		
MD 12030: $MN_OVR_FACTOR_FEEDRATE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

OVR_FACTOR_LIMIT_BIN (N, M)		MD 12100: $MN_OVR_FACTOR_LIMIT_BIN		
MD 12100: $MN_OVR_FACTOR_LIMIT_BIN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

OVR_FACTOR_RAPID_TRA (N, M)		MD 12050: $MN_OVR_FACTOR_RAPID_TRA		
MD 12050: $MN_OVR_FACTOR_RAPID_TRA				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

OVR_FACTOR_SPIND_SPEED (N, M)	MD 12070: $MN_OVR_FACTOR_SPIND_SPEED	
MD 12070: $MN_OVR_FACTOR_SPIND_SPEED		

OVR_FACTOR_SPIND_SPEED (N, M)		MD 12070: $MN_OVR_FACTOR_SPIND_SPEED		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

OVR_FEED_IS_GRAY_CODE (N, M)		MD 12020: $MN_OVR_FEED_IS_GRAY_CODE		
MD 12020: $MN_OVR_FEED_IS_GRAY_CODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

OVR_RAPID_IS_GRAY_CODE (N, M)		MD 12040: $MN_OVR_RAPID_IS_GRAY_CODE		
MD 12040: $MN_OVR_RAPID_IS_GRAY_CODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

OVR_REFERENCE_IS_MIN_FEED (N, M)		MD 12082: $MN_OVR_REFERENCE_IS_MIN_FEED		
MD 12082: $MN_OVR_REFERENCE_IS_MIN_FEED				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

OVR_REFERENCE_IS_PROG_FEED (N, M)		MD 12080: $MN_OVR_REFERENCE_IS_PROG_FEED		
MD 12080: $MN_OVR_REFERENCE_IS_PROG_FEED				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

OVR_SPIND_IS_GRAY_CODE (N, M)		MD 12060: $MN_OVR_SPIND_IS_GRAY_CODE		
MD 12060: $MN_OVR_SPIND_IS_GRAY_CODE				

OVR_SPIND_IS_GRAY_CODE (N, M)		MD 12060: $MN_OVR_SPIND_IS_GRAY_CODE		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

PERMANENT_FEED (N, M)		MD 12202: $MN_PERMANENT_FEED		
MD 12202: $MN_PERMANENT_FEED				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

PERMANENT_ROT_AX_FEED (N, M)		MD 12204: $MN_PERMANENT_ROT_AX_FEED		
MD 12204: $MN_PERMANENT_ROT_AX_FEED				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

PLC_CYCLE_TIME_AVERAGE (N, M)		MD 10110: $MN_PLC_CYCLE_TIME_AVERAGE		
MD 10110: $MN_PLC_CYCLE_TIME_AVERAGE				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

PLC_CYCLIC_TIMEOUT (N, M)		MD 10100: $MN_PLC_CYCLIC_TIMEOUT		
MD 10100: $MN_PLC_CYCLIC_TIMEOUT				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

PLC_RUNNINGUP_TIMEOUT (N, M)		MD 10120: $MN_PLC_RUNNINGUP_TIMEOUT		
MD 10120: $MN_PLC_RUNNINGUP_TIMEOUT				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

PREPROCESSING_LEVEL (N, M)		MD 10700: $MN_PREPROCESSING_LEVEL		
MD 10700: $MN_PREPROCESSING_LEVEL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

PREP_COM_TASK_CYCLE_RATIO (N, M)		MD 10160: $MN_PREP_COM_TASK_CYCLE_RATIO		
MD 10160: $MN_PREP_COM_TASK_CYCLE_RATIO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

PREP_DRIVE_TASK_CYCLE_RATIO (N, M)		MD 10150: $MN_PREP_DRIVE_TASK_CYCLE_RATIO		
MD 10150: $MN_PREP_DRIVE_TASK_CYCLE_RATIO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

PREVENT_SYNACT_LOCK (N, M)		MD 11500: $MN_PREVENT_SYNACT_LOCK		
MD 11500: $MN_PREVENT_SYNACT_LOCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

PROG_SD_RESET_SAVE_TAB (N, M)		MD 10710: $MN_PROG_SD_RESET_SAVE_TAB		
MD 10710: $MN_PROG_SD_RESET_SAVE_TAB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

RUN_OVERRIDE_0 (N, M)		MD 12200: $MN_RUN_OVERRIDE_0		
MD 12200: $MN_RUN_OVERRIDE_0				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

SAFETY_SYSCLOCK_TIME_RATIO (N, M)		MD 10090: $MN_SAFETY_SYSCLOCK_TIME_RATIO		
MD 10090: $MN_SAFETY_SYSCLOCK_TIME_RATIO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SCALING_FACTORS_USER_DEF (N, M)		MD 10230: $MN_SCALING_FACTORS_USER_DEF		
MD 10230: $MN_SCALING_FACTORS_USER_DEF				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SCALING_SYSTEM_IS_METRIC (N, M)	MD 10240: $MN_SCALING_SYSTEM_IS_METRIC	

Define basic system for length specifications

Specifies the basic system used by the control for scaling length-dependent physical variables for data input/output.

For access from the interpreter (part program and download), from the operator panel (variable service) or through external communication, scaling takes place in the following units:

MD10240 $MN_SCALING_SYSTEM_IS_METRIC = 1: scaled in:

mm, mm/min, m/s2 , m/s3, mm/rev.

MD10240 $MN_SCALING_SYSTEM_IS_METRIC = 0: scaled in:

inch, inch/min, inch/s2, inch/s3, inch/rev.

The selection of the basic system also defines the interpretation of the programmed F value

for linear axes:

metric inch

G94 mm/min inch/min

G95 mm/rev. inch/rev.

If this machine data is changed, a run-up is required because otherwise the associated

machine data that have physical units would be scaled incorrectly.

Proceed as follows:

o MD changed manually:

Run-up and then enter the associated machine data with physical

units.

o MD changed via machine data file:

Run-up and then reload the machine data file so that the new

physical units are taken into account.

If the machine data are changed, alarm 4070 "Scaling machine data altered"

is output.

Application example(s):

Setup in the metric system and then change to the inch system.

Units and value ranges

Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

SCALING_USER_DEF_MASK (N, M)	MD 10220: $MN_SCALING_USER_DEF_MASK	

MD 10220: $MN_SCALING_USER_DEF_MASK

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SCALING_VALUE_INCH (N, M)	MD 10250: $MN_SCALING_VALUE_INCH	

MD 10250: $MN_SCALING_VALUE_INCH

SCALING_VALUE_INCH (N, M)		MD 10250: $MN_SCALING_VALUE_INCH		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

SEARCH_RUN_MODE (N, M)		MD 11450: $MN_SEARCH_RUN_MODE		
MD 11450: $MN_SEARCH_RUN_MODE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

SUPPRESS_ALARM_MASK (N, M)		MD 11410: $MN_SUP-PRESS_ALARM_MASK		
MD 11410: $MN_SUPPRESS_ALARM_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SW_CAM_ASSIGN_FASTOUT_1 (N, M)		MD 10470: $MN_SW_CAM_AS-SIGN_FASTOUT_1		
MD 10470: $MN_SW_CAM_ASSIGN_FASTOUT_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SW_CAM_ASSIGN_FASTOUT_2 (N, M)		MD 10471: $MN_SW_CAM_AS-SIGN_FASTOUT_2		
MD 10471: $MN_SW_CAM_ASSIGN_FASTOUT_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SW_CAM_ASSIGN_FASTOUT_3 (N, M)		MD 10472: $MN_SW_CAM_AS-SIGN_FASTOUT_3		
MD 10472: $MN_SW_CAM_ASSIGN_FASTOUT_3				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SW_CAM_ASSIGN_FASTOUT_4 (N, M)		MD 10473: $MN_SW_CAM_ASSIGN_FASTOUT_4		
MD 10473: $MN_SW_CAM_ASSIGN_FASTOUT_4				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SW_CAM_ASSIGN_TAB (N, M)		MD 10450: $MN_SW_CAM_ASSIGN_TAB		
MD 10450: $MN_SW_CAM_ASSIGN_TAB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_MINUS_LEAD_TIME (N, M)		MD 10460: $MN_SW_CAM_MINUS_LEAD_TIME		
MD 10460: $MN_SW_CAM_MINUS_LEAD_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_PLUS_LEAD_TIME (N, M)		MD 10461: $MN_SW_CAM_PLUS_LEAD_TIME		
MD 10461: $MN_SW_CAM_PLUS_LEAD_TIME				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_TIMER_FASTOUT_MASK (N, M)	MD 10480: $MN_SW_CAM_TIMER_FASTOUT_MASK	
MD 10480: $MN_SW_CAM_TIMER_FASTOUT_MASK		

SW_CAM_TIMER_FASTOUT_MASK (N, M)		MD 10480: $MN_SW_CAM_TIMER_FASTOUT_MASK		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

SYSCLOCK_SAMPL_TIME_RATIO (N, M)		MD 10080: $MN_SYSCLOCK_SAMPL_TIME_RATIO		
MD 10080: $MN_SYSCLOCK_SAMPL_TIME_RATIO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

TIME_LIMIT_NETTO_COM_TASK (N, M)		MD 10130: $MN_TIME_LIMIT_NETTO_COM_TASK		
MD 10130: $MN_TIME_LIMIT_NETTO_COM_TASK				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

TIME_LIMIT_NETTO_DRIVE_TASK (N, M)		MD 10140: $MN_TIME_LIMIT_NETTO_DRIVE_TASK		
MD 10140: $MN_TIME_LIMIT_NETTO_DRIVE_TASK				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
			Read access	

TOOL_CHANGE_TIME (N, M)		MD 10190: $MN_TOOL_CHANGE_TIME		
MD 10190: $MN_TOOL_CHANGE_TIME				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

TRACE_SELECT (N, M)		MD 11400: $MN_TRACE_SELECT		
MD 11400: $MN_TRACE_SELECT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

UPLOAD_MD_CHANGES_ONLY (N, M)		MD 11210: $MN_UP-LOAD_MD_CHANGES_ONLY		
MD 11210: $MN_UPLOAD_MD_CHANGES_ONLY				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

USER_DATA_FLOAT (N, M)		MD 14514: $MN_USER_DATA_FLOAT		
MD 14514: $MN_USER_DATA_FLOAT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

USER_DATA_HEX (N, M)		MD 14512: $MN_USER_DATA_HEX		
MD 14512: $MN_USER_DATA_HEX				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

USER_DATA_INT (N, M)		MD 14510: $MN_USER_DATA_INT		
MD 14510: $MN_USER_DATA_INT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

USER_DATA_PLC_ALARM (N, M)		MD 14516: $MN_USER_DA-TA_PLC_ALARM		
MD 14516: $MN_USER_DATA_PLC_ALARM				
Units and value ranges				
Physical unit:	Data type:			

USER_DATA_PLC_ALARM (N, M)		MD 14516: $MN_USER_DATA_PLC_ALARM		
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

VDI_UPDATE_IN_ONE_IPO_CYCLE (N, M)		MD 18000: $MN_VDI_UPDATE_IN_ONE_IPO_CYCLE		
MD 18000: $MN_VDI_UPDATE_IN_ONE_IPO_CYCLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

VERSION_INFO (N, M)		MD 18040: $MN_VERSION_INFO		
MD 18040: $MN_VERSION_INFO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

DRIVE_LOGIC_ADDRESS (N, M)		MD 13050: $MN_DRIVE_LOGIC_ADDRESS		
MD 13050: $MN_DRIVE_LOGIC_ADDRESS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

PROFIBUS_ALARM_ACCESS (N, M)		MD 13140: $MN_PROFIBUS_ALARM_ACCESS		
MD 13140: $MN_PROFIBUS_ALARM_ACCESS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	

PROFIBUS_ALARM_ACCESS (N, M)	MD 13140: $MN_PROFI-BUS_ALARM_ACCESS	
	1	
	Read access	

CONVERT_SCALING_SYSTEM (N, M)		MD 10260: $MN_CONVERT_SCAL-ING_SYSTEM		
MD 10260: $MN_CONVERT_SCALING_SYSTEM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

GMMC_INFO_NO_UNIT (N, M)		MD 17200: $MN_GMMC_INFO_NO_UNIT		
MD 17200: $MN_GMMC_INFO_NO_UNIT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

STAT_NAME (N, M)		MD 10670: $MN_STAT_NAME		
MD 10670: $MN_STAT_NAME				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

TU_NAME (N, M)		MD 10672: $MN_TU_NAME		
MD 10672: $MN_TU_NAME				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	

TU_NAME (N, M)	MD 10672: $MN_TU_NAME	
	1	
	Read access	

3.2.6.10 O: Internal

N-O

ASUP_MASK (N, O)		MD 19500: $ON_ASUP_MASK		
MD 19500: $ON_ASUP_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

AXIS_FUNCTION_MASK (N, O)		MD 19310: $ON_AXIS_FUNCTION_MASK		
MD 19310: $ON_AXIS_FUNCTION_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

CC_EVENT_MASK (N, O)		MD 19600: $ON_CC_EVENT_MASK		
MD 19600: $ON_CC_EVENT_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

COMP_MASK (N, O)		MD 19300: $ON_COMP_MASK		
MD 19300: $ON_COMP_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

FFW_MODE_MASK (N, O)		MD 19400: $ON_FFW_MODE_MASK		
MD 19400: $ON_FFW_MODE_MASK				
Units and value ranges				
Physical unit:	Data type:			

FFW_MODE_MASK (N, O)		MD 19400: $ON_FFW_MODE_MASK		
-	TYPE_CHAR			
			Read access	

IPO_FUNCTION_MASK (N, O)		MD 19330: $ON_IPO_FUNCTION_MASK		
MD 19330: $ON_IPO_FUNCTION_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

NUM_ABS_INC_REFP_AXES (N, O)		MD 19130: $ON_NUM_ABS_INC_REFP_AXES		
MD 19130: $ON_NUM_ABS_INC_REFP_AXES				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

NUM_AXES_IN_SYSTEM (N, O)		MD 19100: $ON_NUM_AXES_IN_SYS-TEM		
MD 19100: $ON_NUM_AXES_IN_SYSTEM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

NUM_CHANNELS (N, O)		MD 19200: $ON_NUM_CHANNELS		
MD 19200: $ON_NUM_CHANNELS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

NUM_IPO_AXES (N, O)		MD 19110: $ON_NUM_IPO_AXES		
MD 19110: $ON_NUM_IPO_AXES				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

NUM_MODE_GROUPS (N, O)		MD 19220: $ON_NUM_MODE_GROUPS		
MD 19220: $ON_NUM_MODE_GROUPS				
Units and value ranges				
Physical unit:	Data type:			

NUM_MODE_GROUPS (N, O)		MD 19220: $ON_NUM_MODE_GROUPS		
-	TYPE_CHAR			
			Read access	

NUM_SAFE_AXES (N, O)		MD 19120: $ON_NUM_SAFE_AXES		
MD 19120: $ON_NUM_SAFE_AXES				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

PLC_C_USER_MEM_SIZE (N, O)		MD 19280: $ON_PLC_C_USER_MEM_SIZE		
MD 19280: $ON_PLC_C_USER_MEM_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

PLC_USER_MEM_SIZE (N, O)		MD 19270: $ON_PLC_USER_MEM_SIZE		
MD 19270: $ON_PLC_USER_MEM_SIZE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

PROFIBUS (N, O)		MD 19290: $ON_PROFIBUS		
MD 19290: $ON_PROFIBUS				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
			Read access	

PROG_MASK (N, O)		MD 19340: $ON_PROG_MASK		
MD 19340: $ON_PROG_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

TECHNO_FUNCTION_MASK (N, O)		MD 19320: $ON_TECHNO_FUNC-TION_MASK		
MD 19320: $ON_TECHNO_FUNCTION_MASK				
Units and value ranges				
Physical unit:	Data type:			

TECHNO_FUNCTION_MASK (N, O)		MD 19320: $ON_TECHNO_FUNC-TION_MASK		
-	TYPE_DWORD			
			Read access	

TRAFO_TYPE_MASK (N, O)		MD 19410: $ON_TRAFO_TYPE_MASK		
MD 19410: $ON_TRAFO_TYPE_MASK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

USER_MEM_BUFFERED (N, O)		MD 19250: $ON_USER_MEM_BUF-FERED		
MD 19250: $ON_USER_MEM_BUFFERED				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
			Read access	

3.2.6.11 PA: Protection zones

Up to 10 protection zones can be defined. Each protection zone is described by a polygon function consisting of up to 10 elements. The module PA contains the individual coordinates of the polygon elements. The protection zones are addressed via the variable indices. The physical unit of the parameters can be read from the variable "$$basicLengthUnit" in the module Y in area N.

The classification as NCK or channel-specific protection zones does not affect the protection zone monitoring function, but indicates the area in which the protection zone is registered.

MDD_PA_CENT_ABS_0 (N, PA)		$SN_PA_CENT_ABS[x,0] x = Number protection zone		A3
Abscissa arc centre of 1st contour element Absolute abscissa value of arc centre of 1st contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_1 (N, PA)		$SN_PA_CENT_ABS[x,1] x = Number protection zone		A3
Abscissa arc centre of 2nd contour element Absolute abscissa value of arc centre of 2nd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_2 (N, PA)		$SN_PA_CENT_ABS[x,2] x = Number protection zone		A3
Abscissa arc centre of 3rd contour element Absolute abscissa value of arc centre of 3rd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_3 (N, PA)		$SN_PA_CENT_ABS[x,3] x = Number protection zone		A3
Abscissa arc centre of 4th contour element Absolute abscissa value of arc centre of 4th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_4 (N, PA)		$SN_PA_CENT_ABS[x,4] x = Number protection zone		A3
Abscissa arc centre of 5th contour element Absolute abscissa value of arc centre of 5th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CENT_ABS_4 (N, PA)		$SN_PA_CENT_ABS[x,4] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_5 (N, PA)		$SN_PA_CENT_ABS[x,5] x = Number protection zone		A3
Abscissa arc centre of 6th contour element Absolute abscissa value of arc centre of 6th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_6 (N, PA)		$SN_PA_CENT_ABS[x,6] x = Number protection zone		A3
Abscissa arc centre of 7th contour element Absolute abscissa value of arc centre of 7th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_7 (N, PA)		$SN_PA_CENT_ABS[x,7] x = Number protection zone		A3
Abscissa arc centre of 8th contour element Absolute abscissa value of arc centre of 8th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_8 (N, PA)		$SN_PA_CENT_ABS[x,8] x = Number protection zone		A3
Abscissa arc centre of 9th contour element Absolute abscissa value of arc centre of 9th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ABS_9 (N, PA)		$SN_PA_CENT_ABS[x,9] x = Number protection zone		A3
Abscissa arc centre of 10th contour element Absolute abscissa value of arc centre of 10th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_0 (N, PA)		$SN_PA_CENT_ORD[x,0] x = Number protection zone		A3
Ordinate arc centre of 1st contour element Absolute ordinate value of arc centre of 1st contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_1 (N, PA)		$SN_PA_CENT_ORD[x,1] x = Number protection zone		A3
Ordinate arc centre of 2nd contour element Absolute ordinate value of arc centre of 2nd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CENT_ORD_1 (N, PA)		$SN_PA_CENT_ORD[x,1] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_2 (N, PA)		$SN_PA_CENT_ORD[x,2] x = Number protection zone		A3
Odinate arc centre of 3rd contour element Absolute ordinate value of arc centre of 3rd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_3 (N, PA)		$SN_PA_CENT_ORD[x,3] x = Number protection zone		A3
Ordinate arc centre of 4th contour element Absolute ordinate value of arc centre of 4th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_4 (N, PA)		$SN_PA_CENT_ORD[x,4] x = Number protection zone		A3
Ordinate arc centre of 5th contour element Absolute ordinate value of arc centre of 5th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_5 (N, PA)		$SN_PA_CENT_ORD[x,5] x = Number protection zone		A3
Ordinate arc centre of 6th contour element Absolute ordinate value of arc centre of 6th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_6 (N, PA)		$SN_PA_CENT_ORD[x,6] x = Number protection zone		A3
Ordinate arc centre of 7th contour element Absolute ordinate value of arc centre of 7th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_7 (N, PA)		$SN_PA_CENT_ORD[x,7] x = Number protection zone		A3
Ordinate arc centre of 8th contour element Absolute ordinate value of arc centre of 8th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_8 (N, PA)		$SN_PA_CENT_ORD[x,8] x = Number protection zone		A3
Ordinate arc centre of 9th contour element Absolute ordinate value of arc centre of 9th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CENT_ORD_8 (N, PA)		$SN_PA_CENT_ORD[x,8] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CENT_ORD_9 (N, PA)		$SN_PA_CENT_ORD[x,9] x = Number protection zone		A3
Ordinate arc centre of 10th contour element Absolute ordinate value of arc centre of 10th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_0 (N, PA)		$SN_PA_CONT_ABS[x,0] x = Number protection zone		A3
Abscissa end point of 1st contour element Absolute abscissa value of end point of 1st contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_1 (N, PA)		$SN_PA_CONT_ABS[x,1] x = Number protection zone		A3
Abscissa end point of 2nd contour element Absolute abscissa value of end point of 2nd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_2 (N, PA)		$SN_PA_CONT_ABS[x,2] x = Number protection zone		A3
Abscissa end point of 3rd contour element Absolute abscissa value of end point of 3rd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_3 (N, PA)		$SN_PA_CONT_ABS[x,3] x = Number protection zone		A3
Abscissa end point of 4th contour element Absolute abscissa value of end point of 4th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_4 (N, PA)		$SN_PA_CONT_ABS[x,4] x = Number protection zone		A3
Abscissa end point of 5th contour element Absolute abscissa value of end point of 5th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_5 (N, PA)		$SN_PA_CONT_ABS[x,5] x = Number protection zone		A3
Abscissa end point of 6th contour element Absolute abscissa value of end point of 6th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CONT_ABS_5 (N, PA)		$SN_PA_CONT_ABS[x,5] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_6 (N, PA)		$SN_PA_CONT_ABS[x,6] x = Number protection zone		A3
Abscissa end point of 7th contour element Absolute abscissa value of end point of 7th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_7 (N, PA)		$SN_PA_CONT_ABS[x,7] x = Number protection zone		A3
Abscissa end point of 8th contour element Absolute abscissa value of end point of 8th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_8 (N, PA)		$SN_PA_CONT_ABS[x,8] x = Number protection zone		A3
Abscissa end point of 9th contour element Absolute abscissa value of end point of 9th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ABS_9 (N, PA)		$SN_PA_CONT_ABS[x,9] x = Number protection zone		A3
Abscissa end point of 10th contour element Absolute abscissa value of end point of 10th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_0 (N, PA)		$SN_PA_CONT_ORD[x,0] x = Number protection zone		A3
Ordinate end point of 1st contour element Absolute ordinate value of end point of 1st contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_1 (N, PA)		$SN_PA_CONT_ORD[x,1] x = Number protection zone		A3
Ordinate end point of 2nd contour element Absolute ordinate value of end point of 2nd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_2 (N, PA)		$SN_PA_CONT_ORD[x,2] x = Number protection zone		A3
Odinate end point of 3rd contour element Absolute ordinate value of end point of 3rd contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CONT_ORD_2 (N, PA)		$SN_PA_CONT_ORD[x,2] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_3 (N, PA)		$SN_PA_CONT_ORD[x,3] x = Number protection zone		A3
Ordinate end point of 4th contour element Absolute ordinate value of end point of 4th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_4 (N, PA)		$SN_PA_CONT_ORD[x,4] x = Number protection zone		A3
Ordinate end point of 5th contour element Absolute ordinate value of end point of 5th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_5 (N, PA)		$SN_PA_CONT_ORD[x,5] x = Number protection zone		A3
Ordinate end point of 6th contour element Absolute ordinate value of end point of 6th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_6 (N, PA)		$SN_PA_CONT_ORD[x,6] x = Number protection zone		A3
Ordinate end point of 7th contour element Absolute ordinate value of end point of 7th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_7 (N, PA)		$SN_PA_CONT_ORD[x,7] x = Number protection zone		A3
Ordinate end point of 8th contour element Absolute ordinate value of end point of 8th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_8 (N, PA)		$SN_PA_CONT_ORD[x,8] x = Number protection zone		A3
Ordinate end point of 9th contour element Absolute ordinate value of end point of 9th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_CONT_ORD_9 (N, PA)		$SN_PA_CONT_ORD[x,9] x = Number protection zone		A3
Ordinate end point of 10th contour element Absolute ordinate value of end point of 10th contour element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

MDD_PA_CONT_ORD_9 (N, PA)		$SN_PA_CONT_ORD[x,9] x = Number protection zone		A3
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_MINUS_LIM (N, PA)		$SN_PA_MINUS_LIM[x] x = Number protection zone		A3
Lower limit of protection zone, applicate Limitation in the minus direction of the protection zone in the axis that is perpendicular to the polygon definition (applicate)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDD_PA_PLUS_LIM (N, PA)		$SN_PA_PLUS_LIM[x] x = Number protection zone		A3
Upper boundary of protection zone, applicate Limitation in the plus direction of the protection zone in the axis that is perpendicular to the polygon definition (applicate)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_ACTIV_IMMED (N, PA)		$SN_PA_ACTIV_IMMED[x] x = Number protection zone		A3
Active immediately after referencing Code for "active immediately after referencing", i.e. the protection zone is active as soon as the control has been started up and the axes have been referenced				
Return value	0 = protection zone is not active immediately 1 = protection zone is active immediately			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_NUM (N, PA)		$SN_PA_CONT_NUM[x] x = Number protection zone		A3
Number of valid contour elements				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	$$numContourInProtArea
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_0 (N, PA)		$SN_PA_CONT_TYP[x,0] x = Number protection zone		A3
Contour type of 1st contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_1 (N, PA)		$SN_PA_CONT_TYP[x,1] x = Number protection zone		A3
Contour type of 2nd contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_2 (N, PA)		$SN_PA_CONT_TYP[x,2] x = Number protection zone		A3
Contour type of 3rd contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_3 (N, PA)		$SN_PA_CONT_TYP[x,3] x = Number protection zone		A3
Contour type of 4th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_4 (N, PA)		$SN_PA_CONT_TYP[x,4] x = Number protection zone		A3
Contour type of 5th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_5 (N, PA)		$SN_PA_CONT_TYP[x,5] x = Number protection zone		A3
Contour type of 6th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_6 (N, PA)		$SN_PA_CONT_TYP[x,6] x = Number protection zone		A3
Contour type of 7th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_7 (N, PA)		$SN_PA_CONT_TYP[x,7] x = Number protection zone		A3
Contour type of 8th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_8 (N, PA)		$SN_PA_CONT_TYP[x,8] x = Number protection zone		A3
Contour type of 9th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_CONT_TYP_9 (N, PA)		$SN_PA_CONT_TYP[x,9] x = Number protection zone		A3
Contour type of 10th contour element				
Return value	0 = G1 1 = G2 2 = G3			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_LIM_3DIM (N, PA)		$SN_PA_LIM_3DIM[x] x = Number protection zone		A3
Limitation of protection zone applicate Code for limitation of protection zone in the axis that is perpendicular to polygon definition (applicate)				
Return value	0 = no limitation 1 = limitation in positive direction 2 = limitation in negative direction 3 = limitation in both directions			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_ORI (N, PA)		$SN_PA_ORI[x] x = Number protection zone		A3
Plane assignment of protection zone Code for plane assignment of protection zone				
Return value	0 = G17 1 = G18 2 = G19			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

MDU_PA_TW (N, PA)		$SN_PA_T_W[x] x = Number protection zone		A3
Workpiece or tool-related protection zone Code for workpiece or tool-oriented protection zone				
Return value	0 = workpiece-related 1 = reserved 2 = reserved 3 = tool-related			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numProtArea	
Validity:	From NCK version 3.6		Read access	

modelChangeCounter (N, PA)				
Modification counter of the machine model				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

pa3DState (N, PA)				
Activation state of a protection zone				

pa3DState (N, PA)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_CHAR	0	0	3
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access	

pa3DProtDState (N, PA)				
Status of the PI service _N_PROT_D Status of the PI service _N_PROT_D. The line index has the following meaning: 1: Counter for PI calls 2: Current status (0=not calculated, 1=running calculation, 2=ready) 3: Number of a potentially occuring alarm during calculation (0=no alarm)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

pa3DProtDistance (N, PA)				
Distance vector of the PI service _N_PROT_D Distance vector of the PI service _N_PROT_D. The line index has the following meaning: 1: X component 2: Y component 3: Z component 4. Absolute value of the vector				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			4	
			Read access	

kinChainName (N, PA)	$NK_CHAIN_NAME			
Name of the kinematic chain Name of the kinematic chain This variable no longer exists in versions higher than NCK.P7_92.				
Units and value ranges				
Physical unit:	Data type:			

kinChainName (N, PA)		$NK_CHAIN_NAME		
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAINS	
			Read access and write access	

kinChain1stElem (N, PA)		$NK_1ST_ELEM		
First element of the kinematic chain First element of the kinematic chain This variable no longer exists in versions higher than NCK.P7_92.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAINS	
			Read access and write access	

kinElemName (N, PA)		$NK_NAME		
Name of a kinematic element				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemNext (N, PA)		$NK_NEXT		
Reference to the next kinematic element				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemParallel (N, PA)		$NK_PARALLEL		
Reference to 1st element of a diverging chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

kinElemParallel (N, PA)	$NK_PARALLEL	
Addressing		
Line index:	Max. line index:	
	$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
	Read access and write access	

kinElemNextP (N, PA)		$NK_NEXTP		
Reference to a branching kinematic chain Reference to a branching kinematic chain This variable no longer exists in versions higher than NCK.P7_92.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemSwitchIndex (N, PA)		$NK_SWITCH_INDEX		
Index of a switch Index of a switch in the kinematic chain.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemSwitchPos (N, PA)		$NK_SWITCH_POS		
Position of a switch Position of a switch in a kinematic chain.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemType (N, PA)		$NK_TYPE		
Type of kinematic element				
Units and value ranges				
Physical unit:	Data type:			

kinElemType (N, PA)		$NK_TYPE		
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemNameOld (N, PA)				
Name of a kinematic element (old) Name of a kinematic element (old). Using this variable, you can access the same data as with kinElemName (column index 1030). On the one hand, the variable is necessary due to an address shift compared with software version 83, and, on the other hand, to lock the (old) column index 1040 for other applications.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemNextOld (N, PA)				
Reference to next kinematic element (old) Reference to next kinematic element (old). Using this variable, you can access the same data as with kinElemNameNext (column index 1032). On the one hand, the variable is necessary due to an address shift compared with software version 83, and, on the other hand, to lock the (old) column index 1041 for other applications.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemOffDir0 (N, PA)		$NK_OFF_DIR[0]		
Offset or direction component in the X direction				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemOffDir1 (N, PA)	$NK_OFF_DIR[1]	
Offset or direction component in the Y direction		

kinElemOffDir1 (N, PA)		$NK_OFF_DIR[1]		
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemOffDir2 (N, PA)		$NK_OFF_DIR[2]		
Offset or direction component in the Z direction				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemOffDirFine0 (N, PA)				
Fine offset of direction component in X direction dummy, formerly: Fine offset of direction component in X direction				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
			No access	

kinElemOffDirFine1 (N, PA)				
Fine offset of direction component in Y direction dummy, formerly: Fine offset of direction component in Y direction				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
			No access	

kinElemOffDirFine2 (N, PA)				
Fine offset of direction component in Z direction dummy, formerly: Fine offset of direction component in Z direction				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
			No access	

kinElemOffDirFineLim (N, PA)				
Input limit for the fine offset of the offset dummy, formerly: Input limit for the fine offset of the offset				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
			No access	

kinElemAxis (N, PA)		$NK_AXIS		
Machine axis or OEM object name				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemAxisOffset (N, PA)		$NK_A_OFF		
Axis offset				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_CHAIN_ELEM	
			Read access and write access	

kinElemAxisOffsetFine (N, PA)				
Fine offset of axis offset dummy, formerly: Fine offset of axis offset				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
			No access	

kinElemAxisOffsetFineLim (N, PA)				
Input limit for the fine offset of the axis offset dummy, formerly: Input limit for the fine offset of the axis offset				
Units and value ranges				
Physical unit:	Data type:			

kinElemAxisOffsetFineLim (N, PA)				
mm, inch, degree, user defined	TYPE_DOUBLE			
			No access	

kinSwitch (N, PA)		$NK_SWITCH		
Switch position Position of a switch in a kinematic chain.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_KIN_SWITCHES	
			Read access and write access	

pa3DProtName (N, PA)		$NP_PROT_NAME		
Name of a protection zone				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access and write access	

pa3DChainName (N, PA)		$NP_CHAIN_NAME		
Name of chain carrying the protection zone Name of the kinematic chain carrying the protection area This variable no longer exists in versions higher than NCK.P7_92.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access and write access	

pa3DChainElem (N, PA)		$NP_CHAIN_ELEM		
Name of element carrying the protection zone Name of the kinematic element carrying the protection zone				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

pa3DChainElem (N, PA)	$NP_CHAIN_ELEM	
Addressing		
Line index:	Max. line index:	
	MN_MM_MAXNUM_3D_PROT_AREAS	
	Read access and write access	

pa3DProtType (N, PA)		$NP_PROT_TYPE		
Type of protection area Type of protection area. Valid are the values "MACHINE" or "TOOL". No difference made between lower and upper cases.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:		Max. line index:		
		MN_MM_MAXNUM_3D_PROT_AREAS		
		Read access and write access		

pa3D1stProt (N, PA)		$NP_1ST_PROT		
Name of the first element of the protection zone				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:		Max. line index:		
		MN_MM_MAXNUM_3D_PROT_AREAS		
		Read access and write access		

pa3DProtColor (N, PA)		$NP_PROT_COLOR		
Color and transparency of the protection area Color and transparency of the protection area in QT format (HighByte: Transparency. Bytes 0 - 2: RGB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:		Max. line index:		
		MN_MM_MAXNUM_3D_PROT_AREAS		
		Read access and write access		

pa3DBitIndex (N, PA)		$NP_BIT_NO		
Index bit on the VDI interface Index of the bits assigned on the VDI interface				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			

pa3DBitIndex (N, PA)	$NP_BIT_NO	
Addressing		
Line index:	Max. line index:	
	MN_MM_MAXNUM_3D_PROT_AREAS	
	Read access and write access	

pa3DInitStat (N, PA)		$NP_INIT_STAT		
Initialization status of the protection zone				
Return value	ASCII code of one of the following letters: 'A', 'a', 'I', 'i', 'P', 'p'			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access and write access	

pa3DAuxIndex0 (N, PA)		$NP_INDEX[0]		
1st index for definition of var. protection zones 1st index for definition of variable protection zones				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access and write access	

pa3DAuxIndex1 (N, PA)		$NP_INDEX[1]		
2nd index for definition of var. protection zones 2nd index for definition of variable protection zones				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access and write access	

pa3DAuxIndex2 (N, PA)		$NP_INDEX[2]		
3rd index to define variable protection areas 3rd index for definition of variable protection zones				
Units and value ranges				
Physical unit:	Data type:			

pa3DAuxIndex2 (N, PA)		$NP_INDEX[2]		
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access and write access	

pa3DProtDLevel (N, PA)		$NP_PROT_D_LEVEL		
Detailing level of a protection area				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access and write access	

pa3DElemName (N, PA)		$NP_NAME		
Name of the protection zone element				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemNext (N, PA)		$NP_NEXT		
Name of the next protection zone element				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemNextP (N, PA)		$NP_NEXTP		
Name of next parallel protection area element				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	

pa3DElemNextP (N, PA)	$NP_NEXTP	
		$MN_MM_MAXNUM_3D_PROT_AREA_ELEM
		Read access and write access

pa3DElemAdd (N, PA)		$NP_ADD		
Name of a protection zone to be inserted				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemColor (N, PA)		$NP_COLOR		
Color and transparency of a prot. area element Color and transparency of a protection area element in QT format (HighByte: Transparency. Bytes 0 - 2: RGB				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemUsage (N, PA)		$NP_USAGE		
Use of a protection area element Use of the protection area element to visualize ('V' or 'v'), to avoid collisions ('C' or 'c') or both ('A' or 'a').				
Return value	ASCII code of one of the following letters: 'A' 'a', 'C', 'c', 'V', 'v'			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Addressing				
Line index:			Max. line index:	
			MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemType (N, PA)		$NP_TYPE		
Type of the protection zone element				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	

pa3DElemType (N, PA)	$NP_TYPE	
		$MN_MM_MAXNUM_3D_PROT_AREA_ELEM
		Read access and write access

pa3DElemFileName (N, PA)		$NP_FILENAME		
File name of a prot. area elem. of the "FILE" type Name of file containing the description of a protection area element of the "FILE" type.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemDLevel (N, PA)		$NP_D_LEVEL		
Detailing level of a protection area element				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access and write access	

pa3DElemPara0 (N, PA)		$NP_PARA[0]		
1st geometry parameter of protection zone element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemPara1 (N, PA)		$NP_PARA[1]		
2nd geometry parameter of protection zone element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemPara2 (N, PA)		$NP_PARA[2]		
3rd geometry parameter of protection zone element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemOffset0 (N, PA)		$NP_OFF[0]		
X components of the offset				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemOffset1 (N, PA)		$NP_OFF[1]		
Y components of the offset				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemOffset2 (N, PA)		$NP_OFF[2]		
Z components of the offset				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemDir0 (N, PA)	$NP_DIR[0]	
X components of rotary axis		

pa3DElemDir0 (N, PA)		$NP_DIR[0]		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemDir1 (N, PA)		$NP_DIR[1]		
Y components of rotary axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemDir2 (N, PA)		$NP_DIR[2]		
Z components of rotary axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DElemAngle (N, PA)		$NP_ANG		
Turning angle				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREA_ELEM	
			Read access and write access	

pa3DTElemName (N, PA)		$NP_T_NAME		
Name of the tool protection area element				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

pa3DTElemName (N, PA)	$NP_T_NAME	
Addressing		
Line index:	Max. line index:	
	$MN_MM_MAXNUM_3D_T_PROT_ELEM	
	Read access	

pa3DTElemType (N, PA)		$NP_T_TYPE		
Type of the tool protection area element				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_T_PROT_ELEM	
			Read access	

pa3DTElemFileName (N, PA)		$NP_T_FILENAME		
File name tool protec area elem of the "FILE" type Name of file containing the description of the tool protection area element of the "FILE" type.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_T_PROT_ELEM	
			Read access	

pa3DTElemPara0 (N, PA)		$NP_T_PARA[0]		
1st geometry param. of the tool prot. area element 1st geometry parameter of the tool protection area element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_T_PROT_ELEM	
			Read access	

pa3DTElemPara1 (N, PA)		$NP_T_PARA[1]		
2nd geometry param. of the tool prot. area element 2nd geometry parameter of the tool protection area element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			

pa3DTElemPara1 (N, PA)	$NP_T_PARA[1]	
Addressing		
Line index:	Max. line index:	
	$MN_MM_MAXNUM_3D_T_PROT_ELEM	
	Read access	

pa3DTElemPara2 (N, PA)		$NP_T_PARA[2]		
3rd geometry param. of the tool prot. area element 3rd geometry parameter of the tool protection area element				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		$MN_MM_MAXNUM_3D_T_PROT_ELEM		
		Read access		

pa3DTElemOffset0 (N, PA)		$NP_T_OFF[0]		
X component of the offset X components of the offset				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		$MN_MM_MAXNUM_3D_T_PROT_ELEM		
		Read access		

pa3DTElemOffset1 (N, PA)		$NP_T_OFF[1]		
Y component of the offset Y components of the offset				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		$MN_MM_MAXNUM_3D_T_PROT_ELEM		
		Read access		

pa3DTElemOffset2 (N, PA)		$NP_T_OFF[2]		
Z component of the offset Z components of the offset				
Units and value ranges				
Physical unit:	Data type:			

pa3DTElemOffset2 (N, PA)		$NP_T_OFF[2]		
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_T_PROT_ELEM	
			Read access	

pa3DTElemDir0 (N, PA)		$NP_T_DIR[0]		
X component of the axis of rotation X components of rotary axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_T_PROT_ELEM	
			Read access	

pa3DTElemDir1 (N, PA)		$NP_T_DIR[1]		
Y component of the axis of rotation Y components of rotary axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_T_PROT_ELEM	
			Read access	

pa3DTElemDir2 (N, PA)		$NP_T_DIR[2]		
Z component of the axis of rotation Z components of rotary axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_T_PROT_ELEM	
			Read access	

pa3DTElemAngle (N, PA)	$NP_T_ANG	
Rotation angle Turning angle		

pa3DTElemAngle (N, PA)		$NP_T_ANG		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_T_PROT_ELEM	
			Read access	

pa3DCollPair0 (N, PA)		$NP_COLL_PAIR[n, 0]		
1. Name of a protection area of a collision pair				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			MM_MAXNUM_3D_PROT_AREAS * (MM_MAXNUM_3D_PROT_AREAS - 1) / 2	
			Read access and write access	

pa3DCollPair1 (N, PA)		$NP_COLL_PAIR[n, 1]		
2. Name of a protection area of a collision pair				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			MM_MAXNUM_3D_PROT_AREAS * (MM_MAXNUM_3D_PROT_AREAS - 1) / 2	
			Read access and write access	

pa3DCollPairSafetyDist (N, PA)		$NP_SAFETY_DIST		
Safety distance of a collision pair				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			MM_MAXNUM_3D_PROT_AREAS * (MM_MAXNUM_3D_PROT_AREAS - 1) / 2	
			Read access and write access	

trafoDatBaseTool0 (N, PA)	$NT_BASE_TOOL[n,0]	
X components of the base tool. X components of the base tool		

trafoDatBaseTool0 (N, PA)		$NT_BASE_TOOL[n,0]		
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatBaseTool1 (N, PA)		$NT_BASE_TOOL[n,1]		
Y components of the base tool. Y components of the base tool				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatBaseTool2 (N, PA)		$NT_BASE_TOOL[n,2]		
Z components of the base tool. Z components of the base tool				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatName (N, PA)		$NT_NAME[n]		
Name of transformer data records Name of transformer data record				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatPChainFirstElem (N, PA)		$NT_P_CHAIN_FIRST_ELEM[n]		
Name of first element of kin. chain for workpiece Name of first element of kinetic chain for workpiece				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatTChainFirstElem (N, PA)		$NT_T_CHAIN_FIRST_ELEM[n]		
Name of first element of kinetic chain for tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatTrafoType (N, PA)		$NT_TRAFO_TYPE		
Transformer type				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatTChainName (N, PA)		$NT_T_CHAIN_NAME[n]		
Name of kin. chain for tool Name of kin. chain for tool This variable no longer exists in versions higher than NCK.P7_92.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatTChainLastElem (N, PA)		$NT_T_CHAIN_LAST_ELEM[n]		
Name of last element of kin. chain for tool				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatPChainName (N, PA)		$NT_P_CHAIN_NAME[n]		
Name of kin. chain for workpiece Name of kin. chain for workpiece This variable no longer exists in versions higher than NCK.P7_92.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatPChainLastElem (N, PA)		$NT_P_CHAIN_LAST_ELEM[n]		
Name of last element of kin. chain for workpiece				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatTRefElem (N, PA)		$NT_T_REF_ELEM[n]		
Reference point for tool length calculation				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatGeoAxName0 (N, PA)		$NT_GEO_AX_NAME[n, 0]		
Name of first geometry axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatGeoAxName1 (N, PA)		$NT_GEO_AX_NAME[n, 1]		
Name of second geometry axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatGeoAxName2 (N, PA)		$NT_GEO_AX_NAME[n, 2]		
Name of third geometry axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxName0 (N, PA)		$NT_ROT_AX_NAME[n, 0]		
Name of first rotary axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxName1 (N, PA)	$NT_ROT_AX_NAME[n, 1]	
Name of second rotary axis		

trafoDatRotAxName1 (N, PA)		$NT_ROT_AX_NAME[n, 1]		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxName2 (N, PA)		$NT_ROT_AX_NAME[n, 2]		
Name of third rotary axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatCloseChainT (N, PA)		$NT_CLOSE_CHAIN_T[n]		
Reference point for closing the tool chain Element whose end point is used as a reference point for closing the tool chain.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatCloseChainP (N, PA)		$NT_CLOSE_CHAIN_P[n]		
Reference point for closing the part chain Element whose end point is used as a reference point for closing the part chain.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxPos0 (N, PA)	$NT_ROT_AX_POS[n, 0]	
Position of the 1st manual rotary axis		

trafoDatRotAxPos0 (N, PA)		$NT_ROT_AX_POS[n, 0]		
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxPos1 (N, PA)		$NT_ROT_AX_POS[n, 1]		
Position of the 2nd manual rotary axis				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxPos2 (N, PA)		$NT_ROT_AX_POS[n, 2]		
Position of the 3rd manual rotary axis				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatCorrElemT0 (N, PA)		$NT_CORR_ELEM_T[n, 0]		
Alternative name:	trafoDatCorrElem0			
Name of 1st correction element in the tool chain Name of the 1st correction element in the tool chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatCorrElemT1 (N, PA)		$NT_CORR_ELEM_T[n, 1]		
Alternative name:	trafoDatCorrElem1			
Name of 2nd correction element in the tool chain Name of the 2nd correction element in the tool chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatCorrElemT2 (N, PA)		$NT_CORR_ELEM_T[n, 2]		
Alternative name:	trafoDatCorrElem2			
Name of 3rd correction element in the tool chain Name of the 3rd correction element in the tool chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatCorrElemT3 (N, PA)		$NT_CORR_ELEM_T[n, 3]		
Alternative name:	trafoDatCorrElem3			
Name of 4th correction element in the tool chain Name of the 4th correction element in the tool chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatHirthOff0 (N, PA)		$NT_HIRTH_OFF[n, 0]		
Angle offs. of 1st rot. axis w. Hirth tooth system Angle offset of 1st rotary axis with Hirth tooth system				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			

trafoDatHirthOff0 (N, PA)	$NT_HIRTH_OFF[n, 0]	
Addressing		
Line index:	Max. line index:	
	$MN_MM_NUM_TRAFO_DATA_SETS	
	Read access	

trafoDatHirthOff1 (N, PA)		$NT_HIRTH_OFF[n, 1]		
Angle offs. of 2nd rot. axis w. Hirth tooth system Angle offset of 2nd rotary axis with Hirth tooth system				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatHirthOff2 (N, PA)		$NT_HIRTH_OFF[n, 2]		
Angle offs. of 3rd rot. axis w. Hirth tooth system Angle offset of 3rd rotary axis with Hirth tooth system				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatHirthInc0 (N, PA)		$NT_HIRTH_INC[n, 0]		
Angle incr. of 1st rot. axis w. Hirth tooth system Angle increment of 1st rotary axis with Hirth tooth system				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatHirthInc1 (N, PA)		$NT_HIRTH_INC[n, 1]		
Angle incr. of 2nd rot. axis w. Hirth tooth system Angle increment of 2nd rotary axis with Hirth tooth system				
Units and value ranges				
Physical unit:	Data type:			

trafoDatHirthInc1 (N, PA)		$NT_HIRTH_INC[n, 1]		
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatHirthInc2 (N, PA)		$NT_HIRTH_INC[n, 2]		
Angle incr. of 3rd rot. axis w. Hirth tooth system Angle increment of 3rd rotary axis with Hirth tooth system				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxMin0 (N, PA)		$NT_ROT_AX_MIN[n, 0]		
Minimum position of 1st manual rotary axis				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxMin1 (N, PA)		$NT_ROT_AX_MIN[n, 1]		
Minimum position of 2nd manual rotary axis				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxMin2 (N, PA)		$NT_ROT_AX_MIN[n, 2]		
Minimum position of 3rd manual rotary axis				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			

trafoDatRotAxMin2 (N, PA)	$NT_ROT_AX_MIN[n, 2]	
Addressing		
Line index:	Max. line index:	
	$MN_MM_NUM_TRAFO_DATA_SETS	
	Read access	

trafoDatRotAxMax0 (N, PA)		$NT_ROT_AX_MAX[n, 0]		
Maximum position of 1st manual rotary axis				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxMax1 (N, PA)		$NT_ROT_AX_MAX[n, 1]		
Maximum position of 2nd manual rotary axis				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxMax2 (N, PA)		$NT_ROT_AX_MAX[n, 2]		
Maximum position of 3rd manual rotary axis				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatBaseOrient0 (N, PA)		$NT_BASE_ORIENT[n, 0]		
X component of basic tool orientation				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

trafoDatBaseOrient0 (N, PA)	$NT_BASE_ORIENT[n, 0]	
	$MN_MM_NUM_TRAFO_DATA_SETS	
	Read access	

trafoDatBaseOrient1 (N, PA)		$NT_BASE_ORIENT[n, 1]		
Y component of basic tool orientation				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatBaseOrient2 (N, PA)		$NT_BASE_ORIENT[n, 2]		
Z component of basic tool orientation				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatBaseOrientNormal0 (N, PA)		$NT_BASE_ORIENT_NORMAL[n, 0]		
X component of the normal vector of orientation				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatBaseOrientNormal1 (N, PA)		$NT_BASE_ORIENT_NORMAL[n, 1]		
Y component of the normal vector of orientation				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatBaseOrientNormal2 (N, PA)		$NT_BASE_ORIENT_NORMAL[n, 2]		
Z component of the normal vector of orientation				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatPoleLimit (N, PA)		$NT_POLE_LIMIT[n]		
End angle tolerance with interpol. through pole End angle tolerance with interpolation through pole				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatPoleTol (N, PA)		$NT_POLE_TOL[n]		
End angle tolerance for pole interpolation				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotOffsetFromFrame (N, PA)		$NT_ROT_OFFSET_FROM_FRAME[n]		
Accept rotary offs. for transformer select. fr. WO Accept rotary offset for transformer selection from WO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatIgnoreToolOrient (N, PA)		$NT_IGNORE_TOOL_ORIENT[n]		
Ignore tool orientation If this parameter is set, the orientation stored in the transformation data ($NT_BASE_ORIENT, $NT_BASE_ORIENT_NORMAL) is always used, regardless of the basic orientation contained in the tool data of an active tool, i.e. the orientation defined in the transformation data set takes precedence over the tool orientation.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatTrafoIncludesTool (N, PA)		$NT_TRAFO_INCLUDES_TOOL[n]		
Tool treated internally or externally This system variable indicates whether the tool is handled internally or externally in the case of an active transformation.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatPoleSideFix (N, PA)		$NT_POLE_SIDE_FIX[n]		
Working area in front of/behind pole for TRANSMIT Restriction of working area in front of/behind pole or no restriction, i.e. traversal through the pole. The assigned values have the following meanings: 0: No restriction of the working area. Traversal through the pole allowed. 1: Working area of linear axis for positions >=0, (if tool length compensation parallel to linear axis = 0) 2: Working area of linear axis for positions <=0, (if tool length compensation parallel to linear axis = 0)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatCntrl (N, PA)	$NT_CNTRL[n]	

Bit-coded control word

This data is a bit-coded control word, with which the behavior can be influenced in specific situations.

The individual bits have the following meaning:

Bit 0 : Not assigned

Bit 1 - 3: The orientation axis assigned to the bit (Bit 1: First orientation axis, Bit 2: Second orientation axis, Bit 3: Third orientation axis) is interpreted as speed-controlled spindle.

We currently only support the cases in which either the first or the third orientation axis is parameterized as a spindle (turning on milling machines or 5-axis milling on machines on which the third orientation axis is not operated in position control mode).

Bit 4 - 6: The orientation axis assigned to the bit (Bit 4: First orientation axis, Bit 5: Second orientation axis, Bit 6: Third orientation axis) is Hirth-toothed. For the Hirth toothing, only the machine data $MA_INDEX_AX_NUMERATOR, $MA_INDEX_AX_DENOMINATOR and $MA_INDEX_AX_OFFSET are evaluated.

Der Inhalt des Maschinendatums $MA_HIRTH_IS_ACTIVE wird nicht ausgewertet, d.h. die Achse muss nicht als echte Hirthachse prametriert sein.

Ist die Achse als Moduloachse parameriert, wird das Maschinendatum $MA_INDEX_AX_NUMERATOR durch das Maschinendatum $MA_MODULO_RANGE ersetzt. Die Abstände der zulässigen Achspositionen ergeben sich dann durch $MA_MODULO_RANGE / $MA_INDEX_AX_DENOMINATOR.

Das Maschinendatum $MA_INDEX_AX_OFFSET wird auch bei Moduloachsen ausgewertet.

Bit 7 - 8: Sind diese Bits gesetzt, werden an den Startpunkten der Teilketten (Bit7: Part-Kette; Bit 8: Tool-Kette) bei Bedarf intern automatisch zusätzliche konstante Kettenelmnente eingefügt, die eine Veebindung vom Endpunkt der Kette zum Maschinennullpunkt herstellen ("Kette schließen").

Bit 9 - 31: Nicht belegt

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			

Addressing

Line index:	Max. line index:
	$MN_MM_NUM_TRAFO_DATA_SETS
	Read access

trafoDatTrafoIndex (N, PA)	$NT_TRAFO_INDEX[n]	

Index f. transformation call w. convention. syntax

A transformation defined by kinematic chains can also be activated by conventional language commands, such as TRAORI(<n>) or TRANSMIT(<n>), instead of the call TRAFOON(<Name>), if a value not equal to zero is entered in this system data, and the transformation type is compatible with the conventionally named transformation type.

The hundreds and thousands digits indicate the channel in which the transformation can be called with a conventional language command. If both these digits are empty (zero), the definition applies to the first channel; this means that, for example, the entries "1" and "101" are equivalent.

In order for a transformation defined by kinematic chains to be called with a conventional language command, the three lowest-value decimal positions of this system data must not be zero. The orientation transformation indicated by the index 1 is also alternatively activated for compatibility reasons with the conventional call syntax with TRAORI(0), TRAORI() or TRAORI but not with TRAORI(1). The same applies to the other transformation types (TRANSMIT, TRACYL and TRAANG).

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			

trafoDatTrafoIndex (N, PA)	$NT_TRAFO_INDEX[n]	
Addressing		
Line index:	Max. line index:	
	$MN_MM_NUM_TRAFO_DATA_SETS	
	Read access	

trafoDatAuxPos0 (N, PA)		$NT_AUX_POS[n,0]		
X comp. of the auxil. position for meas. cycles X component of the auxiliary position for measuring cycles				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		$MN_MM_NUM_TRAFO_DATA_SETS		
		Read access		

trafoDatAuxPos1 (N, PA)		$NT_AUX_POS[n,1]		
Y comp. of the auxil. position for meas. cycles Y component of the auxiliary position for measuring cycles				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		$MN_MM_NUM_TRAFO_DATA_SETS		
		Read access		

trafoDatAuxPos2 (N, PA)		$NT_AUX_POS[n,2]		
Z comp. of the auxil. position for meas. cycles Z component of the auxiliary position for measuring cycles				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		$MN_MM_NUM_TRAFO_DATA_SETS		
		Read access		

trafoDatIdent0 (N, PA)		$NT_IDENT[n, 0]		
Identifier 0, ID number 0, no significance in NCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			

trafoDatIdent0 (N, PA)	$NT_IDENT[n, 0]	
Addressing		
Line index:	Max. line index:	
	$MN_MM_NUM_TRAFO_DATA_SETS	
	Read access	

trafoDatIdent1 (N, PA)		$NT_IDENT[n, 1]		
Identifier 1, ID number 1, no significance in NCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatIdent2 (N, PA)		$NT_IDENT[n, 2]		
Identifier 2, ID number 2, no significance in NCK				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatCorrElemP0 (N, PA)		$NT_CORR_ELEM_P[n, 0]		
Name of 1st correction element in part chain Name of the 1st correction element in the part chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatCorrElemP1 (N, PA)		$NT_CORR_ELEM_P[n, 1]		
Name of 2st correction element in part chain Name of the 2st correction element in the part chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

trafoDatCorrElemP1 (N, PA)	$NT_CORR_ELEM_P[n, 1]	
Addressing		
Line index:	Max. line index:	
	$MN_MM_NUM_TRAFO_DATA_SETS	
	Read access	

trafoDatCorrElemP2 (N, PA)		$NT_CORR_ELEM_P[n, 2]		
Name of 3st correction element in part chain Name of the 3st correction element in the part chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatCorrElemP3 (N, PA)		$NT_CORR_ELEM_P[n, 3]		
Name of 4st correction element in part chain Name of the 4st correction element in the part chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxCnt0 (N, PA)		$NT_ROT_AX_CNT[n, 0]		
Number of relevant rotary axes in the part chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxCnt1 (N, PA)		$NT_ROT_AX_CNT[n, 1]		
Number of relevant rotary axes in the tool chain				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			

trafoDatRotAxCnt1 (N, PA)	$NT_ROT_AX_CNT[n, 1]	
Addressing		
Line index:	Max. line index:	
	$MN_MM_NUM_TRAFO_DATA_SETS	
	Read access	

trafoDatBaseToolComp (N, PA)	$NT_BASE_TOOL_COMP[n]			
Compensation of the tool axes with frames. This bit-coded system variable means that components of the BaseTool can be compensated via the transformation frame so that there is no change in the WCS component on transformation selection. Bit0: MD24920 $NT_BASE_TOOL[n, 0] is compensated via $P_TRAFRAME. Bit1: MD24920 $NT_BASE_TOOL[n, 1] is compensated via $P_TRAFRAME. Bit2: MD24920 $NT_BASE_TOOL[n, 2] is compensated via $P_TRAFRAME. This function is available only if the system frame $P_TRAFRAME is configured via MD28082 $MC_MM_SYSTEM_FRAME_MASK bit6.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxOffset0 (N, PA)	$NT_ROT_AX_OFFSET[n,0]			
Offset of the first rotary axis in degrees. Position offset of the first (or only) rotary axis of the active transformation.				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

trafoDatRotAxOffset1 (N, PA)	$NT_ROT_AX_OFFSET[n,1]			
Offset of the second rotary axis in degrees. Position offset of the second rotary axis of the active transformation (only TRAORI).				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			

trafoDatRotAxOffset1 (N, PA)	$NT_ROT_AX_OFFSET[n,1]	
Addressing		
Line index:	Max. line index:	
	$MN_MM_NUM_TRAFO_DATA_SETS	
	Read access	

trafoDatRotAxOffset2 (N, PA)		$NT_ROT_AX_OFFSET[n,2]		
Offset of the third rotary axis in degrees. Position offset of the third rotary axis of the active transformation (only TRAORI).				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TRAFO_DATA_SETS	
			Read access	

3.2.6.12 RP: Arithmetic parameter

Arithmetic parameters are specific, predefined variables which are addressed with the address R and subsequent number. The contents and meaning of arithmetic parameters is specified by person programming the part program. The number of parameters can be set in the machine data 18156 (MM_NUM_R_PARAM_NCK).

RG (N, RP)		$RG[x] x = ParameterNo		PA
Global R variables				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			MM_NUM_R_PARAM_NCK	
Validity:	From NCK version 3.6		Read access and write access	

3.2.6.13 S: State data

During NC operation different internal states occur and system-specific data may change during operation. To distinguish those from system variables, they are classified as state data.

A distinction is made between:
- NCK-specific state data
- Mode-group-specific state data
- Channel-specific state data
- Drive-specific state data (FDD)
- Drive-specific state data (MSD)

aDbb (N, S)		$A_DBB[x] x = ByteNo		
PLC data byte The variable is used to read and write a data byte (8 bits) from PLC. The byte is unsigned and can be read in the range from 0 to 255 and written in the range from -128 to 255. A memory area is reserved in the communications buffer of these modules (DPR) for high-speed information exchange between PLC and NC. The PLC uses function calls (FC) and the NCK uses $ variables to access this memory.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

aDbd (N, S)		$A_DBD[x] x = Offset		
PLC data doubleword The variable is used to read and write a data doubleword (32 bits) from PLC. A memory area is reserved in the communications buffer of these modules (DPR) for high-speed information exchange between PLC and NC. The PLC uses function calls (FC) and the NCK uses $ variables to access this memory.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access and write access	

aDbr (N, S)		$A_DBR[x] x = Offset		
PLC Real data (32 bits) The variable is used to read and write real data (32 bits) from PLC. A memory area is reserved in the communications buffer of these modules (DPR) for high-speed information exchange between PLC and NC. The PLC uses function calls (FC) and the NCK uses $ variables to access this memory.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access and write access	

aDbsb (N, S)	$A_DBSB	
PLC data byte		

aDbsb (N, S)		$A_DBSB		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-128	127
Addressing				
Line index:			Max. line index:	
			1023	
			Read access	

aDbsw (N, S)		$A_DBSW		
PLC data word				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-32768	32767
Addressing				
Line index:			Max. line index:	
			1022	
			Read access	

aDbw (N, S)		$A_DBW[x] x = Offset		
PLC data word (unsigned) The variable is used to read and write a dataword (16 bits) from PLC. The byte is unsigned and can be read in the range from 0 to 65535 and written in the range from -32768 to 65535. A memory area is reserved in the communications buffer of these modules (DPR) for high-speed information exchange between PLC and NC. The PLC uses function calls (FC) and the NCK uses $ variables to access this memory.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

aDlb (N, S)		$A_DLB[index]		
Data byte in link area Data byte (8 bits) in link area				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

aDld (N, S)		$A_DLD[index]		
Access to data double word Data double word (32 bits) in link data area				
Units and value ranges				
Physical unit:	Data type:			

aDld (N, S)		$A_DLD[index]		
-	TYPE_DWORD			
			Read access and write access	

aDlr (N, S)		$A_DLR[index]		
Real data Read data (32 bits) in link data area				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access and write access	

aDlw (N, S)		$A_DLW[index]		
Data word in link area Data word (16 bits) in link data area				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

aDpInConf (N, S)		$A_DP_IN_CONF		
PROFIBUS configured input data areas				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	0xffffffff
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aDpInLength (N, S)		$A_DP_IN_LENGTH		
PROFIBUS length of input data area				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	128
Addressing				
Line index:			Max. line index:	
			32	
			Read access	

aDpInState (N, S)		$A_DP_IN_STATE		
PROFIBUS status of input data area				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

aDpInState (N, S)		$A_DP_IN_STATE		
-	TYPE_DWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			32	
			Read access	

aDpInValid (N, S)		$A_DP_IN_VALID		
PROFIBUS valid input data areas				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	0xffffffff
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aDpOutConf (N, S)		$A_DP_OUT_CONF		
PROFIBUS configured output data areas				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	0xffffffff
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aDpOutLength (N, S)		$A_DP_OUT_LENGTH		
PROFIBUS length of output data area				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	128
Addressing				
Line index:			Max. line index:	
			32	
			Read access	

aDpOutState (N, S)		$A_DP_OUT_STATE		
PROFIBUS status of output data area				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	3
Addressing				
Line index:			Max. line index:	

aDpOutState (N, S)	$A_DP_OUT_STATE	
	32	
	Read access	

aDpOutValid (N, S)		$A_DP_OUT_VALID		
PROFIBUS valid output data areas				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	0xffffffff
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aDpbIn (N, S)		$A_DPB_IN		
PROFIBUS input byte (unsigned)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	255
Addressing				
Line index:			Max. line index:	
			Read access	

aDpbOut (N, S)		$A_DPB_OUT		
PROFIBUS output byte (unsigned)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	255
Addressing				
Line index:			Max. line index:	
			Read access	

aDprIn (N, S)		$A_DPR_IN		
PROFIBUS input data (32 bit REAL)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			Read access	

aDprOut (N, S)		$A_DPR_OUT		
PROFIBUS output data (32 bit REAL)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			Read access	

aDpsbIn (N, S)		$A_DPSB_IN		
PROFIBUS input byte (signed)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-128	127
Addressing				
Line index:			Max. line index:	
			Read access	

aDpsbOut (N, S)		$A_DPSB_OUT		
PROFIBUS output byte (signed)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-128	127
Addressing				
Line index:			Max. line index:	
			Read access	

aDpsdIn (N, S)		$A_DPSD_IN		
PROFIBUS input data double word (signed)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			Read access	

aDpsdOut (N, S)	$A_DPSD_OUT	
PROFIBUS output data double word (signed)		

aDpsdOut (N, S)		$A_DPSD_OUT		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			Read access	

aDpswIn (N, S)		$A_DPSW_IN		
PROFIBUS input word (signed)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-32768	32767
Addressing				
Line index:			Max. line index:	
			Read access	

aDpswOut (N, S)		$A_DPSW_OUT		
PROFIBUS output word (signed)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-32768	32767
Addressing				
Line index:			Max. line index:	
			Read access	

aDpwIn (N, S)		$A_DPW_IN		
PROFIBUS input word (unsigned)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	65535
Addressing				
Line index:			Max. line index:	
			Read access	

aDpwOut (N, S)		$A_DPW_OUT		
PROFIBUS output word (unsigned)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	65535

aDpwOut (N, S)	$A_DPW_OUT	
Addressing		
Line index:	Max. line index:	
	Read access	

aInco (N, S)		$A_INCO[x] x = InputNo		
Comperator input NC				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

aInsip (N, S)				
Corresponds to $$safeIntInpValPlcBit				
Return value	0: Input not set 1: Input set			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			64	
			Read access	

aPbbIn (N, S)	$A_PBB_IN[index]			
IN data byte Data byte (8bits) in PLC input/output area IN (also available on 810D CCU2)				
Return value	Neg. values are also permitted in spite of TYPE_UWORD			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

aPbbOut (N, S)	$A_PBB_OUT[index]	
OUT data byte Data byte (8 bits) in PLC input/output area OUT (also available on 810D CCU2)		
Return value	Neg. values are also permitted in spite of TYPE_UWORD	

aPbbOut (N, S)		$A_PBB_OUT[index]		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

aPbdIn (N, S)		$A_PBD_IN[index]		
IN data double word Data double word (32bits) in PLC input/output area IN				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

aPbdOut (N, S)		$A_PBD_OUT[index]		
OUT data double word Data double word (32 bits) in the PLC input/output area OUT (also available on 810D CCU2)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access and write access	

aPbrIn (N, S)		$A_PBR_IN[index]		
IN Real data Real data (32bits) in PLC input/output area IN (also available on 810D CCU2)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

aPbrOut (N, S)		$A_PBR_OUT[index]		
OUT Real data Real data (32 bits) in the PLC input/output area OUT (also available on 810D CCU2)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access and write access	

aPbwIn (N, S)		$A_PBW_IN[index]		
IN data word Data word (16bits) in PLC input/output area IN (also available on 810D CCU2)				
Return value	Neg. values are also permitted in spite of TYPE_UWORD			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

aPbwOut (N, S)		$A_PBW_OUT[index]		
OUT data word Data word (16 bits) in the PLC input/output area OUT (also available on 810D CCU2)				
Return value	Neg. values are also permitted in spite of TYPE_UWORD			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access and write access	

aProbe (N, S)		$A_PROBE		
Probe status				
Return value	0: Not deflected 1: Deflected			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

aProbeLimited (N, S)		$A_PROBE_LIMITED		
Measuring velocity exceeded Contains the accumulated number of DP communication cycles where at least one limitation was effective.				
Return value	An increasing value signals that the frequency of the measuring probe signals must be reduced (i.e. by reducing the speed of the toothed wheel to be measured).			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	

aProbeLimited (N, S)	$A_PROBE_LIMITED	
Addressing		
Line index:	Max. line index:	
	2	
	Read access and write access	

anCollStateCond (N, S)	$AN_COLL_STATE_COND[i]	
Activation conditions of a protection area The system variable indicates if a protection area can currently be part of the collision monitoring. The individual conditions which have to be fulfilled so that a protection area can actively prevent collisions are also shown. The variable is coded as follows: Bit 0: The protection area is monitored (this bit has the same significance as the system variable $AN_COLL_STATE). Bit 1: The protection area is included in the internally mapped model. Bit 2: The protection area has the status 'P' (PLC-controlled). Bit 3: The protection area has the status 'A' (active). Bit 4: All axes which can move the protection area are referenced. Bit 5: Indicates whether a PLC bit is assigned to the protection area. Bit 6: Status of the interface bit assigned to the SB. An active protection area (bit 0 = TRUE) only then enters real collision monitoring when it is part of at least one collision pair ($NP_COLL_PAIR). The other partner must also be an active protection area.		

Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access	

aStopesi (N, S)	$A_STOPESI			
Current Safety Integrated stop Current Safety Integrated Stop E for some axis				
Return value	Value 0: No Stop E Value not equal to 0: A Stop E is currently applied for some axis			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

aXfaultsi (N, S)		$A_XFAULTSI		
Information about Stop F Information about Stop F for a safety axis:				
Return value	Bit 0 = 1: An actual value error has been discovered in the cross-check between NCK and the drive of any safety axis. Bit 1 = 1: An error has been discovered in the cross-check between NCK and the drive of any axis, and the waiting time until Stop B is triggered is running or has expired in this axis ($MA_SAFE_STOP_SWITCH_TIME_F)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

accIndex (N, S)				
Global uplaoad starting point for ACC entries Global upload starting point for ACC entries. If a value is set here, upload access to _N_xx_yyy_ACC modules starts from this entry.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	1		
Validity:	From NCK version 3.6		Read access and write access	

anAxCtAS (N, S)		$AN_AXCTAS[n]		
Container actual address Current container rotation, i.e. by how many slots the axis container has been currently advanced. The original container assignment is valid after Power On and outputs value 0.				
Return value	maxCount = Maximum number of assigned locations in the axis container - 1			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$maxnumContainer-Slots - 1
Addressing				
Line index:			Max. line index:	
			$$numContainer	
			Read access	

anAxCtSwA (N, S)		$AN_AXCTSWA[CTn]		
Axis container rotation A rotation is currently being executed on the axis container.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$numContainer	
			Read access	

anAxEsrTrigger (N, S)		$AN_ESR_TRIGGER		
ESR start signal (Global) control signal "Start Stop/Retract". With a signal edge change from 0 to 1, the reactions parameterized beforehand in axial MD $MA_ESR_REACTION and enabled via system variable $AA_ESR_ENABLE are started. Independent drive reactions subsequently require a Power-Off / Power-On, independent NC reactions require at least an opposite edge change in the relevant system variable as well as a Reset.				
Return value	0: FALSE 1: TRUE			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anCollState (N, S)		$AN_COLL_STATE[i]		
Activation state of a protection zone The system variable indicates if a protection area can currently be part of collision monitoring. However, the following requirements must be met first: 1. The activation status of the protection area is active ("A") or the activation status is PLC-controlled ("P") and the interface bit assigned to the protection area is set. 2. The protection area group ("Machine", "TOOL" etc.) has been activated in the current operating mode via the associated interface bit. A protection area for which this system variable gives the value TRUE only then enters real collision monitoring when it is part of at least one collision pair ($NP_COLL_PAIR). The other partner must also be an active protection area.				
Units and value ranges				
Physical unit:	Data type:			

anCollState (N, S)		$AN_COLL_STATE[i]		
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$MN_MM_MAXNUM_3D_PROT_AREAS	
			Read access	

anCollIpoActive (N, S)		$AN_COLL_IPO_ACTIVE		
Status main run monitoring collision avoidance The system variable indicates if the main run monitoring of the collision avoidance is active.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

anCollIpoLimit (N, S)		$AN_COLL_IPO_LIMIT		
Velocity reduction collision monitoring The system variable indicates if the main run monitoring of the collision avoidance leads to a velocity reduction.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

anCollLoad (N, S)		$AN_COLL_LOAD		
Calcul. time requir. for collision avoidance in ms Gives the required calculation time in ms - required for certain operations in connection with collision avoidance. The operation is defined by index i. i = 0: Time requirement for last call of PROTA i = 1: Time requirement for last call of collision avoidance during preprocessing i = 2: Time requirement for last call of the calculation of free space (real-time monitoring) The variables can be reset by describing with value 0. Every write attempt with a value other than 0 is refused with an error message.				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			3	
			Read access and write access	

anIpoActLoad (N, S)	$AN_IPO_ACT_LOAD	
Current IPO runtime Current IPO runtime including the runtime of the synchronized actions of all channels		

anIpoActLoad (N, S)		$AN_IPO_ACT_LOAD		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anIpoLoadLimit (N, S)		$AN_IPO_LOAD_LIMIT		
IPO utilization limit reached				
Return value	0: Utilization limit not reached 1: Utilization limit reached			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anIpoLoadPercent (N, S)		$AN_IPO_LOAD_PERCENT		
Ratio of curr. IPO runtime / IPO cycle				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anIpoMaxLoad (N, S)		$AN_IPO_MAX_LOAD		
Maximum IPO runtime Maximum IPO runtime including the runtime of the synchronized actions of all channels				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anIpoMinLoad (N, S)		$AN_IPO_MIN_LOAD		
Minimum IPO runtime Minimum IPO runtime including the runtime of the synchronized actions of all channels				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anLinkCommState (N, S)		$AN_LINK_COMM_STATE		
Status of the NCU link communication Status of the NCU link communication between all NCUs in the NCU link cluster. Decimal values of the variable: 0: NCU link communication is not active (MD18780 $MN_MM_NCU_LINK_MASK) 1: NCU link communication is active (MD18780 $MN_MM_NCU_LINK_MASK) and functions correctly, that is signs of life are received from all NCUs in the cluster 2: NCU link communication is active (MD18780 $MN_MM_NCU_LINK_MASK), but does not function correctly (e.g. start-up with inactive link, communication error etc.)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anLinkConnRcv (N, S)		$AN_LINK_CONN_RCV		
Number of link variable changes Number of link variable changes per cycle from the specified to the current NCU number. The variable $AN_LINK_CONN_RCV[NCU-No] shows the transmission capacity reserved for non-cyclic messages from NCU-No to NCU-Curr in bytes. Systems without an NCU link return the value 0. Note: This value is only available in the IPO, and therefore cannot be read via the OPI variable service for the display.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	

anLinkConnRcv (N, S)	$AN_LINK_CONN_RCV	
	$$maxNumNcusInNcuCluster	
	Read access	

anLinkConnSizeLinkvar (N, S)	$AN_LINK_CONN_SIZE_LINKVAR			
Gross number of bytes required				
Gross number of bytes required for a link variable to be transmitted in each PTP relationship The assignment of a link variable (e.g. $a_dlb[9] = 1) loads the non-cyclic link connections with a message of length $AN_LINK_CONN_SIZE_LINKVAR. It is irrelevant here whether a double-link or a byte-link variable is written. This enables the customer to estimate the maximum number of transmittable link variables per IPO cycle ($AN_LINK_CONN_SND[NCU-No] / $AN_LINK_CONN_SIZE_LINKVAR= number of link-variable changes per IPO cycle from NCU-Curr to NCU-No). Note: This value is only available in the IPO, and therefore cannot be read via the OPI variable service for the display.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anLinkTransRateLast (N, S)	$AN_LINK_TRANS_RATE_LAST			
Unused bandwidth of link variables				
Number of link variables that should have been sent off in a previous IPO cycle. Note: This value is only available in the IPO, and therefore cannot be read via the OPI variable service for the display.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anLinkTransRateLastSum (N, S)	$AN_LINK_TRANS_RATE_LAST_SUM			
Free link variables of the specif. send direction				
Number of link variables for the send direction to the specified NCU number that could have been sent off in a previous IPO cycle. Note: This value is only available in the IPO, and therefore cannot be read via the OPI variable service for the display.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		

anLinkTransRateLastSum (N, S)	$AN_LINK_TRANS_RATE_LAST_SUM	
Addressing		
Line index:	Max. line index:	
	$$maxNumNcusInNcuCluster	
	Read access	

anLinkConnSnd (N, S)	$AN_LINK_CONN_SND	
Number of link variable changes per cycle Number of link variable changes per cycle from the current to the specified NCU number. The index NCU number of the variable $AN_LINK_CONN_SDN[NCU-No] currently ranges from 1 to 16. The variable returns the number of bytes reserved from the current NCU-Curr to the NCU-No in order to exchange any non-cyclic messages. Depending on the utilization of this transmission capacity, SIEMENS can supply new SDB blocks for the CBE-30 which reduce the total transmission capacity from NCU-Curr to NCU-No. This makes the link faster and thus the servo cycle shorter. Note: If NCU-Curr == NCU-No, the variable returns "0". Note: This value is only available in the IPO, and therefore cannot be read via the OPI variable service for the display.		

Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$maxNumNcusInNcuCluster	
			Read access	

anRebootDelayTime (N, S)	$AN_REBOOT_DELAY_TIME		
Time until reboot			
Units and value ranges			
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anServoActLoad (N, S)	$AN_SERVO_ACT_LOAD		
Current runtime of the position controller			
Units and value ranges			
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	

anServoActLoad (N, S)	$AN_SERVO_ACT_LOAD	
	1	
	Read access	

anServoMaxLoad (N, S)		$AN_SERVO_MAX_LOAD		
Maximum runtime of the position controller				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anServoMinLoad (N, S)		$AN_SERVO_MIN_LOAD		
Minimum runtime of the position controller				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anSimChanMask (N, S)		$AN_SIM_CHAN_MASK		
Channels for the synchronized simulation Bit-coded mask of channels to be considered in the synchronized multi-channel simulation. The variable is valid only in connection with the synchronized simulation (see Bit4 $MN_PROG_TEST_MASK).				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	0x3FF
			Read access and write access	

anSimMaxIpoStep (N, S)		$AN_SIM_MAX_IPOSTEP		
Maximum step for the synchronized simulation Using this variable you can specify the maximum step width in real-time IPO cycles. After each step width an event is output to the HMI interface. This allows setting the number of breakpoints. If value 0 is specified, then the system determines the maximum possible step width. The variable is valid only in connection with the synchronized simulation (see Bit4 $MN_PROG_TEST_MASK).				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	

anSimMaxIpoStep (N, S)		$AN_SIM_MAX_IPOSTEP		
-	TYPE_DWORD	0	0	
			Read access and write access	

anSyncActLoad (N, S)		$AN_SYNC_ACT_LOAD		
Current runtime for synchronized actions				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anSyncMaxLoad (N, S)		$AN_SYNC_MAX_LOAD		
Maximum runtime for synchronized actions				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anSyncToIpo (N, S)		$AN_SYNC_TO_IPO		
Percentage of Synact / IPO computing time				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anTimer (N, S)		$AN_TIMER[n]		
NCK timer Global NCK timer in seconds.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_AN_TIMER	
			Read access	

analogInpVal (N, S)		$A_INA[x] x = AnaloginputNo		
Value of HW analog input				
Units and value ranges				
Physical unit:	Data type:			
A or V	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numAnalogInp	
Validity:	From NCK version 3.6		Read access	

analogOutpVal (N, S)		$A_OUTA[x] x = AnalogoutputNo		
Number of HW analog output				
Units and value ranges				
Physical unit:	Data type:			
A or V	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numAnalogOutp	
Validity:	From NCK version 3.6		Read access and write access	

axisActivInNcu (N, S)				
Display of active axis Display indicating whether the axis is active, i.e. whether it can be traversed via a channel of its own NCU or via another NCU (link axis). This data can be utilized by HMIs in order to suppress the display of any non-active axes.				
Return value	Bits 0-31 stand for the axes of the NCU. Bit n = 1: Axis can be traversed. Bit n = 0: Axis cannot be traversed.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

badMemFfs (N, S)				
FFS memory defect Only with 840D-powerline: Number of bytes which are defective in the Flash File System (FFS)				
Units and value ranges				
Physical unit:	Data type:	Initial value		

badMemFfs (N, S)				
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

basisFrameMask (N, S)		$P_NCBFRMASK		
Active channel-independent basic frames Display indicating which channel-independent basic frames are active.				
Return value	Every bit in the mask indicates whether the appropriate basic frame is active. Bit0 = 1st basic frame, Bit1 = 2nd basic frame, etc.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

completeDpcAcxChangeCnt (N, S)				
Other counter for _N_COMPLETE_DPC_ACX Modification counter of ACX for the PROFIBUS configuration of all PROFIBUS segments (_N_COMPLETE_DPC_ACX) that is incremented when ACX is changed. If the contents of ACX is or becomes invalid, the modification counter will be set to 0. If the contents of ACX is valid again, the modification counter will be reset to the value it had before the contents of ACX became invalid, and will simultaneously be incremented (only the value), if the contents of ACX has really changed.				
Return value	== 0: Contents of _N_COMPLETE_DPC_ACX is invalid != 0: Contents of _N_COMPLETE_DPC_ACX is valid			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

completeDocAcxChangeCnt (N, S)				
Other counter for _N_COMPLETE_DOC_ACX Modification counter of ACX for the configuration of DO of all SINAMICS on all PROFIBUS segments (_N_COMPLETE_DOC_ACX) that is incremented when the ACX is changed. If the contents of ACX is or becomes invalid, the modification counter will be set to 0. If the contents of ACX is valid again, the modification counter will be reset to the value it had before the contents of ACX became invalid, and will simultaneously be incremented (only the value), if the contents of ACX has really changed.				
Return value	== 0: Contents of _N_COMPLETE_DOC_ACX is invalid != 0: Contents of _N_COMPLETE_DOC_ACX is valid			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

completeDotAcxChangeCnt (N, S)				
Other counter for _N_COMPLETE_DOT_ACX Modification counter of ACX that describes all SINAMICS DO types known to the OPI (_N_COMPLETE_DOT_ACX) and that is incremented when ACX changes. If the contents of ACX is or becomes invalid, the modification counter will be set to 0. If the contents of ACX is valid again, the modification counter will be reset to the value it had before the contents of ACX became invalid and will be incremented (only the value) simultaneously, if the contents of ACX has really changed.				
Return value	== 0: Contents of _N_COMPLETE_DOT_ACX is invalid != 0: Contents of _N_COMPLETE_DOT_ACX is valid			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

diagnoseDataFfs (N, S)				
FFS diagnostic data Only with 840D-powerline: Diagnostic data for Flash File System (FFS)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

digitInpVal (N, S)		$A_IN[x] x = DigitalinputNo		
Value of HW digital input				
Return value	0 = low 1 = high			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numDigitInp	
Validity:	From NCK version 3.6		Read access	

digitOutpVal (N, S)		$A_OUT[x] x = DigitaloutputNo		
Value of HW digital output				
Return value	0 = low 1 = high			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numDigitOutp	
Validity:	From NCK version 3.6		Read access and write access	

driveType (N, S)				
Digital drive type Digital drive type. Coded according to machine data 13040, but additional code. Note: As long as the OPI variable contains the identifier 0x100 "Drive type unknown" after an NCK ramp-up, the information is not yet consistent and must not be evaluated. As soon as the identifier 0x100 is deleted, in NCU systems with SIMODRIVE 611D drives it can be assumed that the content can only change after renewed link to the NCK. (e.g. after modification of the drive modules), i.e. it need not be cyclically checked for change.				
Return value	0x100: Drive type unknown. 0x200: This identifier is entered in addition to the code according to the machine data 13040 if a 611D-Performance2 module is detected. For other codes, see MD 13040.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumDrives	
			Read access	

driveTypeChangeCnt (N, S)				
Modification counter for $$driveType This counter is incremented by 1 every time $$driveType is modified. The next value after 65535 is 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeDirectorys (N, S)				
Number of directories that can be created				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

freeDirectorys (N, S)				
			1	
			Read access	

freeFiles (N, S)				
Number of files that can be created				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			4	
			Read access	

freeMem (N, S)				
Free SRAM in bytes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemDram (N, S)				
Free memory in bytes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemDramEPassF (N, S)				
Free memory of the E file system Memory in bytes available in the passive file system for executing from external drives				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemDramMPassF (N, S)				
Free memory of the M file system Free memory of the passive file system of the "Machine manufacturer" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemDramPassF (N, S)				
Available memory DRAM file system no. 1 Memory available in passive file system (DRAM no. 1) in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemDramSPassF (N, S)				
Free memory of the C file system Free memory of the passive file system of the "Control manufacturer" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemDramTPassF (N, S)				
Free memory of the T file system Free memory of the passive file system of the "Temp" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemDramUPassF (N, S)				
Alternative name:	freeMemDram2PassF			
Free memory of the User file system Free memory of the passive file system of the "User" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemFfs (N, S)				
FFS free memory Only with 840D-powerline: Number of bytes that are still available in the Flash File System (FFS)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemISram (N, S)				
Free internal SRAM				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemSettings (N, S)				
Free SRAM (for the present memory layout) Free SRAM for the part programs and persistent data for the present (possibly not yet active) memory layout. It is updated when memory-configuring machine data is changed. The size of the current file system is taken into account.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	

freeMemSettings (N, S)				
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemSettingsDram (N, S)				
Free DRAM (for the present memory layout) Free DRAM memory for data for the present (may not yet be active) memory layout. Is updated when memory-configuring machine data is changed.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemSettingsISram (N, S)				
Free internal SRAM Free internal SRAM for the current (possibly not yet active) memory layout. It is updated when the memory-configuring machine data is updated.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemSramPassF (N, S)				
Available memory SRAM file system Memory available in passive file system (SRAM) in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeProtokolFiles (N, S)				
Logging: res. no. free files Logging: Number of protocol files that can still be created				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

handWheelTestDiffPulses (N, S)				
Differential handwheel pulses Define differential handwheel pulses for handwheel simulation via OPI				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$$numHandWheels	
			Read access and write access	

handwheelStatus (N, S)				
Handwheel status				
Return value	0 = PASSIVE 1 = ACTIVE			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numHandWheels	
Validity:	From NCK version 3.6		Read access and write access	

hwMLFB (N, S)				
MLFB of the NCU module				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

hwProductSerialNr (N, S)				
Unique hardware number of the NCU Unique hardware number of the NCU module With solution line, this is the serial number of the CF card.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

hwProductSerialNrL (N, S)				
Unique hardware number of the NCU (long) Unique hardware number of the NCU module				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

licenseStatus (N, S)				
Licensing status				
Return value	0: Licensed, 1: Insufficiently licensed 2: Not licensed 3: PIN is expanded 4: PIN is OK 5: PIN entered incorrectly 6: PIN missing 7: Trial license active 8: Trial license expired			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

licensedOptions (N, S)				
Licensed Options Licensed options field				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			128	
			Read access	

licTrialPeriod (N, S)				
Number of trial periods used				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_WORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

licTrialTime (N, S)				
Remaining time of the trial periods Remaining time of the current trial period				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_WORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

licTrialPeriodMaxNum (N, S)				
Number of permissible trial periods Maximum number of permissible trial periods				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_WORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

memSize (N, S)				
Result variable for PI:_N_MMSIZE Result variable for the PI service _N_MMSIZE The change in size of the memory, which is to be expected when a memory-configuring machine data item is changed, is stored here.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

mmcCmdPrepCounter (N, S)				
Counter incremented with each call of EXTCALL Counter that is incremented with each call of EXTCALL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckAliveAndWell (N, S)	DB10, DBX104.7	A4

NCK sign-of-life

NCK sign of life.
The value is increased on each reading, so that an HMI can detect whether the NCK is still working correctly by reading this variable cyclically.
The value itself has no significance.
Cyclic result acknowledgments are also returned on this variable, even when the cyclic services are not served by the NCK because of block change time problems. This behavior is only ensured if the variable is not mixed with others in one orde. In other words, nckAliveAndWell must be integrated into a cluster as a sole variable.
As long as a cyclic reading service is set on this variable, one of the HMI-CPU ready signals is set in the PLC interface.
Which of the signals is set is specified, on the one hand by the line number and, on the other hand, at the "gloports" of the client:

The following applies to Powerline:
An HMI on the MPI communicates via the gloports 0x20-0x2f -> DB10.DBX108 Bit2 is set
An HMI on the OPI communicates via the gloports 0x10-0x1f -> DB10.DBX108 Bit3 is set
In the n:m group, the 2nd HMI identifies itself by line=2 -> DB10.DBX108 Bit1 is set

The following applies to Solutionline:
HMIs (int./ext.) communicate via the gloports 0x10-0x17 -> DB10.DBX108 Bit3 is set
Reserved for subsequent extensions: DB10.DBX108 Bit1
Reserved for subsequent extensions: DB10.DBX108 Bit2

Note: The related signal NCK-CPU-ready is in DB10,DBX104.7.

Units and value ranges

Physical unit:	Data type:			
-	TYPE_UWORD			

Addressing

Line index:		Max. line index:
		2
Validity:	From NCK version 3.6	Read access

nckMode (N, S)				
Mode in which the NCK works Mode in which the NCK works. The mode can be set with the PI _N_NCKMOD.				
Return value	Bit0: NCK works accelerated in simulation mode/DRY_RUN. This mode is currently provided only for the VNCK. Bit1: NCK is slowed down to give the simulation more computing time on the same processor. NC start not possible. Bit2: PowerOn-Init_Finish; power-up initialization finished. Meaning: State = 1 All channels of the NCK have been powered up and have interpreted the DEF files. Thus the HMI may receive consistent data from the NCK. Example of a call from HMI to NCK: What macros are available? What GUDs are available? Note: With Init-Finish==1, the POWER-ON Progevent has NOT yet been run. State = 0 Power up has not yet been finished, or initialization could not been executed due to a fatal alarm Bit3: PowerOn-Ready ; power-up finished Meaning: State == 1 NCK has finished initialization, AND the POWER-ON Progevent has been executed. OR the POWER-ON Progevent could not be executed due to an alarm. Note: The next RESET will "catch up" on the POWER-ON Progevent. This will no longer influence PowerOn-Ready. Without PowerOn Progevent bit 3 and bit 2 are the same. Bit4: NCK works accelerated in simulation mode. This mode is not currently provided for the VNCK.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	f
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckModeAccFact (N, S)				
NCK acceleration factor NCK acceleration factor. The acceleration factor can be set with the PI _N_NCKMOD. The NCK executes the programs in SERUPRO mode. This mode is currently provided only for the VNCK.				
Return value	0 VNCK executes a program at normal speed. >0 VNCK executes a program at accelerated speed. $$nckModeAccFact defines the acceleration factor.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

ncuLinkActive (N, S)				
NCU link active Display indicating whether NCU link is activated (via machine data setting) Based on display, HMI decides whether link-specific calculations and displays are required.				
Return value	0: NCU link not activated 1: NCU link activated			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nettoMemFfs (N, S)				
FFS net size Only with 840D-powerline: Net number of bytes which are available for the Flash File System (FFS). This memory stores the files contents and management data (e.g. file names).				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	

nettoMemFfs (N, S)				
			1	
			Read access	

numAlarms (N, S)				
Number of pending general alarms				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numFilesPerDir (N, S)				
Maximum number of files per directory Maximum number of files per directory (see: $MN_MM_NUM_FILES_PER_DIR)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numSubDirsPerDir (N, S)				
Maximum number of subdirectories per directory Maximum number of subdirectories per directory see: $MN_MM_NUM_SUBDIR_PER_DIR				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numTraceProtocDataList (N, S)		$MM_PROTOC_NUM_ETPD_STD_LIST		
Logging: No. std. data lists Logging: Number of standard data lists per user				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_UWORD		0	
Addressing				
Line index:			Max. line index:	

numTraceProtocDataList (N, S)	$MM_PROTOC_NUM_ETPD_STD_LIST	
	10	
	Read access	

numTraceProtocOemDataList (N, S)		$MM_PROTOC_NUM_ETPD_OEM_LIST		
Logging: No. OEM data lists Logging: Number of OEM data lists per user				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

protCnfgAutoLoad (N, S)				
Protokoll. Load trace session while NCK Start up Protokoll.: Controls automatic load from trace session from description file during the NCK Start up process				
Return value	0: Function inactive 1: Logging session should be loaded automatically from a description file during NCK ramp-up 2: Same as (1), but automatic loading is terminated by stop trigger 3: Same as (1), but the description file is updated at each status change			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protCnfgAutoLoadFile (N, S)				
Log: Load file for trace session Log: Directory path and file name from which the trace session is to be loaded during NCK start-up				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protCnfgAutoSave (N, S)				
Log: Configuration of session backup Log: Configuration of the backup of the trace session				
Return value	0: Function inactive 1: Automatically backs up the logging session in a description file when logging is terminated. 2: Automatically backs up the logging session and diagnostic information in a description file when logging is terminated.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protCnfgCtl (N, S)				
Log: Configuration of session backup file Log: Control word to manipulate the trace session description file				
Return value	0: Function inactive 1: Backs up the logging session in a description file 2: Backs up the logging session and diagnostic information in a description file 3: Loads the logging session from a description file, and all active triggers are deactivated 4: Loads the logging session from a description file 5: Deletes the logging file			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	5
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protCnfgFilename (N, S)				
Log: Name of session backup file				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protCnfgStat (N, S)				
Log: Status of session backup/load Log: Result of the most recent backup or loading of a session backup file				
Return value	0: No Error			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protSessAccR (N, S)				
Logging: Access rights of the session				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protSessComm (N, S)				
Logging: Comments on session				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protSessConn (N, S)				
Logging: Connection of the session				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protSessName (N, S)				
Logging: Name of the session				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protSessPrior (N, S)				
Logging: Priority of the session				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocLastValNetIpoCycle (N, S)				
Logging: Last user run time Logging: Runtime of all events of all channels of a user in the last IPO cycle				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

protocMaxValNetIpoCycle (N, S)				
Logging: Max. user run time Logging: Maximum run time of all events of all channels of a user				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtMaskInt16 (N, S)				
Logging: Start screen form integer 16 bit Logging: Integer 16 bit screen form with which the start trigger variable is logically AND-ed before the comparison is made with the trigger value. There is no logic operation with the value 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtMaskInt32 (N, S)				
Logging: Start screen form integer 32 bit Logging: Integer 32 bit screen form with which the start trigger variable is logically AND-ed before the comparison is made with the trigger value. There is no logic operation with the value 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtMatchCount (N, S)				
Logging: Number of start comparisons Logging: Specifies how often the comparison must match before the start trigger fires. The trigger will only fire, if all trigger conditions are fulfilled.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtNumEvDelay (N, S)				
Logging: Delay of start trigger Logging: Number of events which are still to be omitted after the occurrence of the trigger event before logging is started.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	

protocStrtNumEvDelay (N, S)				
			10	
			Read access and write access	

protocStrtRemMatchCount (N, S)				
Logging: Remaining number of start comparisons Logging: Specifies how often the comparison must match before the start trigger fires. The trigger will only fire, if all trigger conditions are fulfilled.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

protocStrtState (N, S)				
Logging: Status of start triggering Logging: Status of the start triggering				
Return value	0: Passive (trigger inactive) 1: Active (trigger is active, but has not yet responded) 2: Delay (trigger has responded and is still waiting the delay time) 3: Firing (trigger has responded, but must still respond more frequently until the triggering takes place) 4: Done (trigger has responded and is inactive)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtType (N, S)				
Logging: Type of start triggering				
Return value	0: Monitoring for equality 1: Monitoring for more than or equal to 2: Monitoring for greater than 3: Monitoring for less than or equal to 4: Monitoring for less than 5: Monitoring for inequality 6: Monitoring for value change 7: Monitoring for increasing values 8: Monitoring for falling values			

protocStrtType (N, S)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtValueInt16 (N, S)				
Logging: Start value integer 16 bit Logging: Integer 16 bit value with which the start trigger variable is to be compared				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtValueInt32 (N, S)				
Logging: Start value integer 32 bit Logging: Integer 32 bit value with which the start trigger variable is to be compared				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtValueReal32 (N, S)				
Logging: Start value real 32 bit Logging: Real 32 bit value with which the start trigger variable is to be compared				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_FLOAT	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtValueReal64 (N, S)				
Logging: Start value real 64 bit Logging: Real 64 bit value with which the start trigger variable is to be compared				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtVarCol (N, S)				
Logging: Start variable: Col Logging: Variable which is to be monitored for the start triggering. Statement of the "Col"				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtVarRow (N, S)				
Logging: Start variable: Row Logging: Variable which is to be monitored for the start triggering. Statement of the "Row"				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtVarType (N, S)				
Logging: Start variable: Type Logging: Variable which is to be monitored for the start triggering. Statement of the "Type"				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	

protocStrtVarType (N, S)				
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtVarUnit (N, S)				
Logging: Start variable: Unit Logging: Variable which is to be monitored for the start triggering. Statement of the "Unit".				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigMaskInt16 (N, S)				
Logging: Trigger mask integer 16-bit Logging: Integer 16-bit mask with which the trigger variable is logically ANDed before the comparison with the trigger value is made. Variable is not ANDed if value is 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigMaskInt32 (N, S)				
Logging: Trigger mask integer 32-bit Logging: Integer 32-bit mask with which the trigger variable is logically ANDed before the comparison with the trigger value is made. Variable is not ANDed if value is 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigOperation (N, S)				
Logging: Trigger operation Logging: There are two start trigger variables. Each of these two variables is monitored using $$protocTrigType. The result from each of these monitoring actions is: Trigger: yes/no Since there are two variables, there are also two results, which can now be combined with a logic operator. This operator is specified with this variable				
Return value	0: No logic operation, only the first variable is to be taken into consideration 1: NOT (unary, the result of the first variable is negated, no second variable 2: AND (the result of the first variable is combined with the result of the second variable with logic AND 3: OR (the result of the first variable is combined with the result of the second variable with logic OR 4: XOR (the result of the first variable is combined with the result of the second variable with logic XOR			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocStrtOperation (N, S)				
Logging: Start trigger operation Logging: There are two start trigger variables. Each of these two variables is monitored using $$protocTrigType. The result from each of these monitoring actions is: Trigger: yes/no Since there are two variables, there are also two results, which can now be combined with a logic operator. This operator is specified with this variable				
Return value	0: No logic operation, only the first variable is to be taken into consideration 1: NOT (unary, the result of the first variable is negated, no second variable 2: AND (the result of the first variable is combined with the result of the second variable with logic AND 3: OR (the result of the first variable is combined with the result of the second variable with logic OR 4: XOR (the result of the first variable is combined with the result of the second variable with logic XOR			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	

protocStrtOperation (N, S)				
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigMatchCount (N, S)				
Logging: Number of comparisons Logging: Specifies how often the comparison in total must match before the trigger fires. The trigger will only fire, if all trigger conditions are fulfilled.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigNumEvDelay (N, S)				
Logging: Delay after trigger Logging: Number of events to be recorded after the trigger event has occurred before the logging operation is stopped.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigRemMatchCount (N, S)				
Logging: Remaining number of comparisons Logging: Specifies how often the comparison must match before the trigger fires. The trigger will only fire, if all trigger conditions are fulfilled.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

protocTrigState (N, S)				
Logging: Triggering status				
Return value	0: Passive (trigger inactive) 1: Active (trigger is active, but has not yet responded) 2: Delay (trigger has responded and is still waiting the delay time) 3: Firing (trigger has responded, but must still respond more frequently until the triggering takes place) 4: Done (trigger has responded and is inactive)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigType (N, S)				
Logging: Triggering method				
Return value	0: Monitoring for equality 1: Monitoring for more than or equal to 2: Monitoring for greater than 3: Monitoring for less than or equal to 4: Monitoring for less than 5: Monitoring for inequality 6: Monitoring for value change 7: Monitoring for increasing values 8: Monitoring for falling values			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigValueInt16 (N, S)				
Logging: Trigger value integer 16-bit Logging: Integer 16-bit value with which trigger variable must be compared				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	

protocTrigValueInt16 (N, S)				
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigValueInt32 (N, S)				
Logging: Trigger value integer 32-bit Logging: Integer 32-bit value with which trigger variable must be compared				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigValueReal32 (N, S)				
Logging: Trigger value real 32-bit Logging: Real 32-bit value with which trigger variable must be compared				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_FLOAT	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigValueReal64 (N, S)				
Logging: Trigger value real 64-bit Logging: Real 64-bit value with which trigger variable must be compared				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigVarArea (N, S)				
Logging: Trigger variable: Area Logging: Variable which is to be monitored for the start triggering. Specification of "Area".				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigVarCol (N, S)				
Logging: Trigger variable: Col Logging: Variable to be monitored for triggering. Specification of "Col".				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigVarRow (N, S)				
Logging: Trigger variable: Row Logging: Variable to be monitored for triggering. Specification of "Row".				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigVarType (N, S)				
Logging: Trigger variable: Type Logging: Variable to be monitored for triggering. Specification of "Type".				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	

protocTrigVarType (N, S)				
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

protocTrigVarUnit (N, S)				
Logging: Trigger variable: Unit Logging: Variable to be monitored for triggering. Specification of "Unit".				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

safeExtInpValNckBit (N, S)		$A_INSE[n]		
External NC input of SI prog. log. from NCK peri. External NC input of the SI programmable logic from the NCK periphery				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumExtInput	
			Read access	

safeExtInpValNckWord (N, S)		$A_INSED[n]		
Image of external NC inputs of SI progr. logic Image of the external NC inputs of the SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumExtInput / 32	
			Read access	

safeExtInpValPlcBit (N, S)	$A_INSEP[n]	
External PLC input of SI prog. log. from PLC peri. External PLC input of the SI programmable logic from the PLC periphery		

safeExtInpValPlcBit (N, S)		$A_INSEP[n]		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumExtInput	
			Read access	

safeExtInpValPlcWord (N, S)		$A_INSEPD[n]		
Image of the external PLC inputs of SI progr. log. Image of the external PLC inputs of the SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumExtInput / 32	
			Read access	

safeExtOutpValNckBit (N, S)		$A_OUTSE[n]		
External NC output of SI prog. log. to NCK per. External NC output of the SI programmable logic to the NCK periphery				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumExtOutput	
			Read access	

safeExtOutpValNckWord (N, S)		$A_OUTSED[n]		
Image of the external NC outputs of SI prog. log. Image of the external NC outputs of the SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumExtOutput / 32	
			Read access	

safeExtOutpValPlcBit (N, S)		$A_OUTSEP[n]		
External PLC output of the SI progr. logic External PLC output of the SI programmable logic to the PLC periphery				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumExtOutput	
			Read access	

safeExtOutpValPlcWord (N, S)		$A_OUTSEPD[n]		
Image of the external PLC outputs of SI prog. log. Image of the external PLC outputs of the SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumExtOutput / 32	
			Read access	

safeIntInpValNckBit (N, S)		$A_INSI[n]		
Internal NC input of SI progr. logic Internal NC input of the SI programmable logic from the NCK's SI monitoring channel				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumIntInput	
			Read access	

safeIntInpValNckWord (N, S)		$A_INSID[n]		
Image of int. NC inputs of SI progr. logic Image of the internal NC inputs of the SI programmable logic from the NCK's SI monitoring channel				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumIntInput / 32	
			Read access	

safeIntInpValPlcBit (N, S)		$A_INSIP[n]		
Internal PLC input of SI progr. logic Internal PLC input of the SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumIntInput	
			Read access	

safeIntInpValPlcWord (N, S)		$A_INSIPD[n]		
Image of PLC inputs of SI progr. logic Image of the internal PLC inputs of the SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumIntInput / 32	
			Read access	

safeIntOutpValNckBit (N, S)		$A_OUTSI[n]		
internal NC output of the SI progr. logic Internal NC output of the SI programmable logic to the NCK's SI monitoring channel				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumIntOutput	
			Read access	

safeIntOutpValNckWord (N, S)		$A_OUTSID[n]		
Image of int. NC outputs of the SI progr. logic Image of the internal NC outputs of the SI programmable logic to the NCK's SI monitoring channel				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumIntOutput / 32	
			Read access	

safeIntOutpValPlcBit (N, S)		$A_OUTSIP[n]		
Internal PLC output of SI progr. logic Internal PLC output of the SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumIntOutput	
			Read access	

safeIntOutpValPlcWord (N, S)		$A_OUTSIPD[n]		
Image of int. PLC outputs of the SI progr. logic Image of the internal PLC outputs of the SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumIntOutput / 32	
			Read access	

safeMarkerNck (N, S)		$A_MARKERSI[n]		
NCK flag for the SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumMarker	
			Read access	

safeMarkerPlc (N, S)		$A_MARKERSIP[n]		
Image of the PLC flag-variable for SI prog. logic Image of the PLC flag-variable for SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumMarker	
			Read access	

safePlcIn (N, S)		$A_PLCSIIN[index]		
Single channel SI signal from PLC to NCK Bit image of the single channel safety signals from PLC to NCK				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumPlcInOut	
			Read access	

safePlcOut (N, S)		$A_PLCSIOUT[index]		
Single channel SI signal from NCK to PLC Bit image of the single channel safety signals from NCK to PLC				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumPlcInOut	
			Read access	

safeSplStatus (N, S)			
SPL booting status Status of components and parameter settings required for operation of Safe Programmable Logic			
Return value	Bit 0: SPL interfaces $A_INSE, $A_OUTSE, $A_INSI or $A_OUTSI have been parameterized Bit 1: SPL program file SAFE.SPF loaded Bit 2: NCK is waiting for PLC to power up Bit 3: PLC is in cyclic operating mode. PLC can communicate with drive. Bit 4: Interrupt for ASUB start of SPL must be assigned (FB4 call started) Bit 5: Interrupt for ASUB start of SPL has been assigned (FB4 call ended) Bit 6: Interrupt processing for SPL start called (FC9 call started) Bit 7: Interrupt processing for SPL start ended (FC9 call ended) Bit 8: SPL start by call from PROG_EVENT file Bit 9: NCK data cross-checking has been started Bit10: PLC data cross-checking has been started Bit11: Cyclic SPL checksum check active Bit12: All SPL protective mechanisms active Bit13: SPL program execution ended Bit14: SPL start by PowerOn safety event		

safeSplStatus (N, S)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeTimerNck (N, S)		$A_TIMERSI		
NCK timer variable for the SI programmable logic NCK timer-variable for the SI programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

safeXcmpCmd (N, S)		$A_CMDSI[index]		
Command word SPL-KDV Command word for cross-checking (KDV) between NCK and PLC				
Return value	0:No command 1:Extension of time window for different signal levels in cross-checking operation between NCK and PLC			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			32	
			Read access	

safeXcmpLevel (N, S)		$A_LEVELSID		
Fill-level display SPL-KDV Fill-level display for cross-checking operation (KDV) between NCK and PLC. Specifies the current number of signals of different levels between the NCK and PLC)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	

safeXcmpLevel (N, S)		$A_LEVELSID		
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeXcmpState (N, S)		$A_STATSID		
Error SPL-KDV Cross-checking (KDV) error has occurred between NCK and PLC.				
Return value	0: No error has occurred			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

scalingSystemCounter (N, S)				
Modification counter for dimension system Measuring system change counter (begins at 1 after a restart)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

semaDataAvailable (N, S)				
SEMA data accessible Display indicating whether complete SEMA data are available for individual axes. This is the case if a channel can be assigned to the relevant NCU axis, thus allowing the data in the channel context to be accessed. This does not apply to link axes as these are traversed by a channel of another NCU. This data can be utilized by HMIs in order to conceal specific, inaccessible data in link axis data displays.				
Return value	Bits 0-31 stand for the axes of the NCU. Bit n = 1: Data can be accessed easily Bit n = 0: Not all SEMA data are accessible			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

swLicensePIN (N, S)				
PIN for licensing				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

kryptoKeyInfo (N, S)				
Number of the crypto-key Number of the crypto-key reserved for internal applications				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

sysTimeBCD (N, S)				
Time Time represented in PLC format: <month>.<day>.<year> <hours>:<minutes>:<seconds>.<milliseconds> <weekday> <status> <weekday> can take following values: "SUN", "MON", "TUE", "WED", "THU", "FRI", "SAT"				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DATETIME			
Validity:	From NCK version 3.6		Read access	

sysTimeUdword (N, S)				
Time in special data format Time $$sysTimeBCD in special data format: 6 bits for second (bits of the lowest priority) 6 bits for minute 5 bits for hour 5 bits for day 4 bits for month 6 bits for the last two digits of the year This coding is suitable for using the absolute time as trigger for logging. Also see: $$protocStrtValueInt32 and $$protocTrigValueInt32				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
			Read access	

sysTimeNCSC (N, S)				
NCSC system time NCSC system time in microseconds				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
µs	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

sysTimeNCSCatTraceStart (N, S)				
Trace start time Logging: NCSC time stamp for the trace start time in µs				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
µs	TYPE_DWORD	0	0	

sysTimeNCSCatTraceStart (N, S)				
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

sysTimeNCSCatTraceTrig (N, S)				
Trace trigger time Logging: NCSC time stamp for the trace start trigger time in µs				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
µs	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

sysTimeNCSCdiffTraceStart (N, S)				
Time difference for the trace start Logging: Time difference for the trace start time in µs				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
µs	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

sysTimeNCSCdiffTraceTrig (N, S)				
Time difference for the start trigger Logging: time difference for the trace start trigger time in µs				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
µs	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

sysTimeSinceStartup (N, S)				
Time since NCK ramp-up System run time in seconds since NCK ramp-up				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	

sysTimeSinceStartup (N, S)				
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

tlkNr (N, S)				
Unique number of the TLK Unique copy number for the temporary license key				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

tlkPIN (N, S)				
TLK Temporary license key				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

tlkStatus (N, S)				
Status of the TLK Status of the temporary license key				
Return value	0: active 1: inactive 10: incorrect input 11: max. number of incorrect inputs exceeded 200: internal error (TLK_BUFFER_TOO_SMALL)			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	1		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalDirectorys (N, S)				
Max. number of directories which may be created Maximum number of directories which may be created see: $MN_MM_NUM_DIR_IN_FILESYSTEM				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalFiles (N, S)				
Max. number of files which may be created Maximum number of files which may be created (see: $MM_NUM_FILES_IN_FILESYSTEM)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			4	
			Read access	

totalMem (N, S)				S7
Total user memory in bytes Total SRAM in bytes (user memory)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

totalMemDram (N, S)				
total DRAM in bytes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalMemDramEPassF (N, S)				
Size of the E file system Size in bytes of the passive file system for executing from external drives				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalMemDramMPassF (N, S)				
Size of the M file system Size of the passive file system of the "Machine manufacturer" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalMemDramPassF (N, S)				
Size of DRAM file system no. 1 Size of passive file system (DRAM No. 1) in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalMemDramSPassF (N, S)				
Size of the C file system Size of the passive file system of the "Control manufacturer" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalMemDramTPassF (N, S)				
Size of the T file system Size of the passive file system of the "Temp" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalMemDramUPassF (N, S)				
Alternative name:	totalMemDram2PassF			
Size of the User file system Size of the passive file system of the "User" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalMemFfs (N, S)				
FFS total size Only with 840D-powerline: Number of bytes reserved on the PCMCIA card for the Flash File System (FFS)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalMemISram (N, S)				
Total internal SRAM in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	

totalMemISram (N, S)				
			1	
			Read access	

totalMemSramPassF (N, S)				
Size of SRAM file system Size of passive file system (SRAM) in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

totalProtokolFiles (N, S)		$MM_PROTOC_NUM_FILES		
Logging: Max. no. of files Logging: Maximum number of log files which may be created				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

traceProtocolActive (N, S)		$A_PROTOC		
Logging: User status				
Return value	0 = Not active 1 = Active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

traceProtocolLock (N, S)		$A_PROT_LOCK		
Logging: Disable of a user Logging: Recording disable of a user				
Return value	0: No disable 1: Disable 2: Disable, events from $$protocHmiEvent, however, are enabled			

traceProtocolLock (N, S)		$A_PROT_LOCK		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

traceStopAction (N, S)				
Logging: End actions Logging: Actions on ending the recording				
Return value	Bit0: Automatic restart after the stop trigger has ended the trace Bit1: Reserved Bit2: The parameterization of the session is stored in an ACX file The file name is formed from the log file with the extension "_U00_ACX". Bit3: The parameterization of the session, including diagnostic data, is stored in an ACX file The file name is formed from the log file with the extension "_U00_ACX".			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

usedDirectorys (N, S)				
No.of directories that have already been created Number of directories that have already been created				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedFiles (N, S)				
Number of files that have already been created				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

usedFiles (N, S)				
			4	
			Read access	

usedMem (N, S)				S7
Used memory in bytes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

usedMemDram (N, S)				
Used DRAM in bytes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedMemDramEPassF (N, S)				
Assigned memory of the E file system Memory in bytes occupied by the passive file system for executing from external drives				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedMemDramMPassF (N, S)				
Occupied memory of the M file system Occupied memory of the passive file system of the "Machine manufacturer" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedMemDramPassF (N, S)				
Memory used DRAM file system no. 1 Memory used in passive file system (DRAM No. 1) in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedMemDramSPassF (N, S)				
Occupied memory of the S file system Occupied memory of the passive file system of the "Control manufacturer" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedMemDramTPassF (N, S)				
Occupied memory of the T file system Occupied memory of the passive file system of the "Temp" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedMemDramUPassF (N, S)				
Alternative name:	usedMemDram2PassF			
Occupied memory of the User file system Occupied memory of the passive file system of the "User" area in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	

usedMemDramUPassF (N, S)				
			1	
			Read access	

usedMemFfs (N, S)				
FFS used memory Only with 840D-powerline: Number of bytes assigned in the Flash File System (FFS)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedMemISram (N, S)				
Occupied internal SRAM				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedMemSramPassF (N, S)				
Memory used in SRAM file system Memory used in passive file system (SRAM) in bytes				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedOptionsNotLicensed (N, S)				
Non-licensed options List of options which are not licensed				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

usedOptionsNotLicensed (N, S)				
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

usedProtokolFiles (N, S)				
Logging: No. of files created Logging: Number of protocol files that have already been created				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			10	
			Read access	

vaDpActTel (N, S)		$VA_DP_ACT_TEL[n, Achse]		
PROFIBUS actual value message frame Word for word image of the PROFIBUS actual value message frames from drives on the PROFIBUS/PROFIdrive				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			100 * $$numMachAxes + 19	
			Read access	

safeMarkerNckWord (N, S)		$A_MARKERSID[n]		
NCK flag words for the safe programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumMarker / 32	
			Read access	

safeMarkerPlcWord (N, S)		$A_MARKERSIPD[n]		
Image of PLC flag words safe programmable logic Image of the PLC flag words for the safe programmable logic				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	

safeMarkerPlcWord (N, S)		$A_MARKERSIPD[n]		
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumMarker / 32	
			Read access	

passFChangeCounter (N, S)				
Modification counter for file system Counter is incremented by 1 when there is a change in the passive file system (but not for changes to the FFS)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

simo611dSupport (N, S)				
611D support This data specifies the extent to which the system supports 611 drives.				
Return value	Bit 0 set: NCK software supports 611D drives Bit 1 set: Hardware supports 611D drives (only if bit 0 is also set).			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemWarrant (N, S)				
Guaranteed memory (SRAM) Guaranteed free memory for part programs and persistent data (from catalog NC60)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

freeMemWarrantDram (N, S)				
Guaranteed memory (DRAM)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

protocStrtVarArea (N, S)				
Logging: Start trigger variable: Area Logging: Variable which is to be monitored for start triggering.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			10	
			Read access and write access	

checkSumForAcxData (N, S)				
Checksum for ACX data Current fingerprint of selected ACX data, in order to check relatively quickly whether ACX data has changed. The data required for the OPI access is located at the start of the downloaded ACX file.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			32	
			Read access	

anVModelStatus (N, S)		$AN_VMODEL_STATUS		
Status of the VRML model System variable for the status of the VRML model				
Return value	1: MODIFIED_STATE: The model has been modified internally This is the original status. It may change, if machine parameters such as protection zones change. 2: COPIED_STATE: This status is generated outside the NCK, if the model file is ready for being displayed. 3: DISPLAYED_STATE: If the NCK sends the instruction for display of the model in the display program.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	1	1	3
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

licenseKeyInputCount (N, S)				
Remaining number of license key inputs Statement of how often the license key can still be entered				
Return value	> 0: LicenseKey can still be entered x times = 0: LicenseKey can no longer be entered. An NCK Power On is required before a new entry can be made.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	3	0	3
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

stopCondNumNck (N, S)				
Number of active stop states in the NCK Number of active stop states in the NCK Specifies the number of occupied lines in $$stopCond				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

stopCondChangeCounter (N, S)				
Modification counter for stop states in the NCK Modification counter for stop states in the NCK. This is incremented as soon as one of the stop states has changed.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

stopCond (N, S)				
Stop state number of the NC in the NCK Number of the NC stop state in the NCK More than one stop state can be active simultaneously. The highest priority stop state appears below the first line, this is followed by those with lower priorities. The documentation explains the meanings of the individual stop states.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$stopCondNumNck	
			Read access	

stopCondTime (N, S)				
BCD time stamp for stop state in the NCK BCD time stamp for stop state in the NCK. More than one stop state can be active simultaneously. The highest priority stop state appears below the first line, this is followed by those with lower priorities.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DATETIME			
Addressing				
Line index:			Max. line index:	
			$$stopCondNumNck	
			Read access	

stopCondChan (N, S)				
Stop state channel Channel in which the NC stop state was reported More than one stop state can be active simultaneously. The highest priority stop state appears below the first line, this is followed by those with lower priorities. The documentation explains the meanings of the individual stop states.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	1	$$maxnumChannels
Addressing				
Line index:			Max. line index:	
			$$stopCondNumNck	
			Read access	

stopCondPar (N, S)				
Stop state parameters in the NCK Stop state parameters in the NCK. More than one stop state can be active simultaneously. The highest priority stop state appears below the first line, this is followed by those with lower priorities.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

stopCondParA (N, S)				
ASCII stop state parameter(s) Stop state parameters in the NCK. More than one stop state can be active simultaneously. The highest priority stop state appears below the first line, this is followed by those with lower priorities.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access	

anCollMemAvailable (N, S)		$AN_COLL_MEM_AVAILABLE		
Memory available for collision monitoring in kb Collision calculation requires internal memory, the size of which is either calculated automatically from the number of available protection zones, protection zone elements, facets and the number of machine axes, or it can be explicitly defined by machine data $MN_MM_MAXNUM_3D_COLLISION. The size of the reserved memory area in kbytes can be read with system variable $AN_COLL_MEM_AVAILABLE.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	

anCollMemAvailable (N, S)	$AN_COLL_MEM_AVAILABLE	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

anCollMemUseMin (N, S)	$AN_COLL_MEM_USE_MIN			
Min. percent. of memory req. for coll. monitoring				
Collision calculation requires internal memory, the size of which is either calculated automatically from the number of available protection zones, protection zone elements, facets and the number of machine axes, or it can be explicitly defined by machine data $MN_MM_MAXNUM_3D_COLLISION. The size of the reserved memory area in kbytes can be read with system variable $AN_COLL_MEM_AVAILABLE. The system variable $AN_COLL_MEM_USE_MIN returns the minimum memory space required for collision calculation as a percentage of the reserved memory area. It can be reset by writing with the value 0. Any attempt to write any other value than 0 is rejected with an error message.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

anCollMemUseMax (N, S)	$AN_COLL_MEM_USE_MAX			
Max. percent. of memory req. for coll. monitoring				
Collision calculation requires internal memory, the size of which is either calculated automatically from the number of available protection zones, protection zone elements, facets and the number of machine axes, or it can be explicitly defined by machine data $MN_MM_MAXNUM_3D_COLLISION. The size of the reserved memory area in kbytes can be read with system variable $AN_COLL_MEM_AVAILABLE. The system variable $AN_COLL_MEM_USE_MAX returns the maximum memory space required for collision calculation as a percentage of the reserved memory area. It can be reset by writing with the value 0. Any attempt to write any other value than 0 is rejected with an error message.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

anCollMemUseAct (N, S)	$AN_COLL_MEM_USE_ACT			
Curr. percent. memory req. f. collision monitoring Collision calculation requires internal memory, the size of which is either calculated automatically from the number of available protection zones, protection zone elements, facets and the number of machine axes, or it can be explicitly defined by machine data $MN_MM_MAXNUM_3D_COLLISION. The size of the reserved memory area in kbytes can be read with system variable $AN_COLL_MEM_AVAILABLE. The system variable $AN_COLL_MEM_USE_ACT returns the current (that is the last calculated) memory space required for collision calculation as a percentage of the reserved memory area. It can be reset by writing with the value 0. Any attempt to write any other value than 0 is rejected with an error message.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

fsInfoUsed (N, S)				
Assignment of file system info object Assignment of file system info object. The client first searches for a free info object by reading the list fsInfoUsed. This is assigned by writing fsInfoUsed with 1. If there is no error in the assignment, the desired file or directory can be selected by writing the fsInfoPathName. The information about this object can then be read from the other variables.				
Return value	0: Info object free. It must be explicitly released by writing 0. 1: Info object assigned. If an assigned object is assigned again then this is acknowledged negatively.			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	1
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access and write access	

fsInfoPathName (N, S)				
Name of file or directory Name of a file or directory to be observed				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

fsInfoPathName (N, S)				
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access and write access	

fsInfoObjStatus (N, S)				
Status of $$fsInfoPathName Status of file system object $$fsInfoPathName.				
Return value	Bit-coded (may be added to later): Bit0 = 0: Object not loaded in the NCK. Bit0 = 1: Object loaded in the NCK. Bit1 = 0: Object is a file. Bit1 = 1: Object is a directory			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	3
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access	

fsInfoFileLength (N, S)				
Length of $$fsInfoPathName Length of the file system object $$fsInfoPathName.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access	

fsInfoChangeDateTime (N, S)				
Change time of $$fsInfoPathName Change time of the file system object $$fsInfoPathName.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access	

fsInfoChangeDateTimeSub (N, S)				
Change time of the files contained. Change time of the files contained in the $$fsInfoPathName directory.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access	

fsInfoPartition (N, S)				
Partition of $$fsInfoPathName Partition and lifetime of the file $$fsInfoPathName				
Return value	SRP: SRAM persistent USV: DRAM User volatile USP: DRAM User persistent SIP: DRAM Siemens persistent MAV: DRAM Manufacturer volatile MAP: DRAM Manufacturer persistent TMV: DRAM Temporary volatile D1V: DRAM System 1 volatile EXV: DRAM External volatile EXP: DRAM External persistent			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access	

fsInfoRights (N, S)				
Access protection of $$fsInfoPathName Access protection and lifetime of file $$fsInfoPathName				
Return value	0-7 ASCII-coded for read / write / execute / show / delete rights			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	"77777"		
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access	

fsInfoSeekw (N, S)				
First writable line of the file. First line enabled for modification in part program $$fsInfoPathName				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access	

fsInfoChangeCounter (N, S)				
Content change counter $$fsInfoPathName				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access	

fsInfoAllChangeCounter (N, S)				
Total change counter $$fsInfoPathName				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Addressing				
Line index:			Max. line index:	
			$$fsInfoCount	
			Read access	

fsInfoCount (N, S)				
Number of info objects Number of file system info objects				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeFdpActCycle (N, S)				
Current value of F_DP communication cycle Current value of the F_DP communication cycle in seconds				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeFdpMaxCycle (N, S)				
Maximum value of F_DP communication cycle Maximum value of F_DP communication cycle in s				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeNumActiveFsdp (N, S)				
Number F_SENDDP Number of active F_SENDDP connections				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	16
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeFsdpFDataNck (N, S)				
F_SENDDP user data The F user data sent from F_SENDDP to F_RECVDP				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xFFFF
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFsdpActComTime (N, S)				
Current F_SENDDP communication time Current F_SENDDP communication time in seconds The communication time is the time from F_SENDDP sending the message telegram until the arrival of the correct acknowledge telegram from F_RECVDP.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFsdpMaxComTime (N, S)				
Maximum value of the F_SENDDP communication time Maximum value of the F_SENDDP communication time in s After a communication error, the maximum value is reset to 0 by the user acknowledgment.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFsdpDriverStateNck (N, S)				
Current state of F_SENDDP driver Current state of the F_SENDDP driver				
Return value	0 = Not parameterized 1 = Initialization 2 = F_SENDDP ready: waiting for F_RECVDP 3 = F_RECVDP ready: waiting for sequence no. = 1 4 = F_SENDDP and F_RECVDP ready: waiting for user acknowledgment from F_RECVDP 5 = Normal operation			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	5
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

safeFsdpErrReacNck (N, S)		$A_FSDP_ERR_REAC[n]		
F_SENDDP error reaction NCK The user can define the error reaction irrespective of the machining situation or the coupling of the communication partners.				
Return value	0 = Alarm 27350 + stop D/E 1 = Alarm 27350 2 = Alarm 27351 (display only, self-clearing) 3 = No reaction			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFsdpErrorNck (N, S)		$A_FSDP_ERROR[n]		
NCK communication error A communication error has been detected. The cause is stated in the diagnostics data				
Return value	0 = No communication error 1 = Communication error detected			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFsdpSubsOnNck (N, S)		$A_FSDP_SUBS_ON[n]		
F_SENDDP output of substitute values on NCK The communication relationship is not in normal operation. If F_RECVDP is active, it outputs substitute values. The signal is set during the start-up of the F communication and in the event of a communication error. ERROR = 0 AND SUBS_ON = 1 => Power up ERROR = 1 AND SUBS_ON = 1 => Communication error				
Return value	0 = Process values are output by F_RECVDP 1 = Substitute values are output by F_RECVDP			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	

safeFsdpSubsOnNck (N, S)	$A_FSDP_SUBS_ON[n]	
	16	
	Read access	

safeFsdpStatusSubsNck (N, S)				
F_RECVDP outputs substitute values Status signal in the acknowledgment telegram from F_RECVDP to F_SENDDP. F_RECVDP informs F_SENDDP with this signal that there is a communication error, and it is currently outputting substitute values. F_RECVDP resets the signal when it receives a user acknowledgment				
Return value	0 = F_RECVDP outputs process values 1 = F_RECVDP outputs substitute values			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFsdpDiagNck (N, S)	$A_FSDP_DIAG[n]			
F_SENDDP communication/system error Diagnostics data for F_SENDDP communication/system error				
Return value	10H = Timeout detected (TO) 20H = Sequence number error detected (SN) 40H = CRC error detected (CRC) 2000H = Deviations in the F telegram data detected (TD) 4000H = Sign-of-life monitoring error detected (LS) 8000H = Asynchronous error state detected (SF)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xFFFFFFFF
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeNumActiveFrdp (N, S)				
Number F_RECVDP Number of active F_RECVDP connections				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	16

safeNumActiveFrdp (N, S)				
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeFrdpFDataNck (N, S)				
F_RECVDP user data F user data received				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xFFFF
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpActComTime (N, S)				
Current F_RECVDP communication time Current F_RECVDP communication time in seconds				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpMaxComTime (N, S)				
Maximum value of F_RECVDP communication time Maximum value of F_RECVDP communication time in s				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpDriverStateNck (N, S)				
Current state of F_RECVDP driver Current state of the F_RECVDP driver				
Return value	0 = Not parameterized 1 = Initialization 2 = F_RECVDP ready: waiting for F_SENDDP 3 = F_SENDDP ready: waiting for sequence number = 1 4 = F_SENDDP and F_RECVDP ready: waiting for user acknowledgment after error 5 = Normal operation			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	5
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpSubsNck (N, S)		$A_FRDP_SUBS[n]		
NCK substitute values The user can define substitute values. These are then output to the application during power up, and communication errors are output instead of the process values.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xFFFF
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpErrReacNck (N, S)		$A_FRDP_ERR_REAC[n]		
F_RECVDP error reaction NCK The user can define the error reaction irrespective of the machining situation or the coupling of the communication partners.				
Return value	0 = Alarm 27350 + stop D/E 1 = Alarm 27350 2 = Alarm 27351 (display only, self-clearing) 3 = No reaction			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpErrorNck (N, S)		$A_FRDP_ERROR[n]		
F_RECVDP communication error NCK An F_RECVDP communication error has been detected. The cause is stated in the diagnostics data.				
Return value	0 = No communication error 1 = Communication error detected			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpSubsOnNck (N, S)		$A_FRDP_SUBS_ON[n]		
R_RECVDP output of substitute value NCK During power up and in the event of a communication error, substitute values are output. ERROR = 0 AND SUBS_ON = 1 => Power up ERROR = 1 AND SUBS_ON = 1 => Communication error				
Return value	0 = Process values are output 1 = Substitute values are output			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpAckReqNck (N, S)		$A_FRDP_ACK_REQ[n]		
Requirement for user acknowledgment NCK After a communication error, the F_DP communication is back in cyclic operation. A user acknowledgment is required to release normal operation with output of the process values.				
Return value	0 = No user acknowledgment required 1 = User acknowledgment required			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpDiagNck (N, S)		$A_FRDP_DIAG[n]		
F_RECVDP communication/system error Diagnostics data for F_RECVDP communication/system error				
Return value	0010H = Timeout detected (TO) 0020H = Sequence number error detected (SN) 0040H = CRC error detected (CRC) 2000H = Deviations in the F telegram data detected (TD) 4000H = Sign-of-life monitoring error detected (LS) 8000H = Asynchronous error state detected (SF)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xFFFFFFFF
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpSendModeNck (N, S)		$A_FRDP_SENDMODE[n]		
Safety mode communication partner inactive Current operating mode of the F-CPU of the F_SENDDP communication partner				
Return value	0: FALSE: The F-CPU is in safety mode 1: TRUE: The F-CPU is in deactivated safety mode			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFsdpErrReacPlc (N, S)				
F_SENDDP error reaction PLC The user can define the error reaction irrespective of the machining situation or the coupling of the communication partners.				
Return value	0 = Alarm 27350 + stop D/E 1 = Alarm 27350 2 = Alarm 27351 (display only, self-clearing) 3 = No reaction			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	3
Addressing				
Line index:			Max. line index:	

safeFsdpErrReacPlc (N, S)				
			16	
			Read access	

safeFrdpSubsPlc (N, S)				
PLC substitute values The user can define substitute values. These are then output to the application during power up, and communication errors are output instead of the process values.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xFFFF
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

safeFrdpErrReacPlc (N, S)				
F_RECVDP error reaction PLC The user can define the error reaction irrespective of the machining situation or the coupling of the communication partners.				
Return value	0 = Alarm 27350 + stop D/E 1 = Alarm 27350 2 = Alarm 27351 (display only, self-clearing) 3 = No reaction			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			16	
			Read access	

anPrepActLoad (N, S)		$AN_PREP_ACT_LOAD		
Current preprocessing run time Current preprocessing run time throughout all channels				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anPrepMaxLoad (N, S)		$AN_PREP_MAX_LOAD		
Longest preprocessing run time Longest preprocessing run time throughout all channels				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anPrepMinLoad (N, S)		$AN_PREP_MIN_LOAD		
Shortest preprocessing run time Shortest preprocessing run time throughout all channels				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anPrepActLoadGross (N, S)		$AN_PREP_ACT_LOAD_GROSS		
Current gross preprocessing run time Current gross preprocessing run time throughout all channels				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anPrepMaxLoadGross (N, S)		$AN_PREP_MAX_LOAD_GROSS		
Longest gross preprocessing run time Longest gross preprocessing run time throughout all channels				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anPrepMinLoadGross (N, S)		$AN_PREP_MIN_LOAD_GROSS		
Shortest gross preprocessing run time Shortest gross preprocessing run time throughout all channels				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anAuxfuListGroupindex (N, S)		$AN_AUXFU_LIST_GROUPINDEX[n]		
Global list of auxiliary functions - group index Group index of the auxiliary function collected in the channel. The variable is only valid in combination with block search type 5 (SERUPRO).				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	MD_MAXNUM_AUX-FU_GROUPS - 1
Addressing				
Line index:			Max. line index:	
			1280	
			Read access and write access	

anAuxfuListChanno (N, S)		$AN_AUXFU_LIST_CHANNO[n]		
Global list of auxiliary functions - channel no. Channel number of the auxiliary function collected in the channel. The variable is only valid in combination with block search type 5 (SERUPRO).				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	MD_MAXNUM_AUX-FU_CHANNELS
Addressing				
Line index:			Max. line index:	
			1280	
			Read access and write access	

anAuxfuListEndindex (N, S)		$AN_AUXFU_LIST_ENDINDEX		
Last valid index of the global Auxfu list The variable determines the last valid index for the global auxiliary function list.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-1	MD_MAXNUM_AUX-FU_LIST_INDEX

anAuxfuListEndindex (N, S)	$AN_AUXFU_LIST_ENDINDEX	
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

anAxctSwE (N, S)	$AN_AXCTSWE			
Axis container rotation slot released				
Has a release for rotation been granted for a slot on an axis container?				
Bit mask, each bit corresponds to a slot, e.g. 0x5 corresponds to slots 1 and 3.				
Bit == 1: The slot of an axis container has been released for rotation.				
Bit == 0: The slot of an axis container has not been released for rotation.				
Example: Axis container with 4 slots: 'Hfff5' slot 1 and 3 have been released for rotation.				
As soon as a slot has been released for axis container rotation, bit == 1 is also output for unused slots. See example 'Hfff0'.				
If the slots of an axis container are distributed across several NCUs, the current state of the slots on other NCUs is only displayed				
if all of the slots on the other NCU have been released for axis container rotation.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xffffffff
Addressing				
Line index:			Max. line index:	
			$$numContainer	
			Read access	

anLaiAxIsAxctax (N, S)	$AN_LAI_AX_IS_AXCTAX			
Bit mask: LAI axis is in the axis container				
Bit mask that displays whether an axis in the logical NCK machine axis image				
(machine data 10002 $MN_AXCONF_LOGIC_MACHAX_TAB) is an axis in an axis container				
(machine data 1270x/1271x $MN_AXCT_AXCONF_ASSIGN_TABi).				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xffffffff
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anLaiAxIsLinkax (N, S)	$AN_LAI_AX_IS_LINKAX	
Bit mask: is the LAI axis a link axis?		
Bit mask that displays whether an axis in the logical NCK machine axis image		
(machine data 10002 $MN_AXCONF_LOGIC_MACHAX_TAB) is a link axis		
(axis physically connected to another NCU).		

anLaiAxIsLinkax (N, S)		$AN_LAI_AX_IS_LINKAX		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xffffffff
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anLaiAxIsLeadLinkax (N, S)		$AN_LAI_AX_IS_LEADLINKAX		
Bit mask: is the LAI axis a lead link axis? Bit mask that displays whether an axis in the logical NCK machine axis image (machine data 10002 $MN_AXCONF_LOGIC_MACHAX_TAB) is a lead link axis, i.e. the same machine axis is referred to on several NCUs via MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB and the axial MD30554 $MA_AXCONF_ASSIGN_MASTER_NCU specifies which NCU is the master NCU, which generates the setpoint value for the position controller following power-up.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xffffffff
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anLaiAxToMachax (N, S)		$AN_LAI_AX_TO_MACHAX		
Assignment of physical axis to LAI axis. The NCU and machine axis are output for an LAI axis, representing the physical image of the axis. Here the NCU ID is output as from position 10000, e.g. 20005: NCU 2 axis 5. Without an NCU link, i.e. if there is only one NCU, only the number of the machine axis will be output. In this case, the NCU ID is equal to zero. If the LAI axis is not used, 0 is returned.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			>$$maxnumGlobMachAxes	
			Read access	

anLaiAxToIpoNcChanax (N, S)		$AN_LAI_AX_TO_IPO_NC_CHANAX		
Interpolator (NCU or channel, channel axis) If the LAI axis is currently interpolated to this NCU, the channel and channel axis number which define the interpolator of the axis are output. If the LAI axis is currently interpolated to a different NCU, the NCU identifier of the interpolated NCU and the global axis number of the machine axis are output. This global axis number can then be used to transfer the interpolated channel and the channel axis number to the other NCU, with NCU ID 2, with $AN_IPO_CHANAX[103]. If no LAI axis is used, 0 is returned.				
Return value	The channel is output as from position 100, and the channel axis number as from position 1, e.g. 1005 - channel 10 channel axis 5. These values are always lower than 10000. Here the NCU is output as from position 10000, e.g. 20103: NCU 2 and the global axis number is 103.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			>$$maxnumGlobMachAxes	
			Read access	

anIpoChanax (N, S)		$AN_IPO_CHANAX		
Interpolator (channel, channel axis) of glob. axis For a global axis number returned by valpoNcChanax, the channel and channel axis number that define the writing interpolator of the axis are output. Here the channel is output as from position 100, and the channel axis number as from position 1, e.g. 1005 - channel 10 channel axis 5. If the axis with the specified global axis number is not used on this NCU, 0 will be returned.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			170	
			Read access	

anSLTrace (N, S)		$AN_SLTRACE		
Trigger variable for the logging function This variable is reserved for the application SinUTrace and Operate-Trace. It serves as a trigger variable for the logging function. The following coding is recommended: 0: Inactive 1: Start logging requested 2: Stop logging requested The value is generally set by the part program, the reset by the application via OPI.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

safePsNumActiveDrivers (N, S)				
Number of active PROFIsafe drivers				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$safePsMaxnumDrivers
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safePsNumDisabledDrivers (N, S)				
Number of inactive PROFIsafe drivers				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$safePsMaxnumDrivers
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safePsActCycle (N, S)				
Current value of the PROFIsafe communication cycle Current value of the PROFIsafe communication cycle in s				
Units and value ranges				
Physical unit:	Data type:	Initial value		

safePsActCycle (N, S)				
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safePsMaxCycle (N, S)				
Maximum value of the PROFIsafe communication cycle Maximum value of the PROFIsafe communication cycle in s				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safePsAddress (N, S)				
PROFIsafe address				
Return value	0 = Not parameterized >0 = PROFIsafe address			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsHostAddress (N, S)				
PROFIsafe host address PROFIsafe host address F module				
Return value	0 = Not parameterized >0 = PROFIsafe host address			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsSlaveAddress (N, S)				
PROFIBUS slave address PROFIBUS slave address F module				
Return value	>0 = Not parameterized >0 = PROFIBUS slave address			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsModuleSlotNo (N, S)				
Slot number F module				
Return value	0 = Not parameterized >0 = Slot number			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsDriverMode (N, S)				
PROFIsafe link mode				
Return value	0 = Not parameterized 1 = Inactive 2 = Active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsDriverVersion (N, S)				
PROFIsafe version PROFIsafe version F driver				
Return value	0 = Not parameterized 1 = PROFIsafe V1 2 = PROFIsafe V2			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsModuleType (N, S)				
F module type				
Return value	0 = Not parameterized 1 = F input module 2 = F output module 3 = F input/output module			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsNumSubSlotsIn (N, S)				
Number of F user data IN subslots Number of subslots F user data telegram incoming direction				
Return value	0 = Not parameterized >0 = Number of subslots			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$safePsMaxnumSub-Slots
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsNumSubSlotsOut (N, S)				
Number of F user data OUT subslots Number of subslots F user data telegram outgoing direction				
Return value	0 = Not parameterized >0 = Number of subslots			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$safePsMaxnumSub-Slots
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsFDataIn (N, S)				
User data from F module F user data received from the PROFIsafe driver				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xffffffff
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers * $$safePsMaxnum-SubSlots	
			Read access	

safePsFDataOut (N, S)				
User data to F module F user data sent by the PROFIsafe driver				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0xffffffff
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers * $$safePsMaxnum-SubSlots	
			Read access	

safePsDriverState (N, S)				
Current state of PROFIsafe driver				
Return value	0 = Not parameterized 1 = Communication setup 2 = Communication setup: waiting for error-free telegrams 3 = Communication: waiting for error-free telegrams with the expected consecutive number 4 = Communication: normal operation 5 = Communication: waiting for acknowledgment after error			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	5
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsDriverError (N, S)				
Communication error F driver A communication error has been detected. The cause is stated in the diagnostics data.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_BOOLEAN	0		
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsDiagHost (N, S)				
PROFIsafe-host F communication error Diagnostic data PROFIsafe-host communication and system error				
Return value	0x0004 = Error checksum (CRC) 0x0008 = Error timeout (TO) 0x0010 = Activate substitute values in slave 0x0100 = Host initialization active 0x0200 = Error consecutive number (CN) 0x0400 = Error host state NCK/PLC (SF) 0x0800 = Empty telegram detected (EA) 0x1000 = Timeout host internal (TF)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0xffff
Addressing				
Line index:			Max. line index:	

safePsDiagHost (N, S)			
		$$safePsMaxnumDrivers	
		Read access	

safePsDiagSlave (N, S)				
Status data PROFIsafe slave Status data from PROFIsafe slave				
Return value	0x0002 = Error in slave application 0x0004 = Error checksum (CRC) 0x0008 = Error timeout (TO) 0x0010 = Substitute values activated			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0xffff
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsParamMaxComTime (N, S)				
Maximum configured communication time F module Maximum configured communication time in s. The communication time is the time from the sending of a telegram by the PROFIsafe host until the arrival of the correct response telegram from the PROFIsafe slave/device				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsActComTime (N, S)				
Actual communication time Current communication time in s. The communication time is the time from the sending of a telegram by the PROFIsafe host until the arrival of the correct response telegram from the PROFIsafe slave/device				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsMaxComTime (N, S)				
Maximum communication time Maximum value of the communication time in s. The maximum value is reset to 0 by a Reset after a communication error				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
s, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$safePsMaxnumDrivers	
			Read access	

safePsMaxnumDrivers (N, S)				
Maximum number of PROFIsafe drivers				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safePsMaxnumSubSlots (N, S)				
Maximum number of subslots F user data				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeMaxNumMarker (N, S)				
Max. number MARKERSI for the SPL Maximum number of markers for the SPL				
Return value	192 = SPL with a max. of 192 MARKERSI			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeMaxNumExtInput (N, S)				
Max. number INSE for SPL Maximum number of external inputs of the Safe Programmable Logic				
Return value	192 = SPL with a max. of 192 INSE			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeMaxNumExtOutput (N, S)				
Max. number OUTSE for SPL Maximum number of external outputs of the SPL				
Return value	192 = SPL with a max. of 192 OUTSE			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeMaxNumIntInput (N, S)				
Max. number INSI for SPL Maximum number of internal inputs of the SPL				
Return value	192 = SPL with a max. of 192 INSI			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeMaxNumIntOutput (N, S)				
Max. number OUTSI for the SPL Maximum number of internal outputs of the SPL				
Return value	192 = SPL with a max. of 192 OUTSI			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	

safeMaxNumIntOutput (N, S)				
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeMaxNumPlcInOut (N, S)				
Max. number PLCSIIN/OUT Maximum number of Safety signals from the PLC to the NCK and vice versa				
Return value	96 = SPL with a max. of 96 PLCSIIN and 96 PLCSIOUT			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeMode (N, S)				
Safety mode Configured safety mode				
Return value	0 = Not updated/invalid 1 = SINUMERIK Safety Integrated (Drive Based) 2 = Reserved) 3 = SINUMERIK Safety Integrated plus (F-PLC) 3 = SINUMERIK Safety Integrated (F-PLC) 4 = SINUMERIK Safety Integrated (SPL) 5 = No safety			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

handWheelTestIpoDiffPulses (N, S)				
Differential handwheel pulses per IPO cycle Define differential handwheel pulses per IPO cycle for handwheel simulation via OPI				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	

handWheelTestIpoDiffPulses (N, S)			
		$$numHandWheels	
		Read access and write access	

handWheelTestAbsPulses (N, S)				
Absolute handwheel pulses Query handwheel pulses via OPI				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$$numHandWheels	
			Read access	

handWheelTestMode (N, S)				
Test mode of the handwheel simulation Define test mode of handwheel simulation via OPI				
Return value	0 = TEST_MODE_CLEAR Resetting of all variables of the handwheel simulation 1 = TEST_MODE_STOP Stopping the output of handwheel pulses to the IPO cycle 10 = TEST_MODE_SINGLE Fully output pulses in $$handWheelTestDiffPulses in the next IPO cycle (default) 11 = TEST_MODE_CONTINUE Out put pulses in $$handWheelTestIpoDiffPulses continuously in every IPO cycle 12 = TEST_MODE_PERIODE Output pulses in $$handWheelTestDiffPulses continuously in pulse packages that are defined via $$handWheelTestIpoDiffPulses				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numHandWheels	
			Read access and write access	

anActivateCollCheck (N, S)		$AN_ACTIVATE_COLL_CHECK		
ActivateCollCheck on the interface PLC->NCK Status of the field ActivateCollcheck on the interface PLC->NCK (DB10.DBX234.0 - DB10.DBX241.7). Data are made available in groups of 4 bytes, i.e. with index 1 you receive the first 4 bytes (DB10.DBX234.0 - DB10.DBX237.7), with index 2 the second 4 bytes (DB10.DBX238.0 - DB10.DBX241.7)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

anCollCheckOff (N, S)		$AN_COLL_CHECK_OFF		
Byte DeactivateCollCheckGroups PLC->NCK Status of the byte DeactivateCollCheckGroups on the interface PLC->NCK (DB10.DBB58) for the operating-mode-dependant suppression of the collision avoidance for groups of protection areas.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anFacetsAvailable (N, S)		$AN_FACETS_AVAILABLE		
Available number of collision facets Machine parts can be modeled from 3D facets for the collision avoidance function. The maximum number of 3D facets is limited by the machine data 18895 $MN_MM_MAXNUM_3D_FACETS. The anFacetsAvailable variable indicates how many facets are still available.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anFacetsAct (N, S)	$AN_FACETS_ACT	
Currently used number of collision facets Machine parts can be modeled from 3D facets for the collision avoidance function. The maximum number of 3D facets is limited by the machine data 18895 $MN_MM_MAXNUM_3D_FACETS. The anFacetsAct variable indicates how many facets are currently being used.		

anFacetsAct (N, S)		$AN_FACETS_ACT		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anFacetsMin (N, S)		$AN_FACETS_MIN		
Minimum number of collision facets used Machine parts can be modeled from 3D facets for the collision avoidance function. The maximum number of 3D facets is limited by the machine data 18895 $MN_MM_MAXNUM_3D_FACETS. The anFacetsMin variable indicates the minimum number of facets used so far.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

anFacetsMax (N, S)		$AN_FACETS_MAX		
Maximum number of collision facets Machine parts can be modeled from 3D facets for the collision avoidance function. The maximum number of 3D facets is limited by the machine data 18895 $MN_MM_MAXNUM_3D_FACETS. The anFacetsMax variable indicates the maximum number of facets used so far.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

anFacetsInternAvailable (N, S)		$AN_FACETS_INTERN_AVAILABLE		
Available number of internal collision facets Variable machine parts such as tools are automatically modeled from 3D facets by means of the collision avoidance function. The maximum number of 3D facets is limited by the machine data 18894 $MN_MM_MAXNUM_3D_FACETS_INTERN. The anFacetsInternAvailable variable indicates how many facets are still available.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	

anFacetsInternAvailable (N, S)	$AN_FACETS_INTERN_AVAILABLE	
	1	
	Read access	

anFacetsInternAct (N, S)		$AN_FACETS_INTERN_ACT		
Current number of internal collision facets Variable machine parts such as tools are automatically modeled from 3D facets by means of the collision avoidance function. The maximum number of 3D facets is limited by the machine data 18894 $MN_MM_MAXNUM_3D_FACETS_INTERN. The anFacetsInternAct variable indicates how many facets are currently being used.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anFacetsInternMin (N, S)		$AN_FACETS_INTER_MIN		
Minimum number of internal collision facets Variable machine parts such as tools are automatically modeled from 3D facets by means of the collision avoidance function. The maximum number of 3D facets is limited by the machine data 18894 $MN_MM_MAXNUM_3D_FACETS_INTERN. The anFacetsInternMin variable indicates the minimum number of facets used so far.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

anFacetsInternMax (N, S)		$AN_FACETS_INTERN_MAX		
Maximum number of internal collision facets Variable machine parts such as tools are automatically modeled from triangle areas by means of the Collision prevention function. The maximum number of triangles is limited by the machine data 18894 $MN_MM_MAXNUM_3D_FACETS_INTERN. The variable anFacetsInternMax indicates how many have been used so far.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

anProtAreasAct (N, S)		$AN_PROT_AREAS_ACT		
Number of protection areas used The collision avoidance function can monitor a maximum number of protection areas. This number is determined by machine data 18890 $MN_MM_MAXNUM_3D_PROT_AREAS. The system variable $AN_PROT_AREAS_ACT states how many of them are currently being used.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anProtAreaElemAct (N, S)		$AN_PROT_AREA_ELEM_ACT		
Number of protection area elements used The collision avoidance function can monitor a maximum number of protection area elements. This number is determined by machine data 18892 $MN_MM_MAXNUM_3D_PROT_AREA_ELEM. The system variable $AN_PROT_AREA_ELEM_ACT states how many of them are currently being used.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anCollPairsAct (N, S)		$AN_COLL_PAIRS_ACT		
Number of collision pairs used The collision avoidance function can monitor a maximum number of protection area pairs. This number is determined by machine data 18898 $MN_MM_MAXNUM_3D_COLL_PAIRS. The system variable $AN_COLL_PAIRS_ACT states how many of them are currently being used.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anTProtElemAct (N, S)	$AN_T_PROT_ELEM_ACT	
Number of tool protection area elements used The collision avoidance function can monitor a maximum number of tool protection area elements. This number is determined by machine data 18893 $MN_MM_MAXNUM_3D_T_PROT_ELEM. The system variable $AN_T_PROT_ELEM_ACT states how many of them are currently being used.		

anTProtElemAct (N, S)		$AN_T_PROT_ELEM_ACT		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anKinChainElemAct (N, S)		$AN_KIN_CHAIN_ELEM_ACT		
Number of kinematic elements used The kinematic chains can only use a maximum number of elements. This number is determined by machine data 18880 $MN_MM_MAXNUM_KIN_CHAIN_ELEM. The system variable $AN_KIN_CHAIN_ELEM_ACT states how many of them are currently being used.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

safeExtInputQuality (N, S)				
Property of external SPL input Property of an external NCK-SPL input signal				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumExtInput	
			Read access	

safeExtOutputQuality (N, S)				
Property of external SPL output Property of an external NCK-SPL output signal				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumExtOutput	
			Read access	

safeIntInputQuality (N, S)				
Property of internal SPL input Property of an internal NCK-SPL input signal				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumIntInput	
			Read access	

safeIntOutputQuality (N, S)				
Property of internal SPL output Property of an internal NCK-SPL output signal				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			$$safeMaxNumIntOutput	
			Read access	

anRobin (N, S)		$AN_ROBIN[index]		
NCK-PLC robotic status interface The system variable $AN_ROBIN[index] reads the relevant byte in the NCK-PLC interface robotic status.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

anRobout (N, S)		$AN_ROBOUT[index]		
NCK-PLC robotic control interface The system variable $AN_ROBOUT[index] reads the relevant byte in the NCK-PLC robotic control interface.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

pnRobin (N, S)		$PN_ROBIN[index]		
NCK-PLC robotic status interface (preprocessing) The system variable $PN_ROBIN[index] reads the relevant byte in the NCK-PLC interface robotic status.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

pnRobout (N, S)		$PN_ROBOUT[index]		
NCK-PLC robotic control interface (preprocessing) The system variable $PN_ROBOUT[index] reads the relevant byte in the NCK-PLC robotic control interface.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			8	
			Read access	

anPoweronState (N, S)	$AN_POWERON_STATE	
State of NCK power on The bit-coded variable indicates the state of the NCK power on. All bits = 0: NCK power on has not started. Bit0=1: The NCK power on has started, i.e. all NCK objects (channels etc.) have already been created and are being initialized. Bit1=1: The main run states can now be read. This means that all stations have been initialized, and that power on Reset has been executed together with the Reset INIT blocks. Bit2=1: User interventions (Reset, Stop etc.) are now possible and purposeful. This means that any configured Safety ProgEvent has been correctly completed or possibly could not be executed because of alarms. Any configured PowerOn ProgEvent is executed next provided that its execution is not prevented by alarms. Bit2=1: User interventions (Reset, Stop etc.) are now possible and purposeful. Any configured PowerOn ProgEvent will be executed provided that its execution is not prevented by alarms. Bit24=1: The NCK power on has finished together with all the ProgEvents that could be executed automatically (Safety ProgEvent, PowerOn ProgEvent). The bit does not indicate whether or not an error occurred during the power on (see Bit25). Bit24=1: The NCK power on has finished together with all the ProgEvents that could be executed automatically (Power-On-ProgEvent). The bit does not indicate whether or not an error occurred during power on (see Bit25). Bit25=1: The NCK power on finished with errors. This means, for example, that an error occurred while the stations were being initialized, during the Reset INIT blocks or the execution of the Safety ProgEvent. Other alarms indicate the exact problem, and the alarm responses indicate which actions can be executed. Bit25=1: The NCK power on finished with errors. This means, for example, that an error occurred while the stations were being initialized or with the reset init blocks. Other alarms indicate the exact problem, and the alarm responses indicate which actions can be executed.		

anPoweronState (N, S)		$AN_POWERON_STATE		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

measFctCmd (N, S)				
Start of motion measuring function				
Return value	0 = Function not active or abort 1 = Activate start of motion for all system axes			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

cncLockRequest (N, S)				
Order status for the CNC lock function				
Return value	0 = No task triggered by PI. 1 = The current task is being processed. 2 = The current task is faulty and has been rejected. 3 = The last task has been accepted. See $$cncLockState.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

cncLockState (N, S)				
Current status of the CNC lock function				
Return value	0 = Die CNC-Sperrfunktion ist nicht aktiv. 1 = The CNC lock function is active without errors. 2 = Errors have been detected in the active CNC lock function.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2

cncLockState (N, S)		
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

cncLockTimeBCD (N, S)				
Release date for the CNC lock Date on which the CNC lock function is activated (displayed in the PLC format): <Month>.<Day>.<Year> <Hours>:<Minutes>:<Seconds>.<Milliseconds> <Weekday> <Status> <Weekday> can assume the following values: "SUN", "MON", "TUE", "WED", "THU", "FRI", "SAT"				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DATETIME			
			Read access	

anCecInputNcu (N, S)	$AN_CEC_INPUT_NCU			
CEC table: Basic axis on NCU The variable indicates the number of the NCU, for which the basic axis is calculated. It returns the value 0 if no NCU has been programmed.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			62	
			Read access	

anCecInputAxis (N, S)	$AN_CEC_INPUT_AXIS			
CEC table: Basic axis The variable describes the number of the axis whose setpoint is used as entry for the compensation table. The value -1 indicates that no axis has been programmed.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	-1		
Addressing				
Line index:			Max. line index:	
			62	
			Read access	

anCecOutputNcu (N, S)		$AN_CEC_OUTPUT_NCU		
CEC table: Compensation axis on NCU The variable indicates the number of the NCU which the compensation axis is calculated to. It returns the value 0 if no NCU has been programmed.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			62	
			Read access	

anCecOutputAxis (N, S)		$AN_CEC_OUTPUT_AXIS		
CEC table: Compensation axis The variable describes the number of the axis onto which the output of the compensation table has an effect. The value -1 indicates that no axis has been programmed,				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	-1		
Addressing				
Line index:			Max. line index:	
			62	
			Read access	

anCecStep (N, S)		$AN_CEC_STEP		
CEC table: Distance between interpolation points The variable indicates the distance of the offset values.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			62	
			Read access	

anCecMin (N, S)		$AN_CEC_MIN		
CEC table: Starting position The variable indicates the start position of the compensation table.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0.0		

anCecMin (N, S)	$AN_CEC_MIN	
Addressing		
Line index:	Max. line index:	
	62	
	Read access	

anCecMax (N, S)		$AN_CEC_MAX		
CEC table: End position The variable indicates the end position of the compensation table.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			62	
			Read access	

anCecDirection (N, S)		$AN_CEC_DIRECTION		
CEC table: Direction-dependence The variable activates the direction-dependent activation of the compensation table: 0: Both traversing directions of the basic axis 1: Positive traversing direction of the basic axis -1: Negative traversing direction of the basic axis				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			62	
			Read access	

anCecMultByTable (N, S)		$AN_CEC_MULT_BY_TABLE		
CEC table: Multiplication The variable indicates the number of the table whose output value shall be multiplied with the output value of the compensation table.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			62	
			Read access	

anCecIsModulo (N, S)		$AN_CEC_IS_MODULO		
CEC table: Modulo functionality The variable indicates whether the values of the corresponding compensation table shall be repeated cyclically: TRUE: Cyclic repetition of the compensation table FALSE: No cyclic repetition of the compensation table				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_BOOLEAN	FALSE		
Addressing				
Line index:			Max. line index:	
			62	
			Read access	

anCecType (N, S)		$AN_CEC_TYPE		
CEC table: table type The variable specifies the compensation table type 0: no special table type 1: cylinder error compensation table type				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	FALSE		
Addressing				
Line index:			Max. line index:	
			62	
			Read access	

pModelScalingOverload (N, S)		$P_MODEL_SCALING_OVL		
Machine data description is that of the NCK model The variable indicates whether the original machine data description is overwritten during the NCK power on -1: There is no machine data for the programmed MMC number. 0: The original definition is available for the NCK model (e.g. 840D-732) 1,2,..: The original definition for the NCK model was overwritten (typical in NCK simulation products)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	FALSE		
Addressing				
Line index:			Max. line index:	
			MD_MAXNUM_MMCIDS	
			Read access	

pnChangeCntrNkData (N, S)		$PN_CHANGE_CNTR_NK_DATA		
Modification counter for kinematic chain data Modification counter for kinematic chain data (chain elements and NK_SWITCHes).				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	INT_MAX
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pnChangeCntrNkElem (N, S)		$PN_CHANGE_CNTR_NK_ELEM		
Modification counter for kinematic chain elements Modification counter for kinematic chain elements (without NK_SWITCH).				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	INT_MAX
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pnChangeCntrNkSwitch (N, S)		$PN_CHANGE_CNTR_NK_SWITCH		
Modification counter for $NK_SWITCHes Modification counter for $NK_SWITCHes.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	INT_MAX
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pnChangeCntrNpData (N, S)		$PN_CHANGE_CNTR_NP_DATA		
Modification counter for 3D protection area data Modification counter for 3D protection area data ($NP_xxx).				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	INT_MAX
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

pnChangeCntrNtData (N, S)		$PN_CHANGE_CNTR_NT_DATA		
Modification counter for transf. data ($NT_xxx) Modification counter for transformation data ($NT_xxx).				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	INT_MAX
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anCUTrace (N, S)		$AN_CUTRACE		
Trigger variable for the SINAMICS trace function Variable for triggering the trace function in SINAMICS. If telegram 390, 391 or 395 is used. writing the value 1 sets bit 4 in the drive parameter r898 "Control word drive object 1" in all Control Units activated via MD13120 $MN_CONTROL_UNIT_LOGIC_ADDRESS. Triggering via telegram must have been previously parameterized in the trace.				
Return value	Write: 0: No action 1: Trip trigger Read: Always 0, because the trigger cannot be read back			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_DWORD		0	1
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

3.2.6.14 SALA: Alarm list, oldest alarm

The NCK alarms are sorted in a list in the order they occurred, the oldest alarm appears at the top of the list. The alarm parameters are transferred as ASCII strings, the first character contains the type information for that parameter. The following types are used:

S: General string, e.g. part program name

A: Axis name / spindle name

K: Channel name

N: Block number

Y: System error

D: Drive number

If a parameters is not assigned, an "S" is transferred.

All variables in this module are privileged variables! This means that cyclic acknowledgments are sent for these variables even if the cyclic services are no longer serviced by the NCK because of block cycle problems.

Attention: Privileged variables lose this characteristic if they are combined with non-privileged variables in a request. -> Do not combine alarm variables with other variables in a cluster!

In addition it is presumed that the cyclic services are set "on change" for the alarm variables and are not combined with other variables (not even with privileged variables) in the same request.

The module SALA only contains the alarms that are generated in the NCK. It contains neither PLC nor HMI alarms. In order to read all alarms, the OEM-HMI user should use the alarm server functions and not read the SALA module directly.

alarmNo (N, SALA)				DA
Ordinal number of an alarm Ordinal number of an alarm (how many alarms since control ON)				
Return value	0 = unknown alarm			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

clearInfo (N, SALA)				DA
Acknowledgment criterion for an alarm				
Return value	1 = Power On 2 = Reset 3 = Cancel 4 = Alarm is cancelled by NCK-software 5 = Alarm is cancelled by starting a program 6 = Alarm is cancelled by RESET in all channels of the bags (from SW 4.1) 7 = Alarm is cancelled by RESET in all channels of the NC (from SW 4.1)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

fillText1 (N, SALA)				DA
Parameter 1 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

fillText2 (N, SALA)				DA
Parameter 2 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

fillText3 (N, SALA)				DA
Parameter 3 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

fillText3 (N, SALA)				DA
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

fillText4 (N, SALA)				DA
Parameter 4 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

textIndex (N, SALA)				
Alarm number Alarm number of the actual alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

timeBCD (N, SALA)				
Time stamp of the alarm Time stamp of the alarm. The time stamp is displayed in the corresponding PLC format.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DATETIME			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

3.2.6.15 SALAC: Alarm actions, oldest alarm

In a given alarm, all values in the SALAC module are identical to the corresponding variables in the SALA, SALAP and SALAL modules with the exception of actionType and actionCount.

The same alarm can be found in the various modules by comparing the values of alarmNo.

A client is registered with an alarm server when cyclic reading of the SALAC module has been set.

If the operator panel sets cyclic reading when a data in the module changes and column index 0 has been specified, then the variable server sends the entire data block to the operator panel if the alarm server receives a new alarm action.

Another alarm server client is registered at each cyclic reading of the SALAC module which has been set.

This mechanism therefore functions with more than one user interface connected. Registration is withdrawn when the corresponding cyclic reading is terminated. As cyclic reading only applies to registered clients,

each regular and noncyclic reading returns the default value for the requested variable.

actionType (N, SALAC)				
Type of alarm action Specifies whether the alarm is deleted or activated.				
Return value	0: No alarm action pending 1: Alarm set 2: Alarm deleted			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

actionCount (N, SALAC)				
Number of alarm actions A unique number assigned to Alarm Action. On power ON it is reset to zero by the NCK. For each new alarm action it will be incremented by one.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

alarmNo (N, SALAC)				
Alarm number A unique number assigned to the alarm. It will be incremented by one for each reported alarm.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

textIndex (N, SALAC)				
Alarm number (actual alarm) Alarm number of the actual alarm				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

timeBCD (N, SALAC)				
Time stamp of the alarm Time stamp of the alarm. The time stamp is displayed in the corresponding PLC format.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DATETIME			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

clearInfo (N, SALAC)				
Alarm deletion information Describes the acknowledgment criterion for the alarm.				
Return value	1 = Power ON 2 = Reset 3 = Delete 4 = Alarm is deleted by NCK software 5 = Alarm is deleted by program call 6 = Alarm is deleted by RESET in all channels of the mode groups (from SW 4.1) 7 = Alarm is deleted by RESET in all channels of the NC (from SW 4.1)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	1	1	7
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

fillText1 (N, SALAC)				
Alarm filler text 1 Parameter 1, ASCII string which is inserted in the standard alarm text to supplement the alarm description.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

fillText2 (N, SALAC)				
Alarm filler text 2 Parameter 2, ASCII string which is inserted in the standard alarm text to supplement the alarm description.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

fillText3 (N, SALAC)				
Alarm filler text 3 Parameter 3, ASCII string which is inserted in the standard alarm text to supplement the alarm description.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

fillText4 (N, SALAC)				
Alarm filler text 4 Parameter 4, ASCII string which is inserted in the standard alarm text to supplement the alarm description.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

3.2.6.16 SALAL: Alarm list, most recent alarm

The NCK alarms are sorted in a list in the order they occurred, the most recent alarm appears at the bottom of the list. The alarm parameters are transferred as ASCII strings, the first character contains the type information for that parameter. The following types are used:

S: General string, e.g. part program name

A: Axis name / spindle name

K: Channel name

N: Block number

Y: System error

D: Drive number

If a parameters is not assigned, an "S" is transferred.

All variables in this module are privileged variables! This means that cyclic acknowledgements are sent for these variables even if the cyclic services are no longer serviced by the NCK because of block cycle problems.

Attention: Privileged variables lose this characteristic if they are combined with non-privileged variables in a request. -> Do not combine alarm variables with other variables in a cluster!

In addition it is presumed that the cyclic services are set "on change" for the alarm variables and are not combined with other variables (not even with privileged variables) in the same request.

The module SALA only contains the alarms that are generated in the NCK. It contains neither PLC nor HMI alarms. In order to read all alarms, the OEM-HMI user should use the alarm server functions and not read the SALA module directly.

alarmNo (N, SALAL)				DA
Ordinal number of an alarm Ordinal number of an alarm (how many alarms since control ON)				
Return value	0 = unknown alarm			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

clearInfo (N, SALAL)				DA
Acknowledgment criterion for an alarm				
Return value	1 = Power On 2 = Reset 3 = Cancel 4 = Alarm is cancelled by NCK-software 5 = Alarm is cancelled by starting a program 6 = Alarm is cancelled by RESET in all channels of the bags (from SW 4.1) 7 = Alarm is cancelled by RESET in all channels of the NC (from SW 4.1)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Validity:	From NCK version 3.6		Read access	

fillText1 (N, SALAL)				DA
Parameter 1 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

fillText2 (N, SALAL)				DA
Parameter 2 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

fillText3 (N, SALAL)				DA
Parameter 3 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

fillText4 (N, SALAL)				DA
Parameter 4 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

textIndex (N, SALAL)				
Alarm number Alarm number of the actual alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

timeBCD (N, SALAL)				
Time stamp of the alarm Time stamp of the alarm. The time stamp is displayed in the corresponding PLC format.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DATETIME			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

3.2.6.17 SALAP: Alarm list, highest priority alarm

The alarms of the NCK are arranged in a list in the order of their priorities. The highest-priority alarm is set to position 1. The alarm list is sorted according to the following criteria:

1. Sorting criterion: deletion criterion (highest priority at 1st position)
   - NC power OFF/ON
   - Press reset key
   - Press "Delete alarm"
   - Press "NC start"
   - Press "Recall" key
2. Sorting criterion: alarm occurrence time

The alarm parameters are transferred as ASCII strings, the first character contains the type information for the parameter. The following types are used:

S: General string, e.g. part program name

A: Axis name / spindle name

K: Channel name

N: Block name

Y: System error

D: Drive number

If a parameter is not assigned, an "S" is transferred.

All variables in this module are privileged variables! This means that cyclic acknowledgments are sent for these variables even if the cyclic services are no longer serviced by the NCK because of block cycle problems.

Notice: Privileged variables lose this characteristic if they are combined with non-privileged variables in a request. -> Do not combine alarm variables with other variables in a cluster!

In addition it is presumed that the cyclic services are set "on change" for the alarm variables and are not combined with other variables (not even with privileged variables) in the same request.

The module SALAP only contains the alarms that are generated in the NCK. It contains neither PLC nor HMI alarms. In order to read all alarms, the OEM-HMI user should use the alarm server functions and not read the SALAP module directly.

alarmNo (N, SALAP)				**DA**
Ordinal number of an alarm Ordinal number of an alarm (how many alarms since control ON)				
Return value	0 = unknown alarm			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			

alarmNo (N, SALAP)				DA
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

clearInfo (N, SALAP)				DA
Acknowledgment criterion for an alarm				
Return value	1 = Power On 2 = Reset 3 = Cancel 4 = Alarm is cancelled by NCK-software 5 = Alarm is cancelled by starting a program 6 = Alarm is cancelled by RESET in all channels of the bags (from SW 4.1) 7 = Alarm is cancelled by RESET in all channels of the NC (from SW 4.1)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Validity:	From NCK version 3.6		Read access	

fillText1 (N, SALAP)				DA
Parameter 1 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

fillText2 (N, SALAP)				DA
Parameter 2 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

fillText3 (N, SALAP)				DA
Parameter 3 of the alarm				
Units and value ranges				
Physical unit:	Data type:			

fillText3 (N, SALAP)				DA
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

fillText4 (N, SALAP)				DA
Parameter 4 of the alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

textIndex (N, SALAP)				
Alarm number Alarm number of the actual alarm				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

timeBCD (N, SALAP)				
Time stamp of the alarm Time stamp of the alarm. The time stamp is displayed in the corresponding PLC format.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DATETIME			
Addressing				
Line index:			Max. line index:	
			16	
Validity:	From NCK version 3.6		Read access	

3.2.6.18 SD: Servo data

The SD module makes servo data available.

These data can be accessed only via the cyc. variable service and the logging function (not individual variable service).

The row index is coded as follows:

The lower three places contain the NCK axis index

The fourth place contains the data format.

The column index is coded as follows:

The lower three places contain the signal ID

The fourth place contains the servo cycle

Data format coding:

0: 32 bit float

1: 64 bit float

Servo cycle coding:

0: The average value of all servo cycle values of one IPO cycle must be applied

61: The minimum value must be applied

62: The maximum value must be applied

n: The value of the nth servo cycle in the course of one IPO cycle must be applied

Maximum value of: IPO_SYSCLOCK_TIME_RATIO / POSCTRL_SYSCLOCK_TIME_RATIO

(1 <=n <=60)

Coding of signal ID:

1: Following error

2: Control deviation

3: Contour deviation

4: Actual position value, measuring system 1

5: Actual position value, measuring system 2

6: Position setpoint

7: Actual velocity value of active encoders (NCK)

8: Drive velocity setpoint (NCK)

9: Compensation value, measuring system 1

10: Compensation value, measuring system 2

11: Controller mode

12: Parameter set

13: Active measuring system

14: Position setpoint at controller input

15: Velocity setpoint at controller input

16: Acceleration setpoint at controller input
17: Velocity feedforward value (plus QEC)
18: Torque/force feedforward value
19: Torque/force limit value
20: Actual velocity, measuring system 1
21: Actual velocity, measuring system 2
22: Interpolation ended signal
23: Exact stop fine signal
24: Exact stop coarse signal
25: QEC learning criterion
26: QEC compensation value
27: Master-slave output torque compensatory controller
28: Master-slave tension torque
50: Utilization (m_soll/m_soll,grenz)
51: Active power
52: Torque/force setpoint (limited)
53: Torque generating actual current value i(q)
54: Actual speed/velocity motor
55: Valve lift setpoint
56: Actual valve lift
57: Actual pressure cylinder A end
58: Actual pressure cylinder B end
60: Safe actual position
61: Safe actual drive position
62: Safety-relevant input signal NCK
63: Safety-relevant output signal NCK
64: Safety-relevant input signal drive (from PLC)
65: Safety-relevant output signal drive (to PLC)
66: Reaction identifier for NCK
67: Reaction identifier for NCK/drive
68: Result list 1 NCK
69: Result list 1 drive
70: Result list 2 NCK
71: Result list 2 drive
72: Safety partial actual value
73: Actual velocity limit
74: Setpoint velocity limit
75: SI actual value difference
76: Current SI slip speed
77: Current SBR limit
78: External NCK-SPL inputs 1...32
79: External NCK-SPL inputs 32...64

80: External NCK-SPL outputs 1...32
81: External NCK-SPL outputs 33...64
82: Internal NCK-SPL inputs 1...32
83: Internal NCK-SPL inputs 33...64
84: Internal NCK-SPL outputs 1...32
85: Internal NCK-SPL outputs 33...64
86: NCK-SPL marker 1...32
87: NCK-SPL marker 33...64
88: NCK-SPL timer 1
89: NCK-SPL timer 2
90: NCK-SPL timer 3
91: NCK-SPL timer 4
92: NCK-SPL timer 5
93: NCK-SPL timer 6
94: NCK-SPL timer 7
95: NCK-SPL timer 8
96: External PLC-SPL inputs 1...32
97: External PLC-SPL inputs 33...64
98: External PLC-SPL outputs 1...32
99: External PLC-SPL outputs 33...64
101: Internal PLC-SPL inputs 1...32
102: Internal PLC-SPL inputs 33...64
103: Internal PLC-SPL outputs 1...32
107: Internal PLC-SPL outputs 33...64
108: PLC-SPL marker 1...32
109: PLC-SPL marker 33...64
110: NCK-SPL timer 9
111: NCK-SPL timer 10
112: NCK-SPL timer 11
113: NCK-SPL timer 12
114: NCK-SPL timer 13
115: NCK-SPL timer 14
116: NCK-SPL timer 15
117: NCK-SPL timer 16
119: Cam SGA NCK
120: SGA drive
121: Cam SGA drive
122: Actual value difference fine position - coarse position
123: Results list 3 NCK
124: Results list 3 drive
125: Results list 4 NCK
126: Results list 4 drive

127: Results list 5 NCK
128: Results list 5 drive
129: Results list 6 NCK
130: Results list 6 drive
131: Results list 7 NCK
132: Results list 7 drive
135: Smoothed actual velocity value NCK
200: Compile cycle signal 200
201: Compile cycle signal 201
202: Compile cycle signal 202
203: Compile cycle signal 203
204: Compile cycle signal 204
205: Compile cycle signal 205
206: Compile cycle signal 206
207: Compile cycle signal 207
208: Compile cycle signal 208
209: Compile cycle signal 209
210: Compile cycle signal 210
211: Compile cycle signal 211
212: Compile cycle signal 212
213: Compile cycle signal 213
214: Compile cycle signal 214
215: Compile cycle signal 215
216: Compile cycle signal 216
217: Compile cycle signal 217
218: Compile cycle signal 218
219: Compile cycle signal 219
220: Compile cycle signal 220
221: Compile cycle signal 221
222: Compile cycle signal 222
223: Compile cycle signal 223
224: Compile cycle signal 224
225: Compile cycle signal 225
226: Compile cycle signal 226
227: Compile cycle signal 227
228: Compile cycle signal 228
229: Compile cycle signal 229
230: Compile cycle signal 230
231: Compile cycle signal 231
232: Compile cycle signal 232
233: Compile cycle signal 233
234: Compile cycle signal 234

235: Compile cycle signal 235
236: Compile cycle signal 236
237: Compile cycle signal 237
238: Compile cycle signal 238
239: Compile cycle signal 239
240: Compile cycle signal 240
241: Compile cycle signal 241
242: Compile cycle signal 242
243: Compile cycle signal 243
244: Compile cycle signal 244
245: Compile cycle signal 245
246: Compile cycle signal 246
247: Compile cycle signal 247
248: Compile cycle signal 248
249: Compile cycle signal 249
250: Compile cycle signal 250
251: Compile cycle signal 251
252: Compile cycle signal 252
253: Compile cycle signal 253
254: Compile cycle signal 254
255: Compile cycle signal 255
300: External NCK-SPL inputs 65...96
301: External NCK-SPL inputs 97...128
302: External NCK-SPL inputs 129...160
303: External NCK-SPL inputs 161...192
304: External NCK-SPL outputs 65...96
305: External NCK-SPL outputs 97...128
306: External NCK-SPL outputs 129...160
307: External NCK-SPL outputs 161...192
308: Internal NCK-SPL inputs 65...96
309: Internal NCK-SPL inputs 97...128
310: Internal NCK-SPL inputs 129...160
311: Internal NCK-SPL inputs 161...192
312: Internal NCK-SPL outputs 65...96
313: Internal NCK-SPL outputs 97...128
314: Internal NCK-SPL outputs 129...160
315: Internal NCK-SPL outputs 161...192
316: NCK-SPL marker 65...96
317: NCK-SPL marker 97...128
318: NCK-SPL marker 129...160
319: NCK-SPL marker 161...192
320: External PLC-SPL inputs 65...96

321: External PLC-SPL inputs 97...128
322: External PLC-SPL inputs 129...160
323: External PLC-SPL inputs 161...192
324: External PLC-SPL outputs 65...96
325: External PLC-SPL outputs 97...128
326: External PLC-SPL outputs 129...160
327: External PLC-SPL outputs 161...192
328: Internal PLC-SPL inputs 65...96
329: Internal PLC-SPL inputs 97...128
330: Internal PLC-SPL inputs 129...160
331: Internal PLC-SPL inputs 161...192
332: Internal PLC-SPL outputs 65...96
333: Internal PLC-SPL outputs 97...128
334: Internal PLC-SPL outputs 129...160
335: Internal PLC-SPL outputs 161...192
336: PLC-SPL marker 65...96
337: PLC-SPL marker 97...128
338: PLC-SPL marker 129...160
339: PLC-SPL marker 161...192
340: Single-channel SI signals PLC to NCK 1...32
341: Single-channel SI signals PLC to NCK 33...64
342: Single-channel SI signals PLC to NCK 65...96
343: Single-channel SI signals NCK to PLC 1...32
344: Single-channel SI signals NCK to PLC 33...64
345: Single-channel SI signals NCK to PLC 65...96

nckServoDataAbsComp1stEnc32 (N, SD)				
Abs. compensation value meas. sys.1 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataAbsComp1stEnc64 (N, SD)				
Abs. compensation value meas. sys.1 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			

nckServoDataAbsComp1stEnc64 (N, SD)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataAbsComp2ndEnc32 (N, SD)				
Abs. compensation value meas. sys.2 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataAbsComp2ndEnc64 (N, SD)				
Abs. compensation value meas. sys.2 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActCurr32 (N, SD)				
Torque-prod. Current act. Val. i(q) (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
A	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActCurr64 (N, SD)				
Torque-prod. Current act. Val. i(q) (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
A	TYPE_DOUBLE			

nckServoDataActCurr64 (N, SD)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActEnc32 (N, SD)				
Active measuring system (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActEnc64 (N, SD)				
Active measuring system (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActPos1stEnc32 (N, SD)				
Position actual value meas.system 1 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActPos1stEnc64 (N, SD)				
Position actual value meas.system 1 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			

nckServoDataActPos1stEnc64 (N, SD)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActPos2ndEnc32 (N, SD)				
Position actual value meas.system 2 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActPos2ndEnc64 (N, SD)				
Position actual value meas.system 2 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActPower32 (N, SD)				
Active power (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
W	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActPower64 (N, SD)				
Active power (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
W	TYPE_DOUBLE			

nckServoDataActPower64 (N, SD)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActPressA32 (N, SD)				
Act. pressure cylinder drive end (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActPressA64 (N, SD)				
Act. pressure cylinder drive end (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActPressB32 (N, SD)				
Act.pressure cylinder non-drive end (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActPressB64 (N, SD)				
Act.pressure cylinder non-drive end (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

nckServoDataActPressB64 (N, SD)				
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActValveLift32 (N, SD)				
Valve lift act. value (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
V	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActValveLift64 (N, SD)				
Valve lift act. value (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
V	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActVel1stEnc32 (N, SD)				
Velocity act. value meas. system 1 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActVel1stEnc64 (N, SD)				
Velocity act. value meas. system 1 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

nckServoDataActVel1stEnc64 (N, SD)				
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActVel2ndEnc32 (N, SD)				
Velocity act. value meas. system 2 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActVel2ndEnc64 (N, SD)				
Velocity act. value meas. system 2 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActVelActiveEnc32 (N, SD)				
Velocity act. value active encoder (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActVelActiveEnc64 (N, SD)				
Velocity act. value active encoder (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

nckServoDataActVelActiveEnc64 (N, SD)				
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataActVelMot32 (N, SD)				
Speed / velocity act. Value motor (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataActVelMot64 (N, SD)				
Speed / velocity act. Value motor (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCmdAccCtrlIn32 (N, SD)				
Acceleration setpt controller input (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
m/s2, 1000 inch/ s2, rev/ s2, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCmdAccCtrlIn64 (N, SD)				
Acceleration setpt controller input (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
m/s2, 1000 inch/ s2, rev/ s2, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

nckServoDataCmdAccCtrlIn64 (N, SD)				
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCmdPos2ndEnc32 (N, SD)				
Position setpoint (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCmdPos2ndEnc64 (N, SD)				
Position setpoint (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCmdPosCtrlIn32 (N, SD)				
Position setpoint controller input (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCmdPosCtrlIn64 (N, SD)				
Position setpoint controller input (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

nckServoDataCmdPosCtrlIn64 (N, SD)				
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCmdTorque32 (N, SD)				
Torque / force setpoint (limited) (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
Nm	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCmdTorque64 (N, SD)				
Torque / force setpoint (limited) (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
Nm	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCmdValveLift32 (N, SD)				
Valve lift setpoint (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
V	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCmdValveLift64 (N, SD)				
Valve lift setpoint (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
V	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCmdVelCtrlIn32 (N, SD)				
Velocity setpoint controller input (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCmdVelCtrlIn64 (N, SD)				
Velocity setpoint controller input (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataContDev32 (N, SD)				
Contour deviation (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataContDev64 (N, SD)				
Contour deviation (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCtrlDev32 (N, SD)				
Control deviation (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCtrlDev64 (N, SD)				
Control deviation (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCtrlMode32 (N, SD)				
Controller mode (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCtrlMode64 (N, SD)				
Controller mode (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataDrvCmdVel32 (N, SD)				
Velocity setpoint drive (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataDrvCmdVel64 (N, SD)				
Velocity setpoint drive (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataTorqueBalanceCtrl32 (N, SD)				
Master/slave balance, torque bal.ctroller (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataTorqueBalanceCtrl64 (N, SD)				
Master/slave balance, torque bal.ctroller (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataTensionTorque32 (N, SD)				
Master-slave tension torque (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataTensionTorque64 (N, SD)				
Master-slave tension torque (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCmdPosNodComp32 (N, SD)				
Position setpoint of pitch compensation (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCmdPosNodComp64 (N, SD)				
Position setpoint of pitch compensation (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCmdVelNodComp32 (N, SD)				
Velocity setpoint of pitch compensation (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCmdVelNodComp64 (N, SD)				
Velocity setpoint of pitch compensation (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataCmdAccNodComp32 (N, SD)				
Acceleration setpt. of pitch compensation (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
m/s2, 1000 inch/ s2, rev/ s2, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataCmdAccNodComp64 (N, SD)				
Acceleration setpt. of pitch compensation (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
m/s2, 1000 inch/ s2, rev/ s2, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataDrvLoad32 (N, SD)				
Load (m_set/m_set,limit) (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataDrvLoad64 (N, SD)				
Load (m_set/m_set,limit) (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataFfwTorque32 (N, SD)				
Torque/Force feedforward value (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataFfwTorque64 (N, SD)				
Torque/Force feedforward value (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataFfwVelQEC32 (N, SD)				
Velocity feedfwd. value (plus QEC) (32 bit)				

nckServoDataFfwVelQEC32 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataFfwVelQEC64 (N, SD)				
Velocity feedfwd. value (plus QEC) (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataFlwErr32 (N, SD)				
Following Error (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataFlwErr64 (N, SD)				
Following Error (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataLimTorque32 (N, SD)				
Torque/force limit value (32 bit)				

nckServoDataLimTorque32 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataLimTorque64 (N, SD)				
Torque/force limit value (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataParamSet32 (N, SD)				
Parameter set (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataParamSet64 (N, SD)				
Parameter set (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataQecCompVal32 (N, SD)				
QEC compensation value (32 bit)				
Units and value ranges				
Physical unit:	Data type:			

nckServoDataQecCompVal32 (N, SD)				
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataQecCompVal64 (N, SD)				
QEC compensation value (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataQecLearnCrit32 (N, SD)				
QEC learning criteria (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataQecLearnCrit64 (N, SD)				
QEC learning criteria (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeActPos32 (N, SD)				
Safe actual position (32 bit)				
Units and value ranges				
Physical unit:	Data type:			

nckServoDataSafeActPos32 (N, SD)				
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeActPos64 (N, SD)				
Safe actual position (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeActPosDrv32 (N, SD)				
Safe drive actual position (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeActPosDrv64 (N, SD)				
Safe drive actual position (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeActValDiff32 (N, SD)				
SI actual value difference (32 bit)				
Units and value ranges				
Physical unit:	Data type:			

nckServoDataSafeActValDiff32 (N, SD)				
mm, inch, degree, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeActValDiff64 (N, SD)				
SI actual value difference (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeActVelDiff32 (N, SD)				
Current SI slip speed (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeActVelDiff64 (N, SD)				
Current SI slip speed (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeLimActVel32 (N, SD)				
Actual velocity limit (32 bit)				
Units and value ranges				
Physical unit:	Data type:			

nckServoDataSafeLimActVel32 (N, SD)				
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeLimActVel64 (N, SD)				
Actual velocity limit (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeLimCmdVel32 (N, SD)				
Set velocity limit (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeLimCmdVel64 (N, SD)				
Set velocity limit (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeLimSBR32 (N, SD)				
Current SBR limit value (32 bit)				
Units and value ranges				
Physical unit:	Data type:			

nckServoDataSafeLimSBR32 (N, SD)				
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeLimSBR64 (N, SD)				
Current SBR limit value (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafePartActVal32 (N, SD)				
Safety part actual value (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafePartActVal64 (N, SD)				
Safety part actual value (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst1Drv32 (N, SD)				
Results list 1 drive (32 bit)				
Units and value ranges				
Physical unit:	Data type:			

nckServoDataSafeResLst1Drv32 (N, SD)				
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst1Drv64 (N, SD)				
Results list 1 drive (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst1Nck32 (N, SD)				
Results list 1 NCK (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst1Nck64 (N, SD)				
Results list 1 NCK (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst2Drv32 (N, SD)				
Results list 2 drive (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	

nckServoDataSafeResLst2Drv32 (N, SD)		
	$$maxnumGlobMachAxes	
	Read access	

nckServoDataSafeResLst2Drv64 (N, SD)				
Results list 2 drive (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst2Nck32 (N, SD)				
Results list 2 NCK (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst2Nck64 (N, SD)				
Results list 2 NCK (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeRespIdDrv32 (N, SD)				
Response identifier for NCK/drive (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeRespIdDrv64 (N, SD)				
Response identifier for NCK/drive (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeRespIdNck32 (N, SD)				
Response ID for NCK (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeRespIdNck64 (N, SD)				
Response ID for NCK (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeSgaDrv32 (N, SD)				
SGA drive (to PLC) (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeSgaDrv64 (N, SD)				
SGA drive (to PLC) (64 bit)				

nckServoDataSafeSgaDrv64 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeSgaNck32 (N, SD)				
SGA NCK (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeSgaNck64 (N, SD)				
SGA NCK (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeSgeDrv32 (N, SD)				
SGE drive (from PLC) (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeSgeDrv64 (N, SD)				
SGE drive (from PLC) (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

nckServoDataSafeSgeDrv64 (N, SD)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeSgeNck32 (N, SD)				
SGE NCK (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeSgeNck64 (N, SD)				
SGE NCK (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSigExStopCoarse32 (N, SD)				
Signal exact stop coarse (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSigExStopCoarse64 (N, SD)				
Signal exact stop coarse (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

nckServoDataSigExStopCoarse64 (N, SD)		
	$$maxnumGlobMachAxes+1000	
	Read access	

nckServoDataSigExStopFine32 (N, SD)				
Signal exact stop fine (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSigExStopFine64 (N, SD)				
Signal exact stop fine (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSigIpoEnd32 (N, SD)				
Signal interpolation completed (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSigIpoEnd64 (N, SD)				
Signal interpolation completed (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

servoDatFl32 (N, SD)				
Servo data 32-bit Servo data column: Signal ID / Servo cycle (see module header)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_FLOAT	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

servoDatFl64 (N, SD)				
Servo data 64-bit Servo data column: Signal ID / Servo cycle (see module header)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataSafeInseNck1_32 (N, SD)				
ext. NCK-SPL inputs 1...32 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeInseNck1_64 (N, SD)				
ext. NCK-SPL inputs 1...32 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInseNck2_32 (N, SD)				
ext. NCK-SPL inputs 32...64 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeInseNck2_64 (N, SD)				
ext. NCK-SPL inputs 32...64 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutseNck1_32 (N, SD)				
ext. NCK-SPL outputs 1...32 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeOutseNck1_64 (N, SD)				
ext. NCK-SPL outputs 1...32 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutseNck2_32 (N, SD)				
ext. NCK-SPL outputs 33...64 (32 bits)				

nckServoDataSafeOutseNck2_32 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeOutseNck2_64 (N, SD)				
ext. NCK-SPL outputs 33...64 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiNck1_32 (N, SD)				
int. NCK-SPL inputs 1...32 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeInsiNck1_64 (N, SD)				
int. NCK-SPL inputs 1...32 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiNck2_32 (N, SD)				
int. NCK-SPL inputs 33...64 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			

nckServoDataSafeInsiNck2_32 (N, SD)		
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

nckServoDataSafeInsiNck2_64 (N, SD)				
int. NCK-SPL inputs 33...64 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeOutsiNck1_32 (N, SD)				
int. NCK-SPL outputs 1...32 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:		Max. line index:		
		1		
		Read access		

nckServoDataSafeOutsiNck1_64 (N, SD)				
int. NCK-SPL outputs 1...32 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeOutsiNck2_32 (N, SD)				
int. NCK-SPL outputs 33...64 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:		Max. line index:		

nckServoDataSafeOutsiNck2_32 (N, SD)				
			1	
			Read access	

nckServoDataSafeOutsiNck2_64 (N, SD)				
int. NCK-SPL outputs 33...64 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerNck1_32 (N, SD)				
NCK SPL marker 1...32 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeMarkerNck1_64 (N, SD)				
NCK SPL marker 1...32 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerNck2_32 (N, SD)				
NCK SPL marker 33...64 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeMarkerNck2_64 (N, SD)				
NCK SPL marker 33...64 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer1_32 (N, SD)				
NCK-SPL timer 1 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer1_64 (N, SD)				
NCK-SPL timer 1 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer2_32 (N, SD)				
NCK-SPL timer 2 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer2_64 (N, SD)				
NCK-SPL timer 2 (64 bits)				

nckServoDataSafeTimer2_64 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer3_32 (N, SD)				
NCK-SPL timer 3 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer3_64 (N, SD)				
NCK-SPL timer 3 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer4_32 (N, SD)				
NCK-SPL timer 4 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer4_64 (N, SD)				
NCK-SPL timer 4 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

nckServoDataSafeTimer4_64 (N, SD)		
Addressing		
Line index:	Max. line index:	
	1000+1	
	Read access	

nckServoDataSafeTimer5_32 (N, SD)				
NCK-SPL timer 5 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer5_64 (N, SD)				
NCK-SPL timer 5 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer6_32 (N, SD)				
NCK-SPL timer 6 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer6_64 (N, SD)				
NCK-SPL timer 6 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

nckServoDataSafeTimer6_64 (N, SD)				
			1000+1	
			Read access	

nckServoDataSafeTimer7_32 (N, SD)				
NCK-SPL timer 7 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer7_64 (N, SD)				
NCK-SPL timer 7 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer8_32 (N, SD)				
NCK-SPL timer 8 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer8_64 (N, SD)				
NCK-SPL timer 8 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsePlc1_32 (N, SD)				
ext. PLC-SPL inputs 1...32 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeInsePlc1_64 (N, SD)				
ext. PLC-SPL inputs 1...32 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsePlc2_32 (N, SD)				
ext. PLC-SPL inputs 33...64 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeInsePlc2_64 (N, SD)				
ext. PLC-SPL inputs 33...64 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsePlc1_32 (N, SD)				
ext. PLC-SPL outputs 1...32 (32 bits)				

nckServoDataSafeOutsePlc1_32 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeOutsePlc1_64 (N, SD)				
ext. PLC-SPL outputs 1...32 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsePlc2_32 (N, SD)				
ext. PLC-SPL outputs 33...64 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeOutsePlc2_64 (N, SD)				
ext. PLC-SPL outputs 33...64 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiPlc1_32 (N, SD)				
int. PLC-SPL inputs 1...32 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			

nckServoDataSafeInsiPlc1_32 (N, SD)				
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeInsiPlc1_64 (N, SD)				
int. PLC-SPL inputs 1...32 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiPlc2_32 (N, SD)				
int. PLC-SPL inputs 33...64 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeInsiPlc2_64 (N, SD)				
int. PLC-SPL inputs 33...64 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiPlc1_32 (N, SD)				
int. PLC-SPL outputs 1...32 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	

nckServoDataSafeOutsiPlc1_32 (N, SD)				
			1	
			Read access	

nckServoDataSafeOutsiPlc1_64 (N, SD)				
int. PLC-SPL outputs 1...32 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiPlc2_32 (N, SD)				
int. PLC-SPL outputs 33...64 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeOutsiPlc2_64 (N, SD)				
int. PLC-SPL outputs 33...64 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerPlc1_32 (N, SD)				
PLC SPL marker 1...32 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeMarkerPlc1_64 (N, SD)				
PLC SPL marker 1...32 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerPlc2_32 (N, SD)				
PLC SPL marker 33...64 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeMarkerPlc2_64 (N, SD)				
PLC SPL marker 33...64 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer9_32 (N, SD)				
NCK-SPL timer 9 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer9_64 (N, SD)				
NCK-SPL timer 9 (64 bits)				

nckServoDataSafeTimer9_64 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer10_32 (N, SD)				
NCK-SPL timer 10 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer10_64 (N, SD)				
NCK-SPL timer 10 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer11_32 (N, SD)				
NCK-SPL timer 11 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer11_64 (N, SD)				
NCK-SPL timer 11 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

nckServoDataSafeTimer11_64 (N, SD)		
Addressing		
Line index:	Max. line index:	
	1000+1	
	Read access	

nckServoDataSafeTimer12_32 (N, SD)				
NCK-SPL timer 12 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer12_64 (N, SD)				
NCK-SPL timer 12 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer13_32 (N, SD)				
NCK-SPL timer 13 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer13_64 (N, SD)				
NCK-SPL timer 13 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

nckServoDataSafeTimer13_64 (N, SD)				
			1000+1	
			Read access	

nckServoDataSafeTimer14_32 (N, SD)				
NCK-SPL timer 14 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer14_64 (N, SD)				
NCK-SPL timer 14 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer15_32 (N, SD)				
NCK-SPL timer 15 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer15_64 (N, SD)				
NCK-SPL timer 15 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeTimer16_32 (N, SD)				
NCK-SPL timer 16 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

nckServoDataSafeTimer16_64 (N, SD)				
NCK-SPL timer 16 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeNckCam32 (N, SD)				
Cam SGA NCK (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeNckCam64 (N, SD)				
Cam SGA NCK (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeDrvSga32 (N, SD)				
SGA drive (32 bits)				

nckServoDataSafeDrvSga32 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeDrvSga64 (N, SD)				
SGA drive (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeDrvCam32 (N, SD)				
Cam SGA drive (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeDrvCam64 (N, SD)				
Cam SGA drive (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeFineRawDiff32 (N, SD)				
Act. value difference fine - coarse pos. (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			

nckServoDataSafeFineRawDiff32 (N, SD)		
Addressing		
Line index:	Max. line index:	
	$$maxnumGlobMachAxes	
	Read access	

nckServoDataSafeFineRawDiff64 (N, SD)				
Act. value difference fine - coarse pos. (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst3Nck32 (N, SD)				
Results list 3 NCK (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst3Nck64 (N, SD)				
Results list 3 NCK (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst3Drv32 (N, SD)				
Results list 3 drive (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	

nckServoDataSafeResLst3Drv32 (N, SD)				
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst3Drv64 (N, SD)				
Results list 3 drive (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst4Nck32 (N, SD)				
Results list 4 NCK (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst4Nck64 (N, SD)				
Results list 4 NCK (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst4Drv32 (N, SD)				
Results list 4 drive (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst4Drv64 (N, SD)				
Results list 4 drive (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst5Nck32 (N, SD)				
Results list 5 NCK (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst5Nck64 (N, SD)				
Results list 5 NCK (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst5Drv32 (N, SD)				
Results list 5 drive (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst5Drv64 (N, SD)				
Results list 5 drive (64 bit)				

nckServoDataSafeResLst5Drv64 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst6Nck32 (N, SD)				
Results list 6 NCK (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst6Nck64 (N, SD)				
Results list 6 NCK (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeResLst6Drv32 (N, SD)				
Results list 6 drive (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeResLst6Drv64 (N, SD)				
Results list 6 drive (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

nckServoDataSafeResLst6Drv64 (N, SD)		
Addressing		
Line index:	Max. line index:	
	$$maxnumGlobMachAxes+1000	
	Read access	

nckServoDataSafeResLst7Nck32 (N, SD)				
Results list 7 NCK (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:		Max. line index:		
		$$maxnumGlobMachAxes		
		Read access		

nckServoDataSafeResLst7Nck64 (N, SD)				
Results list 7 NCK (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		$$maxnumGlobMachAxes+1000		
		Read access		

nckServoDataSafeResLst7Drv32 (N, SD)				
Results list 7 drive (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:		Max. line index:		
		$$maxnumGlobMachAxes+1000		
		Read access		

nckServoDataSafeResLst7Drv64 (N, SD)				
Results list 7 drive (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		

nckServoDataSafeResLst7Drv64 (N, SD)				
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataSafeLoadVelFilt32 (N, SD)				
Smoothed actual velocity value NCK (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

nckServoDataSafeLoadVelFilt64 (N, SD)				
Smoothed actual velocity value NCK (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes+1000	
			Read access	

nckServoDataOemId200 (N, SD)				
Compile cycle signal 200 Servo data Compile cycle signal 200				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId201 (N, SD)				
Compile cycle signal 201 Servo data Compile cycle signal 201				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

nckServoDataOemId201 (N, SD)				
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId202 (N, SD)				
Compile cycle signal 202 Servo data Compile cycle signal 202				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId203 (N, SD)				
Compile cycle signal 203 Servo data Compile cycle signal 203				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId204 (N, SD)				
Compile cycle signal 204 Servo data Compile cycle signal 204				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId205 (N, SD)				
Compile cycle signal 205 Servo data Compile cycle signal 205				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId206 (N, SD)				
Compile cycle signal 206 Servo data Compile cycle signal 206				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId207 (N, SD)				
Compile cycle signal 207 Servo data Compile cycle signal 207				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId208 (N, SD)				
Compile cycle signal 208 Servo data Compile cycle signal 208				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

nckServoDataOemId208 (N, SD)				
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId209 (N, SD)				
Compile cycle signal 209 Servo data Compile cycle signal 209				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId210 (N, SD)				
Compile cycle signal 210 Servo data Compile cycle signal 210				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId211 (N, SD)				
Compile cycle signal 211 Servo data Compile cycle signal 211				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId212 (N, SD)				
Compile cycle signal 212 Servo data Compile cycle signal 212				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId213 (N, SD)				
Compile cycle signal 213 Servo data Compile cycle signal 213				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId214 (N, SD)				
Compile cycle signal 214 Servo data Compile cycle signal 214				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId215 (N, SD)				
Compile cycle signal 215 Servo data Compile cycle signal 215				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

nckServoDataOemId215 (N, SD)				
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId216 (N, SD)				
Compile cycle signal 216 Servo data Compile cycle signal 216				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId217 (N, SD)				
Compile cycle signal 217 Servo data Compile cycle signal 217				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId218 (N, SD)				
Compile cycle signal 218 Servo data Compile cycle signal 218				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId219 (N, SD)				
Compile cycle signal 219 Servo data Compile cycle signal 219				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId220 (N, SD)				
Compile cycle signal 220 Servo data Compile cycle signal 220				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId221 (N, SD)				
Compile cycle signal 221 Servo data Compile cycle signal 221				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId222 (N, SD)				
Compile cycle signal 222 Servo data Compile cycle signal 222				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

nckServoDataOemId222 (N, SD)				
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId223 (N, SD)				
Compile cycle signal 223 Servo data Compile cycle signal 223				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId224 (N, SD)				
Compile cycle signal 224 Servo data Compile cycle signal 224				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId225 (N, SD)				
Compile cycle signal 225 Servo data Compile cycle signal 225				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId226 (N, SD)				
Compile cycle signal 226 Servo data Compile cycle signal 226				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId227 (N, SD)				
Compile cycle signal 227 Servo data Compile cycle signal 227				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId228 (N, SD)				
Compile cycle signal 228 Servo data Compile cycle signal 228				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId229 (N, SD)				
Compile cycle signal 229 Servo data Compile cycle signal 229				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

nckServoDataOemId229 (N, SD)				
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId230 (N, SD)				
Compile cycle signal 230 Servo data Compile cycle signal 230				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId231 (N, SD)				
Compile cycle signal 231 Servo data Compile cycle signal 231				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId232 (N, SD)				
Compile cycle signal 232 Servo data Compile cycle signal 232				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId233 (N, SD)				
Compile cycle signal 233 Servo data Compile cycle signal 233				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId234 (N, SD)				
Compile cycle signal 234 Servo data Compile cycle signal 234				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId235 (N, SD)				
Compile cycle signal 235 Servo data Compile cycle signal 235				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId236 (N, SD)				
Compile cycle signal 236 Servo data Compile cycle signal 236				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

nckServoDataOemId236 (N, SD)				
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId237 (N, SD)				
Compile cycle signal 237 Servo data Compile cycle signal 237				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId238 (N, SD)				
Compile cycle signal 238 Servo data Compile cycle signal 238				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId239 (N, SD)				
Compile cycle signal 239 Servo data Compile cycle signal 239				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId240 (N, SD)				
Compile cycle signal 240 Servo data Compile cycle signal 240				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId241 (N, SD)				
Compile cycle signal 241 Servo data Compile cycle signal 241				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId242 (N, SD)				
Compile cycle signal 242 Servo data Compile cycle signal 242				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId243 (N, SD)				
Compile cycle signal 243 Servo data Compile cycle signal 243				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

nckServoDataOemId243 (N, SD)				
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId244 (N, SD)				
Compile cycle signal 244 Servo data Compile cycle signal 244				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId245 (N, SD)				
Compile cycle signal 245 Servo data Compile cycle signal 245				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId246 (N, SD)				
Compile cycle signal 246 Servo data Compile cycle signal 246				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId247 (N, SD)				
Compile cycle signal 247 Servo data Compile cycle signal 247				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId248 (N, SD)				
Compile cycle signal 248 Servo data Compile cycle signal 248				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId249 (N, SD)				
Compile cycle signal 249 Servo data Compile cycle signal 249				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId250 (N, SD)				
Compile cycle signal 250 Servo data Compile cycle signal 250				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

nckServoDataOemId250 (N, SD)				
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId251 (N, SD)				
Compile cycle signal 251 Servo data Compile cycle signal 251				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId252 (N, SD)				
Compile cycle signal 252 Servo data Compile cycle signal 252				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId253 (N, SD)				
Compile cycle signal 253 Servo data Compile cycle signal 253				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId254 (N, SD)				
Compile cycle signal 254 Servo data Compile cycle signal 254				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataOemId255 (N, SD)				
Compile cycle signal 255 Servo data Compile cycle signal 255				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			siehe Bausteinkopf	
			Read access	

nckServoDataSafeInseNck3_32 (N, SD)				
ext. NCK-SPL inputs 65...96 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInseNck3_64 (N, SD)				
ext. NCK-SPL inputs 65...96 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInseNck4_32 (N, SD)				
ext. NCK-SPL inputs 97...128 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInseNck4_64 (N, SD)				
ext. NCK-SPL inputs 97...128 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInseNck5_32 (N, SD)				
ext. NCK-SPL inputs 129...160 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInseNck5_64 (N, SD)				
ext. NCK-SPL inputs 129...160 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInseNck6_32 (N, SD)				
ext. NCK-SPL inputs 161...192 (32 bits)				

nckServoDataSafeInseNck6_32 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInseNck6_64 (N, SD)				
ext. NCK-SPL inputs 161...192 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutseNck3_32 (N, SD)				
ext. NCK-SPL outputs 65...96 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutseNck3_64 (N, SD)				
ext. NCK-SPL outputs 65...96 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutseNck4_32 (N, SD)				
ext. NCK-SPL outputs 97...128 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			

nckServoDataSafeOutseNck4_32 (N, SD)		
Addressing		
Line index:	Max. line index:	
	1000+1	
	Read access	

nckServoDataSafeOutseNck4_64 (N, SD)				
ext. NCK-SPL outputs 97...128 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeOutseNck5_32 (N, SD)				
ext. NCK-SPL outputs 129...160 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeOutseNck5_64 (N, SD)				
ext. NCK-SPL outputs 129...160 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeOutseNck6_32 (N, SD)				
ext. NCK-SPL outputs 161...192 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:		Max. line index:		

nckServoDataSafeOutseNck6_32 (N, SD)				
			1000+1	
			Read access	

nckServoDataSafeOutseNck6_64 (N, SD)				
ext. NCK-SPL outputs 161...192 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiNck3_32 (N, SD)				
int. NCK-SPL inputs 65...96 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiNck3_64 (N, SD)				
int. NCK-SPL inputs 65...96 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiNck4_32 (N, SD)				
int. NCK-SPL inputs 97...128 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiNck4_64 (N, SD)				
int. NCK-SPL inputs 97...128 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiNck5_32 (N, SD)				
int. NCK-SPL inputs 129...160 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiNck5_64 (N, SD)				
int. NCK-SPL inputs 129...160 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiNck6_32 (N, SD)				
int. NCK-SPL inputs 161...192 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiNck6_64 (N, SD)				
int. NCK-SPL inputs 161...192 (64 bits)				

nckServoDataSafeInsiNck6_64 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiNck3_32 (N, SD)				
int. NCK-SPL outputs 65...96 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiNck3_64 (N, SD)				
int. NCK-SPL outputs 65...96 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiNck4_32 (N, SD)				
int. NCK-SPL outputs 97...128 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiNck4_64 (N, SD)				
int. NCK-SPL outputs 97...128 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

nckServoDataSafeOutsiNck4_64 (N, SD)		
Addressing		
Line index:	Max. line index:	
	1000+1	
	Read access	

nckServoDataSafeOutsiNck5_32 (N, SD)				
int. NCK-SPL outputs 129...160 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeOutsiNck5_64 (N, SD)				
int. NCK-SPL outputs 129...160 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeOutsiNck6_32 (N, SD)				
int. NCK-SPL outputs 161...192 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeOutsiNck6_64 (N, SD)				
int. NCK-SPL outputs 161...192 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		

nckServoDataSafeOutsiNck6_64 (N, SD)				
			1000+1	
			Read access	

nckServoDataSafeMarkerNck3_32 (N, SD)				
NCK SPL marker 65...96 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerNck3_64 (N, SD)				
NCK SPL marker 65...96 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerNck4_32 (N, SD)				
NCK SPL marker 97...128 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerNck4_64 (N, SD)				
NCK SPL marker 97...128 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerNck5_32 (N, SD)				
NCK SPL marker 129...160 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerNck5_64 (N, SD)				
NCK SPL marker 129...160 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerNck6_32 (N, SD)				
NCK SPL marker 161...192 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerNck6_64 (N, SD)				
NCK SPL marker 161...192 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsePlc3_32 (N, SD)				
ext. PLC-SPL inputs 65...96 (32 bits)				

nckServoDataSafeInsePlc3_32 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsePlc3_64 (N, SD)				
ext. PLC-SPL inputs 65...96 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsePlc4_32 (N, SD)				
ext. PLC-SPL inputs 97...128 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsePlc4_64 (N, SD)				
ext. PLC-SPL inputs 97...128 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsePlc5_32 (N, SD)				
ext. PLC-SPL inputs 129...160 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			

nckServoDataSafeInsePlc5_32 (N, SD)		
Addressing		
Line index:	Max. line index:	
	1000+1	
	Read access	

nckServoDataSafeInsePlc5_64 (N, SD)				
ext. PLC-SPL inputs 129...160 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeInsePlc6_32 (N, SD)				
ext. PLC-SPL inputs 161...192 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeInsePlc6_64 (N, SD)				
ext. PLC-SPL inputs 161...192 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		1000+1		
		Read access		

nckServoDataSafeOutsePlc3_32 (N, SD)				
ext. PLC-SPL outputs 65...96 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:		Max. line index:		

nckServoDataSafeOutsePlc3_32 (N, SD)				
			1000+1	
			Read access	

nckServoDataSafeOutsePlc3_64 (N, SD)				
ext. PLC-SPL outputs 65...96 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsePlc4_32 (N, SD)				
ext. PLC-SPL outputs 97...128 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsePlc4_64 (N, SD)				
ext. PLC-SPL outputs 97...128 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsePlc5_32 (N, SD)				
ext. PLC-SPL outputs 129...160 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsePlc5_64 (N, SD)				
ext. PLC-SPL outputs 129...160 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsePlc6_32 (N, SD)				
ext. PLC-SPL outputs 161...192 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsePlc6_64 (N, SD)				
ext. PLC-SPL outputs 161...192 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiPlc3_32 (N, SD)				
int. PLC-SPL inputs 65...96 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiPlc3_64 (N, SD)				
int. PLC-SPL inputs 65...96 (64 bits)				

nckServoDataSafeInsiPlc3_64 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiPlc4_32 (N, SD)				
int. PLC-SPL inputs 97...128 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiPlc4_64 (N, SD)				
int. PLC-SPL inputs 97...128 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiPlc5_32 (N, SD)				
int. PLC-SPL inputs 129...160 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiPlc5_64 (N, SD)				
int. PLC-SPL inputs 129...160 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

nckServoDataSafeInsiPlc5_64 (N, SD)		
Addressing		
Line index:	Max. line index:	
	1000+1	
	Read access	

nckServoDataSafeInsiPlc6_32 (N, SD)				
int. PLC-SPL inputs 161...192 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeInsiPlc6_64 (N, SD)				
int. PLC-SPL inputs 161...192 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiPlc3_32 (N, SD)				
int. PLC-SPL outputs 65...96 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiPlc3_64 (N, SD)				
int. PLC-SPL outputs 65...96 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

nckServoDataSafeOutsiPlc3_64 (N, SD)				
			1000+1	
			Read access	

nckServoDataSafeOutsiPlc4_32 (N, SD)				
int. PLC-SPL outputs 97...128 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiPlc4_64 (N, SD)				
int. PLC-SPL outputs 97...128 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiPlc5_32 (N, SD)				
int. PLC-SPL outputs 129...160 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiPlc5_64 (N, SD)				
int. PLC-SPL outputs 129...160 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiPlc6_32 (N, SD)				
int. PLC-SPL outputs 161...192 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeOutsiPlc6_64 (N, SD)				
int. PLC-SPL outputs 161...192 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerPlc3_32 (N, SD)				
PLC SPL marker 65...96 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerPlc3_64 (N, SD)				
PLC SPL marker 65...96 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerPlc4_32 (N, SD)				
PLC SPL marker 97...128 (32 bit)				

nckServoDataSafeMarkerPlc4_32 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerPlc4_64 (N, SD)				
PLC SPL marker 97...128 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerPlc5_32 (N, SD)				
PLC SPL marker 129...160 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerPlc5_64 (N, SD)				
PLC SPL marker 129...160 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafeMarkerPlc6_32 (N, SD)				
PLC SPL marker 161...192 (32 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			

nckServoDataSafeMarkerPlc6_32 (N, SD)		
Addressing		
Line index:	Max. line index:	
	1000+1	
	Read access	

nckServoDataSafeMarkerPlc6_64 (N, SD)				
PLC SPL marker 161...192 (64 bit)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcIn1_32 (N, SD)				
Single-chan. SI sig. PLC to NCK 1...32 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcIn1_64 (N, SD)				
Single-chan. SI sig. PLC to NCK 1...32 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcIn2_32 (N, SD)				
Single-chan. SI sig. PLC to NCK 33...64 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	

nckServoDataSafePlcIn2_32 (N, SD)				
			1000+1	
			Read access	

nckServoDataSafePlcIn2_64 (N, SD)				
Single-chan. SI sig. PLC to NCK 33...64 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcIn3_32 (N, SD)				
Single-chan. SI sig. PLC to NCK 65...96 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcIn3_64 (N, SD)				
Single-chan. SI sig. PLC to NCK 65...96 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcOut1_32 (N, SD)				
Single-chan. SI sig. NCK to PLC 1...32 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcOut1_64 (N, SD)				
Single-chan. SI sig. NCK to PLC 1...32 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcOut2_32 (N, SD)				
Single-chan. SI sig. NCK to PLC 33...64 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcOut2_64 (N, SD)				
Single-chan. SI sig. NCK to PLC 33...64 (64 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcOut3_32 (N, SD)				
Single-chan. SI sig. NCK to PLC 65...96 (32 bits)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

nckServoDataSafePlcOut3_64 (N, SD)				
Single-chan. SI sig. NCK to PLC 65...96 (64 bits)				

nckServoDataSafePlcOut3_64 (N, SD)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			1000+1	
			Read access	

3.2.6.19 SE: Setting data

This module contains all global setting data. The physical units depend on the variable "$$userScale" in module Y of area N.

CEC_TABLE_ENABLE (N, SE)		SD 41300: $SN_CEC_TABLE_ENABLE		
SD 41300: $SN_CEC_TABLE_ENABLE				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_BOOLEAN			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

CEC_TABLE_WEIGHT (N, SE)		SD 41310: $SN_CEC_TABLE_WEIGHT		
SD 41310: $SN_CEC_TABLE_WEIGHT				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

COMPAR_THRESHOLD_1 (N, SE)		SD 41600: $SN_COMPAR_THRESH-OLD_1		
SD 41600: $SN_COMPAR_THRESHOLD_1				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

COMPAR_THRESHOLD_1 (N, SE)		**SD 41600: $SN_COMPAR_THRESHOLD_1**		
			2	
			Read access	

COMPAR_THRESHOLD_2 (N, SE)		**SD 41601: $SN_COMPAR_THRESHOLD_2**		
SD 41601: $SN_COMPAR_THRESHOLD_2				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

JOG_ROT_AX_SET_VELO (N, SE)		**SD 41130: $SN_JOG_ROT_AX_SET_VELO**		
SD 41130: $SN_JOG_ROT_AX_SET_VELO				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
			Read access	

MDB_JOG_CONT_MODE_LEVELTRIGGRD (N, SE)		**SD 41050: $SN_MDB_JOG_CONT_MODE_LEVELTRIGGRD**		
Alternative name:	JOG_CONT_MODE_LEVELTRIGGRD			
Jog mode				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Validity:	From NCK version 3.6		Read access and write access	

MDB_JOG_REV_IS_ACTIVE (N, SE)		**SD 41100: $SN_MDB_JOG_REV_IS_ACTIVE**		
Alternative name:	JOG_REV_IS_ACTIVE			
JOG at revolutional feedrate				
Return value	0 = G94 1 = G95			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_CHAR			
Validity:	From NCK version 3.6		Read access and write access	

MDD_JOG_REV_SET_VELO (N, SE)		SD 41120: $SN_MDD_JOG_REV_SET_VELO		
Alternative name:	JOG_REV_SET_VELO			
JOG velocity for G95				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access and write access	

MDD_JOG_SET_VELO (N, SE)		SD 41110: $SN_MDD_JOG_SET_VELO		
Alternative name:	JOG_SET_VELO			
JOG velocity for G94				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access and write access	

MDD_JOG_SPIND_SET_VELO (N, SE)		SD 41200: $SN_MDD_JOG_SPIND_SET_VELO		
Alternative name:	JOG_SPIND_SET_VELO			
JOG velocity for master spindle				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access and write access	

MDD_JOG_VAR_INCR_SIZE (N, SE)		SD 41010: $SN_MDD_JOG_VAR_INCR_SIZE		
Alternative name:	JOG_VAR_INCR_SIZE			
Variable incremental value for JOG mode				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access and write access	

SW_CAM_MINUS_POS_TAB_1 (N, SE)		SD 41500: $SN_SW_CAM_MINUS_POS_TAB_1		
Trigger points at falling cam 1-8 The cam position of minus cams 1 - 8 is entered in the variable. The positions are entered in the machine coordinate system. The index [n] of the setting data addresses the cam pair: n = 0, 1, ... , 7 corresponds to cam pair 1, 2, ... , 8 If the set switching points are passed in the positive axis direction, the associated minus cam signals in the PLC interface (and any applied fast output signals) switch from 1 to 0.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_MINUS_POS_TAB_2 (N, SE)		SD 41502: $SN_SW_CAM_MINUS_POS_TAB_2		
Trigger points at falling cam edge 9-16 The cam position of minus cams 9 - 16 is entered in the variable. The positions are entered in the machine coordinate system. The index [n] of the setting data addresses the cam pair: n = 8, 9, ... , 15 corresponds to cam pair 9, 10, ... , 16 If the set switching points are passed in the positive axis direction, the associated minus cam signals in the PLC interface (and any applied fast output signals) switch from 1 to 0.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_MINUS_POS_TAB_3 (N, SE)		SD 41504: $SN_SW_CAM_MINUS_POS_TAB_3		
SD 41504: $SN_SW_CAM_MINUS_POS_TAB_3				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

SW_CAM_MINUS_POS_TAB_3 (N, SE)	SD 41504: $SN_SW_CAM_MINUS_POS_TAB_3	
	2	
	Read access	

SW_CAM_MINUS_POS_TAB_4 (N, SE)		SD 41506: $SN_SW_CAM_MINUS_POS_TAB_4		
SD 41506: $SN_SW_CAM_MINUS_POS_TAB_4				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_MINUS_TIME_TAB_1 (N, SE)		SD 41520: $SN_SW_CAM_MINUS_TIME_TAB_1		
SD 41520: $SN_SW_CAM_MINUS_TIME_TAB_1				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_MINUS_TIME_TAB_2 (N, SE)		SD 41522: $SN_SW_CAM_MINUS_TIME_TAB_2		
SD 41522: $SN_SW_CAM_MINUS_TIME_TAB_2				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_MINUS_TIME_TAB_3 (N, SE)		SD 41524: $SN_SW_CAM_MINUS_TIME_TAB_3		
SD 41524: $SN_SW_CAM_MINUS_TIME_TAB_3				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			

SW_CAM_MINUS_TIME_TAB_3 (N, SE)	SD 41524: $SN_SW_CAM_MINUS_TIME_TAB_3	
Addressing		
Line index:	Max. line index:	
	2	
	Read access	

SW_CAM_MINUS_TIME_TAB_4 (N, SE)		SD 41526: $SN_SW_CAM_MINUS_TIME_TAB_4		
SD 41526: $SN_SW_CAM_MINUS_TIME_TAB_4				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		2		
		Read access		

SW_CAM_PLUS_POS_TAB_1 (N, SE)		SD 41501: $SN_SW_CAM_PLUS_POS_TAB_1		
Trigger points at rising cam edge 1-8 The cam position of plus cams 1 - 8 is entered in the variable. The positions are entered in the machine coordinate system. The index [n] of the setting data addresses the cam pair: n = 0, 1, ... , 7 corresponds to cam pair 1, 2, ... , 8 If the set switching points are passed in the positive axis direction, the associated plus cam signals in the PLC interface (and any applied fast output signals) switch from 0 to 1.				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:		Max. line index:		
		2		
		Read access		

SW_CAM_PLUS_POS_TAB_2 (N, SE)		SD 41503: $SN_SW_CAM_PLUS_POS_TAB_2		
Trigger points at rising cam edge The cam position of plus cams 9 - 16 is entered in the variable. The positions are entered in the machine coordinate system. The index [n] of the setting data addresses the cam pair: n = 8, 9, ... , 15 corresponds to cam pair 9, 10, ... , 16 If the set switching points are passed in the positive axis direction, the associated plus cam signals in the PLC interface (and any applied fast output signals) switch from 1 to 0.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_PLUS_POS_TAB_3 (N, SE)		SD 41505: $SN_SW_CAM_PLUS_POS_TAB_3		
SD 41505: $SN_SW_CAM_PLUS_POS_TAB_3				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_PLUS_POS_TAB_4 (N, SE)		SD 41507: $SN_SW_CAM_PLUS_POS_TAB_4		
SD 41507: $SN_SW_CAM_PLUS_POS_TAB_4				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_PLUS_TIME_TAB_1 (N, SE)	SD 41521: $SN_SW_CAM_PLUS_TIME_TAB_1	
SD 41521: $SN_SW_CAM_PLUS_TIME_TAB_1		

SW_CAM_PLUS_TIME_TAB_1 (N, SE)		SD 41521: $SN_SW_CAM_PLUS_TIME_TAB_1		
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_PLUS_TIME_TAB_2 (N, SE)		SD 41523: $SN_SW_CAM_PLUS_TIME_TAB_2		
SD 41523: $SN_SW_CAM_PLUS_TIME_TAB_2				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_PLUS_TIME_TAB_3 (N, SE)		SD 41525: $SN_SW_CAM_PLUS_TIME_TAB_3		
SD 41525: $SN_SW_CAM_PLUS_TIME_TAB_3				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

SW_CAM_PLUS_TIME_TAB_4 (N, SE)		SD 41527: $SN_SW_CAM_PLUS_TIME_TAB_4		
SD 41527: $SN_SW_CAM_PLUS_TIME_TAB_4				
Units and value ranges				
Physical unit:	Data type:			
s, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			2	
			Read access	

3.2.6.20 SEMA: Extended state data in MCS

All state data that are dependent on machine movement and are defined specifically for machine axes (geometry and special axes) are combined in module SMA. Supplementary information is to be found in module SEMA. The individual variables are defined as fields where the line index is the number of the machine axis (assigned to the current channel). The variable "name" in module SMA with the line index in question identifies the axis.

The assignment of the line indices in modules SMA and SEMA is identical.

PRESETActive (N, SEMA)				
Preset state				
Return value	0 = no preset active 1 = preset active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

PRESETVal (N, SEMA)				
PRESET The function PRESETON (...) programs a work offset for an axis. The value of the offset is stored in the variable 'PRESETVal'. The variable can be overwritten by the part program and by the HMI.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access and write access	

aaAcc (N, SEMA)				
Acceleration value Current axial acceleration value				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
m/s2, 1000 inch/ s2, rev/ s2, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaAccPercent (N, SEMA)				
Current acceleration value in percent Current acceleration value for single-axis interpolation in percent				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaActIndexAxPosNo (N, SEMA)				
Current indexing position Current indexing position; the display depends on $MN_INDEX_AX_NO_MODE and the division (via table or equidistant)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaAlarmStat (N, SEMA)				
Display of active alarms Display indicating whether alarms are active for a PLC-controlled axis. The relevant coded alarm reactions can be used as a source for the "Extended Stop and Retract" function.				
Return value	The data is bit-coded, allowing, where necessary, individual states to be masked or evaluated separately (bits not listed supply a value of 0) Bit2 = 1: NOREADY (active rapid deceleration + cancelation of servo enable) Bit6 = 1: STOPBYALARM (rampm stop in all channel axes) Bit9 = 1: SETVDI (VDI interface signal "Setting alarm") Bit13 = 1: FOLLOWUPBYALARM (Follow-up)			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaAxChangeStat (N, SEMA)				
Axis status with respect to axis replacement Axis status with respect to axis replacement				
Return value	0: Axis can be replaced 1: Axis is linked to the channel, but can become the PLC, command or reciprocating axis 2: Axis cannot be replaced			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaAxChangeTyp (N, SEMA)				
Axis type with respect to axis replacement				
Return value	0: Axis assigned to the NC program 1: Axis assigned to the PLC or active as command axis or reciprocating axis 2: Other channel has interpolation right 3: Neutral axis 4: Neutral axis controlled from the PLC 5: Other channel has interpolation right; axis is requested for the NC program 6: Other channel has interpolation right; axis is requested as neutral axis 7: Axis is PLC axis or is active as command axis or reciprocating axis; axis is requested for the NC program 8: Axis is PLC axis or is active as command axis or reciprocating axis; axis is requested as neutral axis			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	8
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaBcsOffset (N, SEMA)				
Axial offsets of an axis Sum of all axial offsets of an axis, such as DRF, online tool offset, $AA_OFF and ext. WO.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

aaBcsOffset (N, SEMA)		
Addressing		
Line index:	Max. line index:	
	$$maxnumGlobMachAxes	
	Read access	

aaBrakeCondB (N, SEMA)				
Conditions for interpolator stop in the BCS Shows the pending braking requests (conditions) for the interpolator stop of the axis / spindle. A braking request consists of a collision direction relating to a coordinate axis in the BCS and a braking priority relating to the machining step. If the axis / spindle receives a current braking request on account of these requirement(s), bit 0 is set in $AA_BRAKE_STATE[X] (in the next IPO cycle).				
Return value	The highest deceleration priority in positive direction is shown in bits 0 to 3: 0x0: No pending deceleration request 0x1: Priority 1 covers all positioning actions (G0, POS, SPOS) 0x2: Priority 2 covers DYNNORM and all priority 1 motions 0x3: Priority 3 covers DYNPOS and all priority 1 to 2 motions 0x4: Priority 4 covers DYNROUGH and all priority 1 to 3 motions 0x5: Priority 5 covers DYNSEMIFIN and all priority 1 to 4 motions 0x6: Priority 6 covers all motions (including DYNFINISH). The request could also have been triggered by a CP SW limit stop. 0x7: Priority 7 covers all motions. The request was triggered by the VDI interface signal DB31,..DBX4.3 "Feed stop / Spindle stop". Deceleration always takes place, irrespective of the direction of motion. 0xD: Priority 13 covers all motions. Axial deceleration takes place with an emergency stop deceleration ramp. The highest deceleration priority in negative direction is shown in bits 16 to 19: 0x0 to 0xD: Same meaning as bits 0 to 3 All other bits are reserved and not set. If the value of the variable is shown in hexadecimal format, the fifth character from the right shows the deceleration priority in negative direction while the first number from the right shows the deceleration priority in positive direction.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0x70007
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaBrakeCondM (N, SEMA)				
Conditions for interpolator stop in the MCS Shows the pending braking requests (conditions) for the interpolator stop of the axis / spindle. A braking request consists of a collision direction relating to a coordinate axis in the MCS and a braking priority relating to the machining step.				
Return value	The highest braking priorities in the positive direction are indicated in bits 0 to 3: 0x0: No pending braking request 0x1: Priority 1 covers all positioning actions (G0, POS, SPOS) 0x2: Priority 2 covers DYNNORM and all priority 1 motions 0x3: Priority 3 covers DYNPOS and all priority 1 to 2 motions 0x4: Priority 4 covers DYNROUGH and all priority 1 to 3 motions 0x5: Priority 5 covers DYNSEMIFIN and all priority 1 to 4 motions 0x6: Priority 6 covers all motions (including DYNFINISH). The request could also have been triggered by a CP SW limit stop. 0x7: Priority 7 covers all motions. The request was triggered by the VDI interface signal DB31,..DBX4.3 "Feed stop / Spindle stop". Braking always takes place, irrespective of the direction of motion. 0xD: Priority 13 covers all motions. Axial braking with an emergency stop braking ramp. The highest braking priority in the negative direction is indicated in bits 16 to 19: 0x0 to 0xD: Same meaning as bits 0 to 3 All other bits are reserved and not set. If the value of the variable is shown in hexadecimal format, the fifth character from the right shows the braking priority in the negative direction and the first character from the right shows it in the positive direction.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	0x70007
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaBrakeState (N, SEMA)				
Current braking status Returns for the axis / spindle whether braking has been initiated on account of the request of aaBrakeCondB or a CP SW limit stop or a VDI interface signal DB31,..DBX4.3 "Feed stop / Spindle stop".				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UDWORD	0	0	1
Addressing				
Line index:			Max. line index:	

aaBrakeState (N, SEMA)				
			$$maxnumGlobMachAxes	
			Read access	

aaChanNo (N, SEMA)				
Axis in the channel The variable supplies the number of the channel in which the axis is currently being interpolated. With value 0, the axis could not be assigned to any channel.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaCoupAct (N, SEMA)				
Current coupling state of the slave spindle				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaCoupOffs (N, SEMA)				
Pos. offset of the synchr. spindle desired value Position offset of the synchronous spindle desired value				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaCurr (N, SEMA)				
Actual value of the axis/spindle current in A Actual value of the axis/spindle current in A (Only available for PROFIdrive drives)				
Units and value ranges				
Physical unit:	Data type:			

aaCurr (N, SEMA)				
A	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDepAxO (N, SEMA)				
Dependence on other axes Dependency on other axes. Returns an axis code for the defined axis AX containing all the machine axes that have a mechanical dependency on the defined axis.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDtbb (N, SEMA)				
Axis distance from beginning of the block in BCS Axis-specific distance from the beginning of the block in the BCS for positioning and synchronous axes used in synchronous actions (note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDtbreb (N, SEMA)				
Total decelaration distance, BCS The estimated total distance until the end of deceleration is reached, BCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDtbrebCmd (N, SEMA)				
Command share of decel. dist. of axis ax in BCS Command share of the overall deceleration distance of axis ax in the BCS. The value is the estimated deceleration distance of the axis up to standstill.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDtbrebCorr (N, SEMA)				
Offset section of the deceleration distance, BCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDtbrebDep (N, SEMA)				
Dependent section of the decelaration dist., BCS Dependent section of the decelaration distance, BCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDtbrem (N, SEMA)				
Total decelaration distance, MCS The estimated total distance until the end of deceleration is reached, MCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDtbremCmd (N, SEMA)				
Spec. section of the decelaration distance, MCS Specified section of the decelaration distance, MCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDtbremCorr (N, SEMA)				
Offset section of the deceleration distance, MCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDtbremDep (N, SEMA)				
Dependent section of the decelaration dist., MCS Dependent section of the decelaration distance, MCS				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDteb (N, SEMA)				
Axis distance to the end of the block in the BCS Axis-specific distance to the end of the block in the BCS for positioning and synchronous axes used in synchronous actions (note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

aaDteb (N, SEMA)				
			$$maxnumGlobMachAxes	
			Read access	

aaDtepb (N, SEMA)				
Axis dist-to-go of infeed during oscill. in BCS Axis-specific distance-to-go of infeed during oscillation in the BCS (note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaEnc1Active (N, SEMA)				
First measuring system is active				
Return value	0: Measuring system is not active 1: Measuring system is active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaEnc2Active (N, SEMA)				
Second measuring system is active				
Return value	0: Measuring system is not active 1: Measuring system is active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaEncActive (N, SEMA)				
Measuring system is active				
Return value	0: Measuring system is not active 1: Measuring system is active			

aaEncActive (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaEsrEnable (N, SEMA)				
ESR axis enable (Axial) enabling of reactions of "Extended Stop and Retract" function. The selected axial ESR reaction must be parameterized in MD $MA_ESR_REACTION. beforehand. The corresponding Stop or Retract reactions can be activated via $AN_ESR_TRIGGER (or for individual drives in the event of communications failure/ DC-link undervoltage), generator-mode operation is automatically activated in response to undervoltage conditions.				
Return value	0: FALSE 1: TRUE			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaEsrStat (N, SEMA)				
ESR axis status (Axial) status checkback signals of "Extended Stop and Retract" function, which can be applied as input signals for the gating logic of the ESR (synchronous actions).				
Return value	The data is bit-coded. Individual states can therefore be masked or evaluated separately if necessary: Bit0 = 1: Generator mode is activated Bit1 = 1: Retract operation is activated Bit2 = 1: Stop operation is activated Bit3 = 1: Risk of undervoltage (DC-link voltage monitoring, voltage has dropped below warning threshold) Bit4 = 1: Speed has dropped below minimum generator mode threshold (i.e. no more regenerative rotation energy is available).			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		

aaEsrStat (N, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaEsrTrigger (N, SEMA)				
Activation of ESR Activation of "NC-controlled ESR" for PLC-controlled axis				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaFixPointSelected (N, SEMA)				
Selected fixed point Selected fixed point: Number of the fixed point that is to be approached				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaIbnCorr (N, SEMA)				
BZS setpoint value (N) incl. override components Current BZS setpoint value of an axis including override components				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaIenCorr (N, SEMA)				
SZS setpoint value (N) incl. override components Current SZS setpoint value of an axis including override components				
Units and value ranges				
Physical unit:	Data type:	Initial value		

aaIenCorr (N, SEMA)				
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaInSync (N, SEMA)				
Synchronization status of the following axis Synchronization status of the following axis with master value coupling and ELG				
Return value	0: Synchronization is not running 1: Synchronization is running, i.e. following axis is being synchronized			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaInposStat (N, SEMA)				
Status for the programmed position				
Return value	0: No status available (axis/spindle is outside of the programmed position) 1: Travel motion pending 2: Position setpoint reached 3: Position reached with 'exact stop coarse' 4: Position reached with 'exact stop fine'			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	4
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaJerkCount (N, SEMA)				
Total traverse processes of an axis with jerk				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaJerkTime (N, SEMA)				
Total traverse time of an axis with jerk				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
s, user defined	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaJerkTotal (N, SEMA)				
Overall total jerk of an axis				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaLeadP (N, SEMA)				
Actual lead value position				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaLeadPTurn (N, SEMA)				
Current master value Current master value - position component lost as a result of modulo reduction				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaLeadSp (N, SEMA)				
Simulated lead value - position				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaLeadSv (N, SEMA)				
Simulated leading value velocity				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaLeadV (N, SEMA)				
Actual lead value velocity Actual lead value - velocity				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaLoad (N, SEMA)				
Drive load in % Drive load in % (only available for PROFIdrive drives)				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaMaslState (N, SEMA)				
Current status of the master-slave coupling Each slave axis currently coupled via master-slave delivers the machine axis number of the corresponding master axis. Zero is displayed as default for inactive coupling. A master axis also shows default value zero.				
Return value	0: No coupling for this axis configured, or axis is master axis, or no coupling active >0: Machine axis number of the master axis with which the slave axis is currently coupled			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$numGlobMachAxes
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaMeaAct (N, SEMA)				
Axial measuring active				
Return value	0: Measuring system is not active 1: Measuring system is active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaMm1 (N, SEMA)				
Access to meas. result 1 in MCS Access to measurement result of trigger event in the MCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access and write access	

aaMm2 (N, SEMA)				
Access to meas. result 2 in MCS Access to measurement result of trigger event in the MCS				
Units and value ranges				
Physical unit:	Data type:			

aaMm2 (N, SEMA)				
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access and write access	

aaMm3 (N, SEMA)				
Access to meas. result 3 in MCS Access to measurement result of trigger event in the MCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access and write access	

aaMm4 (N, SEMA)				
Access to meas. result 4 in MCS Access to measurement result of trigger event in the MCS				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access and write access	

aaMm (N, SEMA)				
Latched probe position in the MCS Latched probe position in the machine coordinate system				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access and write access	

aaOff (N, SEMA)				
Position offset from synchronous actions Superimposed position offset from synchronous actions				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaOffLimit (N, SEMA)				
Limit for axial correction $AA_OFF reached Limit for axial correction $AA_OFF reached (Note: for SYNACT only)				
Return value	0: Limit value not reached 1: Limit value reached in positive axis direction 11: Limit value reached in negative axis direction			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaOffVal (N, SEMA)				
Total value of overlaid motion Integrated value of overlaid motion for an axis. The negative value of this variable can be used to cancel an overlaid motion. e.g. $AA_OFF[axis] = -$AA_OFF_VAL[axis]				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaOnFixPoint (N, SEMA)				
Current fixed point Current fixed point, number of the fixed point at which the axis stands				

aaOnFixPoint (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaOscillBreakPos1 (N, SEMA)				
Oscillation interrupt position 1				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaOscillBreakPos2 (N, SEMA)				
Oscillation interrupt position 2				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaOscillReversePos1 (N, SEMA)				
Current reverse position 1 for oscill. in the BCS Current reverse position 1 for oscillation in the BCS. For synchronous actions the value of the setting data $SA_OSCILL_REVERSE_POS1 is evaluated online; (note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaOscillReversePos2 (N, SEMA)				
Current reverse position 2 for oscill. in the BCS Current reverse position 2 for oscillation in the BCS; For synchronous actions the value of the setting data $SA_OSCILL_REVERSE_POS1 is evaluated online; (note: SYNACT only)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaOvr (N, SEMA)				
Axial override for synchronous actions				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaPlcOvr (N, SEMA)				
PLC override for motion-synchronous actions Axial override specified by PLC for motion-synchronous actions				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	100	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaPolfa (N, SEMA)				
Retraction position of the single axis The programmed retraction position of the single axis				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

aaPolfa (N, SEMA)				
			$$maxnumGlobMachAxes	
			Read access	

aaPolfaValid (N, SEMA)				
Retraction of the single axis is programmed States whether the retraction of the single axis is programmed				
Return value	0: No retraction programmed for the single axis 1: Retraction programmed as position 2: Retraction programmed as distance			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaPower (N, SEMA)				
Drive power in W (611D only) Drive power in W (only available for PROFIdrive drives)				
Units and value ranges				
Physical unit:	Data type:			
W	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaProgIndexAxPosNo (N, SEMA)				
Programmed indexing position				
Return value	0: No indexing axis, therefore no indexing position available >0: Number of the programmed indexing position			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaRef (N, SEMA)				
Axis is referenced				
Return value	0: Axis is not referenced 1: Axis is referenced			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaReposDelay (N, SEMA)				
REPOS suppression active				
Return value	0: REPOS suppression is currently not active for this axis 1: REPOS suppression is currently active for this axis			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaScPar (N, SEMA)				
Current setpoint parameter set				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaSnglAxStat (N, SEMA)				
Single axis status Display status of a PLC-controlled axis				
Return value	0: Not a single axis 1: Reset 2: Ended 3: Interrupted 4: Active 5: Alarm			

aaSnglAxStat (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaSoftendn (N, SEMA)				
Software end position, negative direction				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaSoftendp (N, SEMA)				
Software end position, positive direction				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaStat (N, SEMA)				
Axis status Axis state 0: no axis state available 1: travel command is active 2: axis has reached the IPO end. only for channel axes 3: axis in position (exact stop coarse) for all axes 4: axis in position (exact stop fine) for all axes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaSync (N, SEMA)				
Status slave axis with lead value coupling Coupling status of the following axis with master value coupling The actual-value synchronism difference is analyzed to determine the coupling status. See also $VA_SYNCDIFF				
Return value	0: No synchronism 1: Synchronism coarse 2: Synchronism fine 3: Synchronism coarse and fine			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaSyncDiff (N, SEMA)				
Setpoint synchronism difference				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaSyncDiffStat (N, SEMA)				
Status of the synchronism difference Status of the setpoint synchronism difference				
Return value	-4: No valid value in $$aaSyncDiff, coupled motion from part program -3: Reserved -2: Reserved -1: No valid value in $$aaSyncDiff 0: No valid value in $$aaSyncDiff, coupling not active 1: Valid value in $$aaSyncDiff			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-4	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaTorque (N, SEMA)				
Desired torque value in Nm Drive torque setpoint in Nm (only available for PROFIdrive drives)				
Units and value ranges				
Physical unit:	Data type:			
Nm	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaTotalOvr (N, SEMA)				
Total axial override f. motion-synchronous actions The total axial override for motion-synchronous actions				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DOUBLE	100	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaTravelCount (N, SEMA)				
Total traverse processes of an axis				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaTravelCountHS (N, SEMA)				
Total traverse processes of an axis at high speed				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaTravelDist (N, SEMA)				
Total travel path of an axis Total travel path of an axis in mm or degrees				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaTravelDistHS (N, SEMA)				
Total travel path of an axis at high speed Total travel path of an axis at high speed in mm or degrees				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaTravelTime (N, SEMA)				
Total traverse time of an axis Total traverse time of an axis in seconds				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
s, user defined	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaTravelTimeHS (N, SEMA)				
Total traverse time of an axis at high speed Total traverse time of an axis at high speed in seconds				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
s, user defined	TYPE_DOUBLE		0	
Addressing				
Line index:			Max. line index:	

aaTravelTimeHS (N, SEMA)				
			$$maxnumGlobMachAxes	
			Read access	

aaTyp (N, SEMA)				
Axis type according to $AA_TYP Axis type				
Return value	0: axis in other channel 1: channel axis of same channel 2: neutral axis 3: PLC axis 4: reciprocating axis 5: neutral axis, currently traversing in JOG 6: slave axis coupled via master value 7: coupled motion slave axis 8: command axis 9: compile cycle axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaType (N, SEMA)				
Cross-channel axis type Cross-channel axis type				
Return value	0: Axis type cannot be determined 1: NC program axis 2: Neutral axis 3: PLC axis 4: Reciprocating axis 5: Neutral axis that is currently executing a JOG or homing motion 6: Following axis coupled to the master value 7: Coupled motion of the following axis, activated in a synchronized action 8: Command axis 9: Compile Cycle axis 10: Coupled slave axis (master-slave function.) 11: Program axis that is currently executing a JOG or homing motion			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	11

aaType (N, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaVactB (N, SEMA)				
Axis veloc. in basic coordinates Axis velocity in basic coordinate system				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaVactM (N, SEMA)				
Axis veloc. in machine coordinates Axis velocity in machine coordinate system				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaVc (N, SEMA)				
Add. correction value for path feed or axial feed Additive correction value for path feed or axial feed				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acRpValid (N, SEMA)				
Reapproach position valid Reapproach position valid				
Return value	0: Reapproach position not valid 1: Reapproach position valid			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

ackSafeMeasPos (N, SEMA)				
Confirmation of SI actual position				
Return value	0 = not confirmed 0x00AC = confirmed			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access and write access	

actCouppPosOffset (N, SEMA)				S3
Offset to leading axis/spindle, actual value Position offset of an axis to a leading axis / leading spindle (actual value)				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
mm, inch, degree, user defined	TYPE_DOUBLE		0	360
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

actFeedRate (N, SEMA)				S5
Actual value of axis-specific feedrate Actual value of axis-specific feedrate, if the axis is a positioning axis.				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			

actFeedRate (N, SEMA)				S5
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

actIndexAxPosNo (N, SEMA)				
Current indexing position number Current indexing position number 0 = no indexing position >0 = indexing position number				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

actSpeedRel (N, SEMA)				
Actual value of rotary speed Actual value of rotary speed (referring to the maximum speed in %), for linear drives actual value of the velocity.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

actValResol (N, SEMA)				
Actual value of the resolution Actual value resolution. The physical unit is defined in measUnit (in this module)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

amSetupState (N, SEMA)				
State var. PI Service auto. set-up of asyn. motor State variable of the PI Service Automatic set-up of an asynchronous motor				
Return value	0 = inactive 1 = wait for PLC enable 2 = wait for key NC-start 3 = active 4 = stopped by Servo + fine code in the upper byte 5 = stopped by 611D + fine code in the upper byte 6 = stopped by NCK + fine code in the upper byte			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0xff06
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

axComp (N, SEMA)				
Sum of compensation values Sum of axis-specific compensation values (CEC Cross Error compensation and temperature compensation). The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

axisActiveInChan (N, SEMA)				
Axis active Flag indicating whether axis is active in this channel				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

axisFeedRateUnit (N, SEMA)				
Unit of axial feedrate				
Return value	0 = mm/min 1 = inch/min 2 = degree/min			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

chanAxisNoGap (N, SEMA)				
Axis exists Display of existing axis, i.e. no axis gap in channel.				
Return value	0: Axis does not exist 1: Axis does exist			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

chanNoAxisIsActive (N, SEMA)				
Channel number of axis Channel number in which the channel axis is currently active				
Return value	0 = axis is not assigned to any channel 1 to $$maxnumChannels (Area.:N / Module:Y) = channel number			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdContrPos (N, SEMA)				
Desired value of position after fine interpolation				
Units and value ranges				
Physical unit:	Data type:			

cmdContrPos (N, SEMA)				
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdCouppPosOffset (N, SEMA)				S3
Position offset referring to leading axis/spindle Position offset of an axis referring to the leading axis / leading spindle (desired value)				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
mm, inch, degree, user defined	TYPE_DOUBLE		0	360
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdFeedRate (N, SEMA)				
Desired value of axis-specific feedrate Desired value of axis-specific feedrate for a positioning axis.				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdSpeedRel (N, SEMA)				
Desired value of rotary speed. Speed setpoint (as % of the maximum speed), velocity setpoint in the case of linear motors.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

contrConfirmActive (N, SEMA)				
Controller enable				
Return value	0 = no controller enable 1 = controller enable			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

contrMode (N, SEMA)				
Controller mode servo Identifier for controller mode servo				
Return value	0 = position control 1 = speed control 2 = stop 3 = park 4 = follow-up (set the mode through VDI interface and partly through part program)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

displayAxis (N, SEMA)				
Machine axis identifier Identifier indicating whether axis is displayed by HMI as a machine axis.				
Return value	0 = Do not display at all 0xFFFF = Always display everything bit 0 = Display in actual-value window bit 1 = Display in reference point window bit 2 = Display in Preset / Basic offset / Scratching bit 3 = Display in handwheel selection			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0xFFFF	0	0xFFFF
Addressing				
Line index:			Max. line index:	

displayAxis (N, SEMA)				
			$$maxnumGlobMachAxes	
			Read access	

distPerDriveRevol (N, SEMA)				
Distance per revolution of the drive Rotary drive: Load-side path corresponding to one revolution of the drive. Is returned in the unit of the internal computational resolution INT_INCR_PER_MM (for linear axes) or INT_INCR_PER_DEG (for rotary axes / spindles) taking into account gear ratios etc. In the case of linear axes, the pitch of the ball screw is also included in the calculation. In the case of linear motors, a fixed value of "1mm" is used for the ball screw pitch instead of the non-existent ball screw.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

drfVal (N, SEMA)				
DRF value				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

drive2ndTorqueLimit (N, SEMA)				
2nd torque limit 2nd torque limit. With linear motors: 2nd force limit				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveActMotorSwitch (N, SEMA)				
Actual motor wiring Actual motor wiring (star/delta)				
Return value	0 = star 1 = delta			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveActParamSet (N, SEMA)				
Number of the actual drive parameter set				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		1	8
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveClass1Alarm (N, SEMA)				
Message ZK1 drive alarm				
Return value	0 = no alarm set 1= alarm set (fatal error occured)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveContrMode (N, SEMA)				
Control mode of drive				
Return value	0 = current control 1 = speed control			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

driveContrMode (N, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveCoolerTempWarn (N, SEMA)				
Heatsink temperature monitoring				
Return value	0 = temperature OK 1 = overtemperature			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveDesMotorSwitch (N, SEMA)				
Motor wiring selection (star/delta)				
Return value	0 = star 1 = delta			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveDesParamSet (N, SEMA)				
Desired parameter set of the drive				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		1	8
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveFastStop (N, SEMA)				
Ramp-function generator rapid stop				
Return value	0 = not stopped 1 = stopped			

driveFastStop (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveFreqMode (N, SEMA)				
I/F mode				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveImpulseEnabled (N, SEMA)				
Enable inverter impulse Enable inverter impulse (checkback signal to impulseEnable)				
Return value	0 = not enabled 1 = enabled			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveIndex (N, SEMA)				
Drive assignment (driveIndex) Drive assignment (logical drive number)				
Return value	0 = drive does not exist 1 to 15 = logical drive number			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	15
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveIntegDisable (N, SEMA)				
Integrator disable				
Return value	0 = not disabled 1 = disabled			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveLinkVoltageOk (N, SEMA)				
State of the DC link voltage				
Return value	0 = OK 1 = not OK			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveMotorTempWarn (N, SEMA)				
Motor temperature warning				
Return value	0 = temperature OK 1 = overtemperature			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveNumCrcErrors (N, SEMA)				
CRC errors on the drive bus CRC errors on the drive bus (Transmission errors when writing data to the drive; values may range up to FFFFH)				
Return value	0 = no error			
Units and value ranges				
Physical unit:	Data type:			

driveNumCrcErrors (N, SEMA)				
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveParked (N, SEMA)				
Parking axis				
Return value	0 = no parking axis 1 = parking axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

drivePowerOn (N, SEMA)				
Drive switched on				
Return value	0 = drive not switched on 1 = drive switched on			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveProgMessages (N, SEMA)				
Configurable messages Configurable messages (via machine data)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveReady (N, SEMA)				
Drive ready				
Return value	0 = drive not ready 1 = drive ready			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveRunLevel (N, SEMA)				
Current state reached during the boot process Current state reached during the boot process (range: coarse state (0 to 5) * 100 + fine state (up to 22)				
Return value	Booting the firmware ---> 0 XX entering the configuration ---> 1XX hardware-init, communication-init loading, converting data ---> 2XX changing bus addressing ---> 3XX preparing synchronization ---> 4XX activating interrupt ---> 519 XX ==> fine state			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveSetupMode (N, SEMA)				
Set-up mode				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

driveSetupMode (N, SEMA)				
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveSpeedSmoothing (N, SEMA)				
Smoothing the desired value of the rotary speed Smoothing the desired value of the rotary speed, for linear drives: smoothing the desired value of the velocity				
Return value	0 = no smoothing 1 = smoothing			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

effComp1 (N, SEMA)				
Sum of the compensation values for encoder 1 Sum of the compensation values for encoder 1. The value results from: Temperature compensation, backlash compensation, quadrant error compensation, beam sag compensation, leadscrew error compensation. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

effComp2 (N, SEMA)				
Sum of the compensation values for encoder 2 Sum of the compensation values for encoder 2. The value results from: Temperature compensation, backlash compensation, quadrant error compensation, beam sag compensation, leadscrew error compensation. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

encChoice (N, SEMA)				
Active encoder				
Return value	0 = does not exist 1 = encoder 1 2 = encoder 2			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

fctGenState (N, SEMA)				
State of the function generator				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

feedRateOvr (N, SEMA)				
Feedrate override Feedrate override (only if axis is a positioning axis)				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

focStat (N, SEMA)				
Status of ForceControl function Current status of "Travel with limited torque" function				
Return value	0-2 0: FOC not active 1: FOC modal active (programming of FOCON[]) 2: FOC non-modal active (programming of FOC[])			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

focStat (N, SEMA)				
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

fxsInfo (N, SEMA)				
Additional info on travel to fixed stop Additional information on travel to fixed stop if $VA_FXS[]=2, or OPI variable $$/N/SEMA/fxsStat=2.				
Return value	0 No additional information available 1 No approach motion programmed 2 Programmed end position reached, movement ended 3 Abort by NC RESET (Reset key) 4 Fixed stop window exited 5 Torque reduction was rejected by drive 6 PLC has canceled enable signals			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

fxsStat (N, SEMA)				
State after travelling to fixed stop				
Return value	0 = normal control 1 = fixed stop reached 2 = selection failed			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

handwheelAss (N, SEMA)				
Number of assigned handwheel Number of handwheel assigned to axis				
Return value	0 = no handwheel assigned 1 to 3 = handwheel number			

handwheelAss (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	3
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

impulseEnable (N, SEMA)				
Impulse enable for drive				
Return value	0 = not enabled 1 = enabled			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

index (N, SEMA)				
Absolute axis index Absolute axis index referred to MD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

isDriveUsed (N, SEMA)				
Machine axis has control over the drive One or more machine axes are assigned to each drive. The drive can only be controlled at any one time by one of these machine axes. The machine manufacturer makes the selection. The status of the drive control changes dynamically.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	

isDriveUsed (N, SEMA)				
			$$maxnumGlobMachAxes	
			Read access	

kVFactor (N, SEMA)				
position control gain factor				
Units and value ranges				
Physical unit:	Data type:			
16.667 1/s	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

lag (N, SEMA)				
Following error Following error = desired value of position after fine interpolation - actual value of position. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

logDriveNo (N, SEMA)				
Drive assignment (logDriveNo) Drive assignment (logical drive number)				
Return value	0 = not available 1 to 15 = drive number			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	15
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

measFctState (N, SEMA)				
State of the probing function				
Units and value ranges				
Physical unit:	Data type:			

measFctState (N, SEMA)				
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

measPos1 (N, SEMA)				
Actual position value for measuring system 1. Actual value of position for encoder 1. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

measPos2 (N, SEMA)				
Actual position value for measuring system 2. Actual value of position for encoder 2. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

measPosDev (N, SEMA)				
Actual position difference between 2 encoders Actual position difference between the two encoders. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

measUnit (N, SEMA)				
Unit for service values of the drives				
Return value	0 = mm 1 = inch 2 = grd			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

paramSetNo (N, SEMA)				
Number of parameter set				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		1	8
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

preContrFactTorque (N, SEMA)				
Feed forward control factor torque				
Units and value ranges				
Physical unit:	Data type:			
Nm	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

preContrFactVel (N, SEMA)				
Feed forward control factor velocity				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

preContrMode (N, SEMA)				
Feed forward control mode				
Return value	0 = inactive 1 = velocity feed forward 2 = torque feed forward			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

progIndexAxPosNo (N, SEMA)				
Indexing number Programmed indexing position number 0 = no indexing position >0 = indexing position number				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

qecLrnIsOn (N, SEMA)				
Quadrant error compensation learning active				
Return value	0 = inactive 1 = Neuronal-QEC learning active 2 = Standard-QEC active 3 = Standard-QEC with adaptation of correction value active 4 = Neuronal-QEC active 5 = Neuronal-QEC with adaptation of measuring time active 6 = Neuronal-QEC with adaptation of decay time of correction value active 7 = Neuronal-QEC with adaptation of measuring time and decay time of correction value active			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	7
Addressing				
Line index:			Max. line index:	

qecLrnIsOn (N, SEMA)				
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

refPtBusy (N, SEMA)				
Axis is being referenced				
Return value	0 = axis is not being referenced 1 = axis is being referenced			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

refPtCamNo (N, SEMA)				
Reference point cam				
Return value	0 = no cam approached 1 = cam 1 2 = cam 2 3 = cam 3 4 = cam 4			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

refPtPhase (N, SEMA)				
Referencing phases				
Return value	0 = False 1 = Phase 1 2 = Phase 2 3 = Phase 3 4 = Phase 4			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

refPtPhase (N, SEMA)				
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

refPtStatus (N, SEMA)				
Axis must be referenced and is referenced				
Identifier indicating whether an axis requires referencing and is referenced. Note regarding exchange axes: An exchange axis need only ever be referenced in the channel to which it is currently assigned. A referenced exchange axis is thus logged onto the channel in which it is traversing with value "3" (requires referencing and referenced) and in other channels with value "1" (does not require referencing, but referenced).				
Return value	Set bits have the following meanings: Bit0: current measuring system has been referenced Bit1: current measuring system requires referencing (A busy signal affects the state)			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	Achsindex		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

resolvStatus1 (N, SEMA)				
Encoder status for measuring system 1				
Return value	0 = Undefined 1 = Referenced 2 = Activated 3 = Limit frequency exceeded			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

resolvStatus2 (N, SEMA)				
Encoder status for measuring system 2				
Return value	0 = Undefined 1 = Referenced 2 = Activated 3 = Limit frequency exceeded			

resolvStatus2 (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

safeAcceptCheckPhase (N, SEMA)				
Flag for NCK-side acceptance test phase Flag for NCK-side acceptance test phase, the human-machine interface can determine which acceptance test phase is present on the NCK.				
Return value	0: NCK has acceptance test phase inactive = 0 0ACH: NCK has acceptance test phase active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0ACH
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeAcceptTestMode (N, SEMA)				
SI PowerOn alarms can be acknowledged by Reset SI PowerOn alarms can be acknowledged by Reset in acceptance test mode				
Return value	0: Acceptance test mode: SI PowerOn alarms cannot be acknowledged by Reset 0ACH: Acceptance test mode: SI PowerOn alarms can be acknowledged by Reset			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0FFH
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access and write access	

safeAcceptTestPhase (N, SEMA)				
Flag for acceptance test phase				
Return value	0: Acceptance test Wizard not selected, activate NCK-side alarm suppression 0ACH: Dialogs for acceptance test support selected, deactivate NCK-side alarm suppression			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0FFH

safeAcceptTestPhase (N, SEMA)		
Addressing		
Line index:	Max. line index:	
	$$maxnumGlobMachAxes	
	Read access and write access	

safeAcceptTestSE (N, SEMA)				
Flag for NCK-side SE acceptance test Flag for NCK-side SE acceptance test. The human-machine interface starts checking the safe limit positions during the acceptance test				
Return value	0: NCK has SE acceptance test inactive = 0. The single channel SW limit positions are activated. 0ACH: NCK is to activate SE acceptance test. The single channel SW limit positions are de-activated in this way.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0ACH
Addressing				
Line index:	Max. line index:			
	$$maxnumGlobMachAxes			
	Read access			

safeAcceptTestState (N, SEMA)				
Flag for acceptance test status Flag for acceptance test status, the human-machine interface can determine which acceptance test mode is present on the NCK.				
Return value	0: NCK has inactive acceptance test mode 0CH: Acceptance test mode not activated because SI PowerOn alarms already present. The causes of the SI PowerOn alarms must be eliminated first. 0DH: Acceptance test mode not activated, the HMI writes invalid values in $$/N/SEMA/safe-AcceptTestMode to the NCK. 0ACH: NCK has active acceptance test mode			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0FFH
Addressing				
Line index:	Max. line index:			
	$$maxnumGlobMachAxes			
	Read access			

safeActPosDiff (N, SEMA)		
Curr. safety actual value difference NCK - drive Current actual value difference betw. NCK and drive monitoring channels		

safeActPosDiff (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeActVeloDiff (N, SEMA)				
Curr. safety speed difference NCK - drive Current speed difference between NCK and drive monitoring channels				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeActVeloLimit (N, SEMA)				
Safe limit of actual speed				
Return value	-1 => no actual speed limit active >= 0 => limit of actual speed is active			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE		-1	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeDesVeloLimit (N, SEMA)				
Safe limit of desired speed				
Return value	-1 => no desired speed limit active >= 0 => desired speed limit is active			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE		-1	
Addressing				
Line index:			Max. line index:	

safeDesVeloLimit (N, SEMA)				
			$$maxnumGlobMachAxes	
			Read access	

safeFctEnable (N, SEMA)				
Safe operation active Safe operation active (Safety Integrated / SPL)				
Return value	0 = inactive >0 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

safeInputSig2 (N, SEMA)				
Safe input signals part 2				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	0xffff
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeInputSig (N, SEMA)				
Safe input signals of the axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

safeInputSigDrive2 (N, SEMA)				
Safe input signals of the drive part 2				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	0xffff
Addressing				
Line index:			Max. line index:	

safeInputSigDrive2 (N, SEMA)				
			$$maxnumGlobMachAxes	
			Read access	

safeInputSigDrive (N, SEMA)				
Safe input signals of the drive				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

safeMaxVeloDiff (N, SEMA)				
Max. safety speed difference NCK - drive Maximum speed difference between NCK and drive monitoring channels since last NCK Reset				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeMeasPos (N, SEMA)				
Safe actual position of axis Safe actual position of the axis. The physical unit is defined in the variable measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

safeMeasPosDrive (N, SEMA)				
Safe actual position of the drive Safe actual position of drive. The physical unit is defined in measUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			

safeMeasPosDrive (N, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

safeOutputSig2 (N, SEMA)				
Safe output signals part 2				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	0xffff
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeOutputSig (N, SEMA)				
Safe output signals of the axis				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

safeOutputSigCam (N, SEMA)				
NCK SI cam signals Results of the NCK safe cam evaluation				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	3FFFFFFF
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeOutputSigCamDrive (N, SEMA)				
Drive SI cam signals Results of the drive safe cam evaluation				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	3FFFFFFF

safeOutputSigCamDrive (N, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeOutputSigDrive2 (N, SEMA)				
Safe output signals of the drive part 2				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	0xffff
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeOutputSigDrive (N, SEMA)				
Safe output signals of the drive				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

safeActiveCamTrack (N, SEMA)				
Status Safe cam track Status Safe cam track (active/inactive)				
Return value	Bit 0 = 1/0: Safe cam track 1 active/inactive Bit 1 = 1/0: Safe cam track 2 active/inactive Bit 2 = 1/0: Safe cam track 3 active/inactive Bit 3 = 1/0: Safe cam track 4 active/inactive			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	0xF
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safePosCtrlActive (N, SEMA)				
Axis monitors absolute position				
Return value	0 = Axis does not monitor absolute position (no SE/SN) 1 = Axis monitors absolute position			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeStopOtherAxis (N, SEMA)				
Stop on another axis				
Return value	0: No stop on another axis 1: Stop on another axis			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

spec (N, SEMA)				
Axis specification				
Return value	0 = path axis 1 = positioning axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

stateContrActive (N, SEMA)				
State controller State controller (not available)				
Return value	1 = TRUE 0 = FALSE			
Units and value ranges				
Physical unit:	Data type:			

stateContrActive (N, SEMA)				
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

subSpec (N, SEMA)				T1
Subspecification				
Return value	0 = normal axis 1 = indexing axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

torqLimit (N, SEMA)				
Torque limitation value Torque limitation value (referring to the nominal value of the drive). For linear motors: force limitation value.				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

traceState1 (N, SEMA)				
State of trace channel 1				
Return value	0 = idle state 1 = recording started 2 = trigger reached 3 = recording ended 4 = recording aborted			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

traceState2 (N, SEMA)				
State of trace channel 2				
Return value	0 = idle state 1 = recording started 2 = trigger reached 3 = recording ended 4 = recording aborted			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

traceState3 (N, SEMA)				
State of trace channel 3				
Return value	0 = idle state 1 = recording started 2 = trigger reached 3 = recording ended 4 = recording aborted			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

traceState4 (N, SEMA)				
State of trace channel 4				
Return value	0 = idle state 1 = recording started 2 = trigger reached 3 = recording ended 4 = recording aborted			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

traceState4 (N, SEMA)				
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

trackErrContr (N, SEMA)				
Position controller difference Position controller difference (actual value / desired value of position)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

trackErrDiff (N, SEMA)				
Contour deviation Contour deviation (difference actual value of position and calculated dynamical model)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

type (N, SEMA)				
Axis type Axis type In the case of a link axis, the initial setting of the machine data is returned according to $$axisType. The difference between spindle and rotary axis cannot be determined because there is no possibility of accessing the other NCUs. Thus, in this case, there is no value 2 for spindle.				
Return value	0 = linear axis 1 = rotary axis 2 = spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

type (N, SEMA)				
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

vaAbsoluteEnc1ErrCnt (N, SEMA)				
1st meas. system: Error counter for abs. encoder Enc 1: Error counter for absolute encoder				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaAbsoluteEnc1State (N, SEMA)				
1st meas. system: Status of abs. encoder interface Enc1: Status of absolute encoder interface				
Return value	Bit0: Interface is active Bit1: Error during parity check Bit2: Error bit alarm Bit3: Error bit CRC error Bit4: Start bit missing with EnDat transfer			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaAbsoluteEnc2ErrCnt (N, SEMA)				
2nd meas. system: Error counter for abs. encoder Enc 2: Error counter for absolute encoder				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaAbsoluteEnc2State (N, SEMA)				
2nd meas. system: Status of abs. encoder interface Enc2: Status of absolute encoder interface				
Return value	Bit0: Interface is active Bit1: Error during parity check Bit2: Error bit alarm Bit3: Error bit CRC error Bit4: Start bit missing with EnDat transfer			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCcCompValTotal (N, SEMA)				
Axial OA total compensation value Axial OA total compensation value of the compile cycles				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCecCompVal (N, SEMA)				
Axial sag compensation value				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaEnc1CompVal (N, SEMA)				
Leadscrew error compensation (LEC) value encoder 1				
Units and value ranges				
Physical unit:	Data type:	Initial value		

vaEnc1CompVal (N, SEMA)				
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaEnc1ZeroMonErrCnt (N, SEMA)				
1st meas. system: Error counter for zero mark mon. Enc 1: Error counter for zero mark monitoring				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaEnc2CompVal (N, SEMA)				
Leadscrew error compensation (LEC) value encoder 2				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaEnc2ZeroMonErrCnt (N, SEMA)				
2nd meas. system: Error counter for zero mark mon. Enc 2: Error counter for zero mark monitoring				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_DWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCurr (N, SEMA)				
Drive actual current value				

vaCurr (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaDistTorque (N, SEMA)				
Disturbing torque/max. torque (motor end) Disturbing torque/max. torque (motor end, York)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
%	TYPE_DOUBLE	0	-100	100
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaDpe (N, SEMA)				
Status of power enable Status of power enable of a machine axis				
Return value	0 - 1			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaFoc (N, SEMA)				
Actual status of "ForceControl"				
Return value	0: ForceControl not active 1: Modal ForceControl active 2: Non-modal ForceControl active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	2
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaFxs (N, SEMA)				
Actual status of "Travel to fixed stop"				
Return value	0: Axis not at fixed stop 1: Successful travel to fixed stop 2: Unsuccessful travel to fixed stop 3: Travel to fixed stop selection active 4: Fixed stop has been detected 5: Travel to fixed stop deselection active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	5
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

valm1 (N, SEMA)				
Alternative name:	aalm1			
Actual value encoder 1 Actual value in the machine coordinate system (measured encoder 1)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

valm2 (N, SEMA)				
Alternative name:	aalm2			
Actual value encoder 2 Actual value in the machine coordinate system (measured encoder 2)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

valm (N, SEMA)				
Alternative name:	aalm			
Current MCS actual value of an axis The variable determines the encoder actual value (measured by active measuring system) in the machine coordinate system (MCS). All actual value compensations are corrected (leadscrew error compensation, backlash compensation, quadrant error compensation). For rotary axes/spindles, the returned value is independent of the modulo setting, a transformation is only performed for specific actions. When a spindle or axis disable is active, this variable returns the current setpoint. If, in this case, the actual value also has to be returned, BIT3 must be set in MD30455 $MA_MISC_FUNCTION_MASK.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaLagError (N, SEMA)				
Axis following error				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaLoad (N, SEMA)				
Drive utilization in %				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DOUBLE	0	-100	100
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaPosctrlMode (N, SEMA)				
Position controller mode Position controller mode"				
Return value	0: Position control 1: Speed control 2: Holding 3: Parking 4: Tracking			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	4
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaPower (N, SEMA)				
Active drive power				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaPressureA (N, SEMA)				
Pressure on A end of the cylinder Pressure on A end of the cylinder in bar (only for Hydraulic)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaPressureB (N, SEMA)				
Pressure on B end of the cylinder Pressure on B end of the cylinder in bar (only for Hydraulic)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		

vaPressureB (N, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaSce (N, SEMA)				
Status of speed controller enable				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaStopSi (N, SEMA)				
Stop from Safety Integrated				
Return value	-1: No stop 0: Stop A 1: Stop B 2: Stop C 3: Stop D 4: Stop E 5: Stop F 10: Test stop of NC			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaSyncDiff (N, SEMA)				
Actual value synchronism difference Actual value synchronism difference for all types of coupling				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	

vaSyncDiff (N, SEMA)				
			$$maxnumGlobMachAxes	
			Read access	

vaSyncDiffStat (N, SEMA)				
Status of the actual value synchronism difference				
Return value	-4: Reserved -3: No valid value in $VA_SYNCDIFF, tangential control -2: No valid value in $VA_SYNCDIFF, master value coupling and simulated master value -1: No valid value in $VA_SYNCDIFF 0: No valid value in $VA_SYNCDIFF, coupling not active 1: Valid value in $VA_SYNCDIFF			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
mm, inch, degree, user defined	TYPE_DWORD	0	-4	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaTempCompVal (N, SEMA)				
Axial temperature compensation value				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaTorque (N, SEMA)				
Drive torque setpoint				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaTorqueAtLimit (N, SEMA)				
Status information torque limitation Status "effective torque equals specified torque limit"				
Return value	0: Effective torque lower than torque limit 1: Effective torque has reached torque limit			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaVactm (N, SEMA)				
Axis velocity actual value load side in the MCS Axis velocity actual value on the load side in the MCS				
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaValveLift (N, SEMA)				
Actual valve lift Actual valve lift in mm (only for Hydraulic)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaXfaultSi (N, SEMA)				
Stop F through cross-checking error active				
Return value	Bit 0 set: An actual value error has been discovered in the cross-check between NCK and drive Bit 1 set: Some error has been discovered in the cross-check between NCK and drive and the waiting time until Stop B ($MA_SAFE_STOP_SWITCH_TIME_F) is running or has expired			

vaXfaultSi (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

activeSvOverride (N, SEMA)				
SG override factor Currently active SG override factor in the NCK				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	-1	-1	100
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaJogPosSelected (N, SEMA)				
JOG to position is active				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaJogPosAct (N, SEMA)				
Position reached for JOG to position				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaEnc1ZeroMonAccessCnt (N, SEMA)				
Enc1:Update counter zero monitoring accesses Enc1: Update counter				

vaEnc1ZeroMonAccessCnt (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaEnc2ZeroMonAccessCnt (N, SEMA)				
Enc2:Update counter zero monitoring accesses Enc2: Update counter				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaAbsoluteEnc1DeltaInit (N, SEMA)				
Enc 1: Initial difference with absolute encoder Enc1: Initial difference				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaAbsoluteEnc2DeltaInit (N, SEMA)				
Enc 2: Initial difference with absolute encoder Enc2: Initial difference				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaEnc1ZeroMonAct (N, SEMA)				
Enc1: Current internal zero monitoring values Enc1: Zero monitoring values				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaEnc2ZeroMonAct (N, SEMA)				
Enc2: Current internal zero monitoring values Enc2: Zero monitoring values				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaAbsoluteEnc1ZeroMonMax (N, SEMA)				
Enc1:Maximum of $$vaEnc1ZeroMonAct w. abs. encoder Enc1:Maximum of $$vaEnc1ZeroMonAct with absolute encoder				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaAbsoluteEnc2ZeroMonMax (N, SEMA)				
Enc2:Maximum of $$vaEnc2ZeroMonAct w. abs. encoder Enc2:Maximum of $$vaEnc2ZeroMonAct with absolute encoder				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaCoupCorr (N, SEMA)				
Gen. coupling: comp. value of synchronism diff. This variable is used to execute the function "Correct synchronism error". It returns the compensation value for the position offset for the generic coupling with CPFRS = "MCS". The actual values of this spindle are compared with the setpoints for the duration (MD 30455 MISC_FUNCTION_MASK, bit 7) of the activation of the VDI interface signal DB31...,DBX31.6 'Correct synchronism' for the following spindle with coupling active. The difference is the compensation value, which can be read with this variable.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaCoupCorrDist (N, SEMA)				
Generic coupling: path for $$aaCoupCorr Generic coupling: path still to be retracted for $$aaCoupCorr				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaEnc1ZeroMonInit (N, SEMA)				
Enc1:Hardware counter value of basic zero mark Enc1:Hardware counter value of the basic zero mark				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaEnc2ZeroMonInit (N, SEMA)				
Enc2:Hardware counter value of basic zero mark Enc2:Hardware counter value of the basic zero mark				

vaEnc2ZeroMonInit (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeDesChecksum1 (N, SEMA)				
Setpoint hardware checksum Setpoint checksum of the current values of the safety-relevant machine data dependent on hardware parameterization				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access and write access	

aaMaslDef (N, SEMA)				
Current status of master-slave configuration Each slave axis currently coupled via master-slave delivers the machine axis number of the corresponding master axis. Zero is displayed as default if the coupling is not configured. A master axis also shows default value zero.				
Return value	0: No coupling for this axis configured, or axis is master axis, or no coupling active >0: Machine axis number of the master axis with which the slave axis is currently coupled			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$numGlobMachAxes
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

clampStatus (N, SEMA)				
Axis is connected (VDI input signal)				
Return value	Bit 0 = 1: Axis is connected			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	

clampStatus (N, SEMA)		
		$$maxnumGlobMachAxes
		Read access

aaMachax (N, SEMA)				
Assignment of physical axis The NCU and machine axis are output for an axis, representing the physical image of the axis. The machine axis must be assigned to at least one channel on this NCU, otherwise 0 will be returned. Without an NCU link, i.e. if there is only one NCU, only the number of the machine axis will be output. In this case, the NCU ID is equal to zero.				
Return value	The NCU ID is output as from position 10000, e.g. 20005: NCU 2 axis 5.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaIpoNcChanax (N, SEMA)				
NC, channel and channel axis no. of interpolator. If the axis is currently interpolated to this NCU, the channel and channel number which define the interpolator of the axis are output. If the axis is currently interpolated to a different NCU, the NCU identifier of the interpolated NCU and the global axis number of the machine axis are output. This global axis number can then be used to transfer the interpolated channel and the channel axis number to the other NCU, with NCU ID 2, with $AN_IPO_CHANAX[203]. The axis must be assigned to at least one channel on this NCU, otherwise 0 will be returned.				
Return value	The channel is output as from position 100, and the channel axis number is output as from position 1, e.g. 1005 - channel 10 channel axis 5. These values are always lower than 10000. The NCU is output as from position 10000, e.g. 20203: NCU 2 and the global axis number is 203.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaIpoNcChanax (N, SEMA)				
Machine axis to interpolator. If the machine axis is currently interpolated to this NCU, the channel and channel axis number which define the interpolator of the axis are output. If the machine axis is currently interpolated to a different NCU, the NCU identifier of the interpolated NCU and the global axis number of the machine axis are output. This global axis number can then be used to transfer the interpolated channel and the channel axis number to the other NCU, with NCU ID 2, with $AN_IPO_CHANAX[103]. If no machine axis is used, 0 will be returned.				
Return value	The channel is output as from position 100, and the channel axis number as from position 1, e.g. 1005 - channel 10 channel axis 5. These values are always lower than 10000. The NCU is output as from position 10000, e.g. 20103: NCU 2 and the global axis number is 103.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaMotSensorDigi (N, SEMA)				
Digital sensors on the motor This variable determines the states of the digital sensors S4 and S5. The variable is bit-coded, and has the following meanings: Bit0 = 0: Bit1 = 0: Bit2 = 0: Bit3 = 0: Bit4 = 1: Sensor S4 piston end position Bit5 = 1: Sensor S5 angular position of the shaft				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaMotSensorAna (N, SEMA)				
Analog sensor on the motor This variable determines the analog measured value of sensor S1. At a resolution of 1 mV, the analog value 0 - 10 V is mapped by a maximum of +10000 increments.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaMotSensorConf (N, SEMA)				
Configuration of motor sensors The configuration of the motor sensors can be queried with this variable. The variable is bit-coded, and has the following meanings: Bit0 = 1: Sensor system present. Bit1 = 1: Sensor S1 present. Analog measured value for position of the draw-bar. Bit2 = 0: Bit3 = 0: Bit4 = 1: Sensor S4 present. Digital value for the piston end position Bit5 = 1: Sensor S5 present. Digital value for the angular position of the shaft.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaMotClampingState (N, SEMA)				
State of the clamping system Starting from the position of the draw-bar (value of the S1), this variable determines the clamping state. A maximum speed is assigned to each state. These are stored in the drive parameters p5043[0..6]. The following values are possible: 0: Sensor not present 1: Initial state, speed limit 0 rpm 2: Alarm, speed limit 0 rpm 3: Tool released / ejected, speed limit see drive parameter p5043[0] 4: Clamping (by spring force), speed limit see drive parameter p5043[1] 5: Releasing (by compressed air), speed limit see drive parameter p5043[2] 6: Releasing (by compressed air), speed limit see drive parameter p5043[3] 7: Clamped with tool, speed limit see drive parameter p5043[4] 8: Clamped with tool, speed limit see drive parameter p5043[4] 9: Further clamping (by spring force), speed limit see drive parameter p5043[5] 10: Clamped without tool, speed limit see drive parameter p5043[6] 11: Alarm, speed limit 0 rpm				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UDWORD	0	0	
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCpSync2 (N, SEMA)				
Status synchronism(2) of the foll. axis / spindle Second synchronism monitoring of the following axis / spindle				
Return value	0: Monitoring not active Bit 0 = 1: Monitoring 'Synchronism(2) coarse' active Bit 1 = 1: Synchronism(2) coarse available Bit 2 = 1: Monitoring 'Synchronism(2) fine' active Bit 3 = 1: Synchronism(2) fine available			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaPosRes (N, SEMA)				
Axis status "Position restored" Axis status "Position restored". The value TRUE shows that the position of the axis has been restored after the voltage breakdown. ($MA_ENC_REFP_STATE[] = 3). After referencing of the axis, the value goes to FALSE.				
Return value	1 = TRUE: Axis position not restored 0 = FALSE: Axis position restored			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaCollPos (N, SEMA)				
Collision position Position of a machine axis with risk of collision				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
			Read access	

driveDdsPerMds (N, SEMA)				
Number of drive data sets Number of drive data sets assigned to a motor data set. Refer to the SINAMICS S120 Function Manual for more information about drive and motor data sets.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

spindleModePiState (N, SEMA)				
Status of the spindle mode switchover Status of the spindle mode switchover for this machine axis by PI service _N_SPIMOD				
Return value	0 = PI service not selected 10 = PI service active 50 = PI service ended successfully 101 = PI service rejected because axis/spindle is not known in the channel 102 = PI service rejected because axis/spindle is not available in the channel 104 = PI service rejected because axis/spindle is not defined as a spindle. 105 = PI service rejected because axis/spindle is a permanently assigned PLC axis/spindle 106 = PI service rejected because axis/spindle is an active following axis/spindle 107 = PI service rejected because axis/spindle is a transformed spindle/axis 108 = PI service rejected because axis/spindle is not available as a command axis 200 = PI service rejected because of an internal error			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	999
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaLoadSmooth (N, SEMA)				
Smoothed drive load Smoothed drive load in %				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaPowerSmooth (N, SEMA)				
Smoothed drive power Smoothed drive power in W (only available for PROFIdrive drives)				
Units and value ranges				
Physical unit:	Data type:			
W	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

safeAxisType (N, SEMA)				
Safety axis mode Type of axial safety monitoring 0 = No SINUMERIK Safety Integrated active 1 = SINUMERIK Safety Integrated (SPL) active 2 = SINUMERIK Safety Integrated plus (F-PLC) active 2 = SINUMERIK Safety Integrated (F-PLC) active				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

enc1IsOn (N, SEMA)				
Position measuring system 1 is enabled Operating status of position measuring system 1				
Return value	0 = Position measuring system 1 parked (or is not configured), may be removed 1 = Position measuring system 1 is passive 2 = Position measuring system 1 is active (e.g. position control)			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	2
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

enc2IsOn (N, SEMA)				
Position measuring system 2 is enabled Operating status of position measuring system 2				
Return value	0 = Position measuring system 2 parked (or is not configured), may be removed 1 = Position measuring system 2 is passive 2 = Position measuring system 2 is active (e.g. position control)			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	2
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaAxForce (N, SEMA)				
Feedrate force				
Units and value ranges				
Physical unit:	Data type:	Initial value		
N	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaDesValFiltersDeltaPos (N, SEMA)				
Position difference between setpoint filter chains Position difference between the setpoint filter chains				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, degree, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

aaDesValFiltersSelect (N, SEMA)				
Selection of the setpoint filter chains				
Return value	0 = FALSE = Selection of first setpoint filter chain 1 = TRUE = Selection of second setpoint filter chain			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_BOOLEAN	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaDesValFiltersActive (N, SEMA)				
Status of the setpoint filter chains				
Return value	0 = FALSE = The first setpoint filter chain is active 1 = TRUE = The second setpoint filter chain is active			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_BOOLEAN	0		
Addressing				
Line index:			Max. line index:	

vaDesValFiltersActive (N, SEMA)				
			$$maxnumGlobMachAxes	
			Read access	

vaDesValFiltersDelay1 (N, SEMA)				
eff. delay time of the 1st setpoint filter chain effective delay time of the 1st setpoint filter chain				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaDesValFiltersDelay2 (N, SEMA)				
eff. delay time of the 2st setpoint filter chain effective delay time of the 2st setpoint filter chain				
Units and value ranges				
Physical unit:	Data type:	Initial value		
s, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptAccDynNorm (N, SEMA)				
Adaptation factor for acceleration DYNNORM Active adaptation factor for acceleration value DYNNORM				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptAccDynPos (N, SEMA)				
Adaptation factor for acceleration DYNPOS Active adaptation factor for acceleration value DYNPOS				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		

vaCAdaptAccDynPos (N, SEMA)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptAccDynRough (N, SEMA)				
Adaptation factor for acceleration DYNROUGH Active adaptation factor for acceleration value DYNROUGH				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptAccDynSemiFin (N, SEMA)				
Adaptation factor for acceleration DYNSEMIFIN Active adaptation factor for acceleration value DYNSEMIFIN				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptAccDynFinish (N, SEMA)				
Adaptation factor for acceleration DYNFINISH Active adaptation factor for acceleration value DYNFINISH				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptPosCtrlGain (N, SEMA)				
Adaptation factor for position control gain Active adaptation factor for position control gain factor				
Units and value ranges				
Physical unit:	Data type:	Initial value		

vaCAdaptPosCtrlGain (N, SEMA)				
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptFfwTorqueFact (N, SEMA)				
Adaptation factor for torque feedforward control Active adaptation factor for inertia and/or mass of the torque feedforward control				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptDrvAdapt1 (N, SEMA)				
Adaptation factor ADAPT1 for drive parameters Active adaptation factor ADAPT1 for drive parameters				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptDrvAdapt2 (N, SEMA)				
Adaptation factor ADAPT2 for drive parameters Active adaptation factor ADAPT2 for drive parameters				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptDrvAdapt3 (N, SEMA)				
Adaptation factor ADAPT3 for drive parameters Active adaptation factor ADAPT3 for drive parameters				

vaCAdaptDrvAdapt3 (N, SEMA)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptDrvAdapt4 (N, SEMA)				
Adaptation factor ADAPT4 for drive parameters Active adaptation factor ADAPT4 for drive parameters				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptInputLoad (N, SEMA)				
Input value for load adaptation Input value (inertia and/or mass) for load adaptation				
Units and value ranges				
Physical unit:	Data type:	Initial value		
kgm2	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptIsActive (N, SEMA)				
Adaptation is active At least one adaptation factor is active				
Return value	0 = adaptation not active 1 = adaptation active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	1
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptJerkDynNorm (N, SEMA)				
Adaptation factor for jerk DYNNORM Active adaptation factor for jerk value DYNNORM				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptJerkDynPos (N, SEMA)				
Adaptation factor for jerk DYNPOS Active adaptation factor for jerk value DYNPOS				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptJerkDynRough (N, SEMA)				
Adaptation factor for jerk DYNROUGH Active adaptation factor for jerk value DYNROUGH				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptJerkDynSemiFin (N, SEMA)				
Adaptation factor for jerk DYNSEMIFIN Active adaptation factor for jerk value DYNSEMIFIN				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptJerkDynFinish (N, SEMA)				
Adaptation factor for jerk DYNFINISH Active adaptation factor for jerk value DYNFINISH				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptAccDynPrec (N, SEMA)				
Adaptation factor for acceleration DYNPREC Active adaptation factor for acceleration value DYNPREC				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

vaCAdaptJerkDynPrec (N, SEMA)				
Adaptation factor for jerk DYNREC Active adaptation factor for jerk value DYNPREC				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	1.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

3.2.6.21 SMA: State data in the MCS

All state data that are dependent on machine movement and are defined specifically for machine axes (geometry and special axes) are combined in module SMA. Supplementary information is to be found in module SEMA. The individual variables are defined as fields where the line index is the number of the machine axis (assigned to the current channel). The variable "name" in module SMA with the line index in question identifies the axis.

The assignment of the line indices in modules SMA and SEMA is identical.

actIncrVal (N, SMA)		DB31-61, DBB5		H1
Active INC weighting of the axis				
Return value	0 = INC_10000 1 = INC_1000 2 = INC_100 3 = INC_10 4 = INC_1 5 = INC_VAR 6 = INC_JOG_CONT 7 = no incremental mode set			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

actToolBasePos (N, SMA)				
Current setpoint of the axis position in the MCS Determines the current setpoint of axis "x" in the machine coordinate system (MCS) for the corresponding axis. This value contains all axis-specific override components (e.g. Differential Resolver Function, $AA_OFF, external work offset)				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdToolBasePos (N, SMA)				
Pos. setpoint in machine coordinate system (MCS). The axial variable determines the current setpoint in the machine coordinate system (MCS) for the corresponding machine axis. The setpoint is equivalent to the interpolator output value for the current interpolation cycle. The MCS value contains no axial override components (DRF, AA_OFF, ext. work offset, etc.).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

cmdToolBasePos (N, SMA)				
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

extUnit (N, SMA)				
Current physical unit Current physical unit of the axis position				
Return value	0 = mm 1 = inch 2 = degree 3 = indexing position 4 = userdef			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

name (N, SMA)				
Axis name				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

status (N, SMA)				
Axis status				
Return value	0 = travel command in plus direction 1 = travel command in minus direction 2 = exact position coarse reached 3 = exact position fine reached			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

toolBaseDistToGo (N, SMA)				
Tool base distance-to-go Tool base distance-to-go. Physical unit is defined in the variable extUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

toolBaseREPOS (N, SMA)				
Tool base REPOS Tool base REPOS. Physical unit is defined in the variable extUnit (in this module).				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

varIncrVal (N, SMA)				
Settable value for INC_VAR Settable value for INC_VAR. The physical value depends on whether the axis is linear or rotary. Linear axis: unit is 1 mm Rotary axis: unit is 1/1000 degrees				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMachAxes	
Validity:	From NCK version 3.6		Read access	

3.2.6.22 SSP: State data spindle

All status data that refer to the spindle are combined in the module SSP. The individual variables are defined as arrays where the row index is the number of the spindle (assigned to the current channel). The spindle can be identified by reading the variables "name" or "index" in the same module with the respective row index.

The number of spindles can be read from "$$numSpindles" in the module Y in the area C.

Values of 0 or ' ' are supplied for axes which are not spindles. The value SSP:index = 0 indicates that the axis is not a spindle.

acConstCutS (N, SSP)				
Current const. cutting rate Current constant cutting rate				
Units and value ranges				
Physical unit:	Data type:	Initial value		
m/min, ft/min, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSMode (N, SSP)				
Mode according to $AC_SMODE Spindle mode				
Return value	0: No spindle present in channel or spindle is active in another channel or is being used by PLC (FC18) or by synchronized actions. 1: Open-loop speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	0	4
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

actGearStage (N, SSP)				
Actual gear stage of spindle				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

actGearStage (N, SSP)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

actSpeed (N, SSP)				
Actual value of the spindle speed Current speed of the spindle [n]. The sign indicates the direction of rotation. The direction of rotation can vary from machine to machine, and is specified by the machine manufacturer.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

channelNo (N, SSP)				
Channel number of spindle Number of channel in which spindle is configured				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

cmdAngPos (N, SSP)				
Spindle position Spindle position (SPOS)				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdConstCutSpeed (N, SSP)				
Constant cutting speed of the master spindle Constant cutting rate of the master spindle. The requested value for the master spindle differs from SSP:cmdSpeed only if G96 is active. (For a certain OEM customer this variable is now available retroactively in software version 3.2)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

cmdGearStage (N, SSP)				
Requested gear stage				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdGwps (N, SSP)				
Programmed SUG desired value Programmed SUG desired value (SUG is the function "constant perimeter speed of grinding wheel")				
Units and value ranges				
Physical unit:	Data type:			
m/s, ft/s	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdSpeed (N, SSP)				
Spindle speed, desired value Spindle speed desired value				
Units and value ranges				
Physical unit:	Data type:			
rev/min , m/min	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

cmdSpeed (N, SSP)			
		$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6	Read access	

driveLoad (N, SSP)				
Load				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

gwpsActive (N, SSP)				
SUG programming active SUG programming active (SUG is the function "constant perimeter speed of grinding wheel)				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

index (N, SSP)				
Absolute axis index Absolute axis index referred to MD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

name (N, SSP)			
Spindle name Spindle name Note: If several logical spindles are referred to one physical spindle with active spindle conversion and access is made via area N of module SSP2, then the name of the first suitable logical spindle is output.			

name (N, SSP)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

namePhys (N, SSP)				
Name of physical spindle. Name of assigned physical spindle, identical to "name" variable.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

opMode (N, SSP)				
Spindle mode				
Return value	0 = spindle mode 1 = oscillation mode (gear step changeover) 2 = positioning mode 3 = synchronous mode 4 = axis mode			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

psModePos (N, SSP)				
Corresponds to $$/C/SEGA/actToolEdgeCenterPosEns If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$/C/SEGA/actToolEdgeCenterPosEns is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	

psModePos (N, SSP)				
			$$maxnumGlobMachAxes	
			Read access	

psModePosBKS (N, SSP)				
Corresponds to $$/C/SEGA/actProgPosBKS If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$/C/SEGA/actProgPosBKS is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

psModePosS (N, SSP)				
Corresponds to $$cmdToolEdgeCenterPosEnsS If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$cmdToolEdgeCenterPosEnsS is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

speedLimit (N, SSP)				
Current speed limitation for spindle				
Units and value ranges				
Physical unit:	Data type:			
rev/min , m/min	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

speedOvr (N, SSP)				
Spindle override				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			

speedOvr (N, SSP)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

spindleType (N, SSP)				
Spindle type				
Return value	0 = master spindle 1 = no master spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

status (N, SSP)				
Spindle state Spindle status Bit0 = following spindle Bit1 = leading spindle Bit2 = master spindle Bit3 = constant cutting rate (G96) active				
Return value	Bit0 = following spindle Bit1 = leading spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

turnState (N, SSP)				
State of spindle rotation				
Return value	value range to be read via BTSS variable 0 = clockwise 1 = counter-clockwise 2 = stop value range to be read via $ variable 3 = clockwise 4 = counter-clockwise 5 = stop			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

pSMode (N, SSP)				
Prog. spindle mode Last programmed spindle mode				
Return value	0: No spindle configured in channel or spindle is active in another channel or in use by the PLC (FC18) or by synchronized actions. 1: Speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

pSModeS (N, SSP)				
Prog. spindle mode for block search Last programmed spindle mode with block search				
Return value	0: No spindle configured in channel or spindle is active in another channel or in use by the PLC (FC18) or by synchronized actions. 1: Speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSmaxVelo (N, SSP)				
Maximum spindle speed Maximum spindle speed This variable returns the maximum spindle speed for spindle mode. This is formed from the smallest active speed limitation, and cannot be exceeded by speed programming or override > 100%. A speed limitation is indicated by the VDI interface signal DB31..,DBX83.1 'Setpoint speed limited' and by $AC_SPIND_STATE, bit 10 (setpoint speed limited). The cause of the speed limitation (machine, setting data, G code, VDI interface signal etc.) can also be determined with the system variable $AC_SMAXVELO_INFO. If the spindle is in axis mode, then the speed is not limited by $AC_SMAXVELO but instead the machine data (MAX_AX_VELO, ...) typical for axis mode are active.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSmaxVeloInfo (N, SSP)		

Max. identifier for speed-limiting data

Identifier (index) for the speed-limiting data (machine/setting data, etc.)

The system variable provides additional information about $AC_SMAXVELO and returns the definitive data

(machine, setting data, G code, VDI interface etc.) as identifier/index. The index can be used to determine the speed-limiting data

on the basis of the following table of existing spindle speed limitations.

0 No limitation (SERUPRO)

1 Maximum speed (chuck speed) of spindle MD 35100 SPIND_VELO_LIMIT

2 Speed limited to maximum speed in the current gear stage MD 35130 GEAR_STEP_MAX_VELO_LIMIT

3 Speed limited due to position control to 90% of the minimum from MD 35100 and MD 35130 (SPCON, SPOS, poss. with COUPON,..)

4 Speed limited due to position control to MD 35132 GEAR_STEP_PC_MAX_VELO_LIMIT

5 Speed limited to SD 43220 SPIND_MAX_VELO_G26 (G26 S.. or specification from HMI)

6 Speed limited to MD 35160 SPIND_EXTERN_VELO_LIMIT based on the set VDI interface signal DB31,...DBX3.6

7 Speed limited to SD 43230 SPIND_MAX_VELO_LIMS at constant cutting speed (G96, G961, G962, G97, LIMS)

8 Speed limited to safe speed (SG) by Safety Integrated

9 Speed limited by preparation calculations

10 Limitation by drive parameter SINAMICS p1082 to maximum speed of the drive

11 Speed limitation to MD 36300 ENC_FREQ_LIMIT with functions that require a functioning measuring system, e.g. position control and G95, G96, G97, G973, G33, G34, G35 for the master spindle. The limitation takes into account the encoder speed, the MS arrangement (direct/indirect), MS limiting frequency and the current parameter set

12 Speed limited by axis mode. In the case of a synchronous spindle, axis mode is enforced by the leading spindle.

13 Speed of the superimposed motion of the following spindle limited to the residual dynamics remaining following the coupling. A larger proportion of the superimposed motion can be achieved by reducing the speed of the leading spindle, e.g. by programming G26 S, VELOLIM for the leading spindle or VELOLIMA for the following spindle. The coupling factor must be taken into account.

14 Speed of the leading spindle limited due to missing following spindle dynamics or a high transformation ratio

15 Speed of the master spindle limited to MD 35550 DRILL_VELO_LIMIT in the case of tapping with G331, G332

16 Speed limitation due to the programming of VELOLIM

17 Speed limitation by tool parameter $TC_TP_MAX_VELO

18 Not used

19 Not used

20 Speed limited due to NCU link

21 Speed limited by SD43235 SD_SPIND_USER_VELO_LIMIT, user-controlled speed limitation, e.g. tensioning device, chuck speed

22 Speed limited by the programming of VELOLIMA

23 Speed limited by the clamping state of the tool. In the case of a Weiss spindle, the clamping state can be read from $VA_MOT_CLAMPING_STATE[axn].

In oscillation mode (gear stage change), the variable returns the value for spindle mode (speed control mode).

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			

acSmaxVeloInfo (N, SSP)		
Addressing		
Line index:	Max. line index:	
	$$maxnumGlobMachAxes	
	Read access	

acSminVelo (N, SSP)				
Minimum spindle speed Minimum spindle speed This variable returns the minimum spindle speed for speed control mode. This is formed from the highest active speed increase, and cannot be undershot by speed programming or override < 100%. A speed increase is indicated by the VDI interface signal DB31..,DBX83.2 'Setpoint speed increased' and by $AC_SPIND_STATE, bit 11 (setpoint speed increased). The cause of the speed increase (machine, setting data, G code, VDI interface signal etc.) can also be determined with the system variable $AC_SMINVELO_INFO. If the spindle is in axis or positioning mode, then the speed is not increased by $AC_SMINVELO.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSminVeloInfo (N, SSP)		
Min. identifier for speed-limiting data Identifier (index) for the speed-limiting data (machine/setting data, etc.) The system variable provides additional information about $AC_SMAXVELO, and returns the definitive data (machine, setting data, G code, VDI interface, etc.) as identifier/index. The speed-limiting data can be determined with the index from the following table of existing spindle speed limitations. The system variable provides additional information about $AC_SMINVELO, and returns the speed increasing data (machine, setting data) as identifier/index. The speed-increasing data can be determined with the index from the following table of existing spindle speed increases. 0 Not used 1 Not used 2 Lower speed limit (minimum speed) of the current gear stage MD 35140 GEAR_STEP_MIN_VELO_LIMIT 3 Not used 4 Not used 5 Lower speed limit (minimum speed) from SD 43210 SPIND_MIN_VELO_G25 (G25 S.. or specification from HMI) In oscillation mode (gear stage change) and axis mode, the variable returns the values from spindle mode.		

acSminVeloInfo (N, SSP)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSmaxAcc (N, SSP)				
Active acceleration of the spindle Active acceleration of the spindle This variable returns the active acceleration of the spindle for spindle mode. Bit 14 of $AC_SPIND_STATE (spindle accelerating) is set for the duration of the acceleration to the defined setpoint speed. Bit 15 of $AC_SPIND_STATE (spindle braking) is set for the duration of the braking to the defined setpoint speed. Apart from that, the acceleration-determining machine and setting data can be determined with the system variable $AC_SMAXACC_INFO. If the spindle is in axis mode, then $AC_SMAXACC does not return the current acceleration, instead the machine data (MAX_AX_VELO, MAX_AX_ACCEL, ...) typical for axis mode are active.				
Units and value ranges				
Physical unit:	Data type:			
Rev/s2, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSmaxAccInfo (N, SSP)		

Identifier for active spindle acceleration data

Identifier for the active spindle acceleration data

The system variable provides additional information about $AC_SMAXACC and returns the definitive machine data as identifier/index. The index can be used to determine the active acceleration data on the basis of the following table of existing spindle accelerations.

The number range is oriented to the system variable $AC_SMAXVELO_INFO:

0 No acceleration limitation (SERUPRO)

1 Not used

2 Acceleration in speed control mode without position control in the current gear stage MD 35200 GEAR_STEP_SPEEDCTRL_ACCEL

3 Not used

4 Acceleration in the current gear stage based on position control MD 35210 GEAR_STEP_POSCTRL_ACCEL (SPCON, SPOS, poss. with COUPON,..)

5 Not used

6 Not used

7 Not used

8 Not used

9 Acceleration limited by preparation calculations

10 Not used

11 Not used

12 Acceleration limited by axis mode. In the case of a synchronous spindle, the axis mode is enforced by the leading spindle.

13 Acceleration of the superimposed motion of the following spindle limited to the residual dynamics remaining following the coupling

14 Acceleration of the leading spindle due to missing following spindle dynamics or a high transformation ratio

15 Acceleration of the master spindle MD 35212 GEAR_STEP_POSCTRL_ACCEL2 in the case of tapping with G331, G332 (only when the second data set is configured accordingly)

16 Acceleration limited by the configuration of ACC or ACCFXS (synchronized action)

17 Acceleration limited by tool parameter $TC_TP_MAX_ACCEL

18 Not used

19 Acceleration limited in JOG mode by MD 32301 MA_JOG_MAX_ACCEL

20 Acceleration limited due to NCU link

21 Not used

22 Acceleration limited by programming ACCLIMA

23 Not used

In oscillation mode (gear stage change), the variable returns the value for spindle mode (speed control mode).

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			

Addressing

Line index:	Max. line index:

acSmaxAccInfo (N, SSP)		
	$$maxnumGlobMachAxes	
	Read access	

acSpindState (N, SSP)				
Spindle status in speed control mode				
This variable returns the selected states of the spindle. For positioning and axis modes, the variable $AA_IN-POS_STATE[Sn] can also be read.				
Bit 0: "Constant cutting speed active" (VDI interface signal DB31..,DBX84.0)				
Bit 1: "SUG active" (VDI interface signal DB31..,DBX84.1)				
Bit 2: "CLGON active" (VDI interface signal DB31..,DBX84.2)				
Bit 3: "Tapping without compensating chuck" (VDI interface signal DB31..,DBX84.3)				
Bit 4: "Synchronous mode" (following spindle with synchronous spindle coupling) (VDI interface signal DB31..,DBX84.4)				
Bit 5: "Positioning mode" (VDI interface signal DB31..,DBX84.5)				
Bit 6: "Oscillating mode" (gear stage change) (VDI interface signal DB31..,DBX84.6)				
Bit 7: "Speed control mode" (VDI interface signal DB31..,DBX84.7)				
Bit 8: "Spindle programmed" (e.g. M3, M4 S.., FC18, ..) (VDI interface signal DB31..,DBX64.4/5 or 6/7)				
Bit 9: "Speed limit exceeded" (VDI interface signal DB31..,DBX83.0)				
Bit 10: "Setpoint speed limited" (VDI interface signal DB31..,DBX83.1), active if the speed would be greater than the maximum speed as a result of programming or override ($AC_SMAXVELO)				
Bit 11: "Setpoint speed increased" (VDI interface signal DB31..,DBX83.2) active if the speed would be less than the minimum speed as a result of programming or override (system variable $AC_SMINVELO)				
Bit 12: "Spindle in setpoint range" (VDI interface signal DB31..,DBX83.5)				
Bit 13: "Actual direction of rotation right" (VDI interface signal DB31..,DBX83.7)				
Bit 14: "Spindle accelerating" remains active as long as the spindle is accelerating to the defined setpoint speed on the setpoint side.				
Bit 15: "Spindle braking" remains active as long as the spindle is braking to the defined setpoint speed or comes to a standstill on the setpoint side.				
Bit 16: "Spindle stopped" (VDI interface signal DB31..,DBX61.4)				
Bit 17: "Tool with dynamic limitation active" (VDI interface signal DB31..,DBX85.0)				
Bit 18: Reserved				
Bit 19: "Spindle in position" (VDI interface signal DB31..,DBX85.5)				
Bit 20: "Position control active" (VDI interface signal DB31..,DBX61.5)				
Bit 21: "Referenced/synchronized 1" (VDI interface signal DB31..,DBX60.4)				
Bit 22: "Referenced/synchronized 2" (VDI interface signal DB31..,DBX60.5)				
Bit 23: Direction of spindle rotation inverted by interface signal "Invert M3/M4" (DB31..,DBX17.6)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSVC (N, SSP)				
Programmed, active cutting rate				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
			Read access	

acSType (N, SSP)				
Type of spindle programming Type of spindle programming Spindle programmed as: 0 Spindle not programmed 1 Spindle speed, S in rev/min 2 Cutting rate, SVC in m/min or ft/min 3 Constant cutting rate, S in m/min or ft/min 4 Constant grinding wheel peripheral speed, S in m/s or ft/s				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
			Read access	

vcSGear (N, SSP)	$VC_SGEAR[spino]			
Currently active spindle gear stage Variable $VC_SGEAR[spino] determines the currently active spindle gear stage. $AC_SGEAR[spino] determines the defined gear stage in the main run. During search the actual gear stage may differ from the defined gear stage, because during search the gear stages are not changed. Using $VC_SGEAR[spino] and $AC_SGEAR[spino] it can be checked whether a gear stage change is to be performed after a search. The following values are possible: 1: 1st gear stage active .... 5: 5th gear stage active				
Return value	1: 1st gear stage active .... 5: 5th gear stage active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_WORD	0	0	5
			Read access	

acSGear (N, SSP)	$AC_SGEAR[spino]			
Active spindle gear stage Variable $VC_SGEAR[spino] determines the currently active spindle gear stage. $AC_SGEAR[spino] determines the defined gear stage in the main run. During search the actual gear stage may differ from the defined gear stage, because during search the gear stages are not changed. Using $VC_SGEAR[spino] and $AC_SGEAR[spino] it can be checked whether a gear stage change is to be performed after a search. The following values are possible: 1: 1st gear stage active .... 5: 5th gear stage active				
Return value	1: 1st gear stage active .... 5: 5th gear stage active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_WORD	0	0	5
			Read access	

3.2.6.23 SSP2: Spindle status data for spindle conversion

All state data that refer to a spindle, if a spindle converter (logical spindles) is active

acConstCutS (N, SSP2)				
Current const. cutting rate Current constant cutting rate				
Units and value ranges				
Physical unit:	Data type:	Initial value		
m/min, ft/min, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSMode (N, SSP2)				
Mode according to $AC_SMODE Spindle mode				
Return value	0: No spindle present in channel or spindle is active in another channel or is being used by PLC (FC18) or by synchronized actions. 1: Open-loop speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			

acSMode (N, SSP2)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	0	4
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

actGearStage (N, SSP2)				
Actual gear stage of spindle				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

actSpeed (N, SSP2)				
Actual value of the spindle speed Current speed of the spindle [n]. The sign indicates the direction of rotation. The direction of rotation can vary from machine to machine, and is specified by the machine manufacturer.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

channelNo (N, SSP2)				
Channel number of spindle Number of channel in which spindle is configured				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

cmdAngPos (N, SSP2)				
Spindle position Spindle position (SPOS)				
Units and value ranges				
Physical unit:	Data type:			
Degree, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdConstCutSpeed (N, SSP2)				
Constant cutting speed of the master spindle Constant cutting rate of the master spindle. The requested value for the master spindle differs from SSP:cmdSpeed only if G96 is active. (For a certain OEM customer this variable is now available retroactively in software version 3.2)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm/min, inch/min, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

cmdGearStage (N, SSP2)				
Requested gear stage				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdGwps (N, SSP2)				
Programmed SUG desired value Programmed SUG desired value (SUG is the function "constant perimeter speed of grinding wheel")				
Units and value ranges				
Physical unit:	Data type:			
m/s, ft/s	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

cmdGwps (N, SSP2)				
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

cmdSpeed (N, SSP2)				
Spindle speed, desired value Spindle speed desired value				
Units and value ranges				
Physical unit:	Data type:			
rev/min , m/min	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

driveLoad (N, SSP2)				
Load				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

gwpsActive (N, SSP2)				
SUG programming active SUG programming active (SUG is the function "constant perimeter speed of grinding wheel)				
Return value	0 = inactive 1 = active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

index (N, SSP2)				
Absolute axis index Absolute axis index referred to MD				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

index (N, SSP2)				
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

name (N, SSP2)				
Spindle name Spindle name Note: If several logical spindles are referred to one physical spindle with active spindle conversion and access is made via area N of module SSP2, then the name of the first suitable logical spindle is output.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

namePhys (N, SSP2)				
Physical spindle name Name of assigned physical spindle.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

opMode (N, SSP2)				
Spindle mode				
Return value	0 = spindle mode 1 = oscillation mode (gear step changeover) 2 = positioning mode 3 = synchronous mode 4 = axis mode			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

psModePos (N, SSP2)				
Corresponds to $$/C/SEGA/actToolEdgeCenterPosEns If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$/C/SEGA/actToolEdgeCenterPosEns is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

psModePosBKS (N, SSP2)				
Corresponds to $$/C/SEGA/actProgPosBKS If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$/C/SEGA/actProgPosBKS is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

psModePosS (N, SSP2)				
Corresponds to $$cmdToolEdgeCenterPosEnsS If the spindle is in positioning mode ($$pSMode = 2) or axis mode ($$pSMode = 4), the value $$cmdToolEdgeCenterPosEnsS is returned, otherwise 0.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

speedLimit (N, SSP2)				
Current speed limitation for spindle				
Units and value ranges				
Physical unit:	Data type:			
rev/min , m/min	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

speedLimit (N, SSP2)				
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

speedOvr (N, SSP2)				
Spindle override				
Units and value ranges				
Physical unit:	Data type:			
%	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

spindleType (N, SSP2)				
Spindle type				
Return value	0 = master spindle 1 = no master spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

status (N, SSP2)				
Spindle state Spindle status Bit0 = following spindle Bit1 = leading spindle Bit2 = master spindle Bit3 = constant cutting rate (G96) active				
Return value	Bit0 = following spindle Bit1 = leading spindle			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

turnState (N, SSP2)				
State of spindle rotation				
Return value	value range to be read via BTSS variable 0 = clockwise 1 = counter-clockwise 2 = stop value range to be read via $ variable 3 = clockwise 4 = counter-clockwise 5 = stop			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

pSMode (N, SSP2)				
Prog. spindle mode Last programmed spindle mode				
Return value	0: No spindle configured in channel or spindle is active in another channel or in use by the PLC (FC18) or by synchronized actions. 1: Speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

pSModeS (N, SSP2)				
Prog. spindle mode for block search Last programmed spindle mode with block search				
Return value	0: No spindle configured in channel or spindle is active in another channel or in use by the PLC (FC18) or by synchronized actions. 1: Speed control mode 2: Positioning mode 3: Synchronous mode 4: Axis mode			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSmaxVelo (N, SSP2)				
Maximum spindle speed Maximum spindle speed This variable returns the maximum spindle speed for spindle mode. This is formed from the smallest active speed limitation, and cannot be exceeded by speed programming or override > 100%. A speed limitation is indicated by the VDI interface signal DB31..,DBX83.1 'Setpoint speed limited' and by $AC_SPIND_STATE, bit 10 (setpoint speed limited). The cause of the speed limitation (machine, setting data, G code, VDI interface signal etc.) can also be determined with the system variable $AC_SMAXVELO_INFO. If the spindle is in axis mode, then the speed is not limited by $AC_SMAXVELO but instead the machine data (MAX_AX_VELO, ...) typical for axis mode are active.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSmaxVeloInfo (N, SSP2)		

Max. identifier for speed-limiting data

Identifier (index) for the speed-limiting data (machine/setting data, etc.)

The system variable provides additional information about $AC_SMAXVELO and returns the definitive data

(machine, setting data, G code, VDI interface etc.) as identifier/index. The index can be used to determine the speed-limiting data

on the basis of the following table of existing spindle speed limitations.

0 No limitation (SERUPRO)

1 Maximum speed (chuck speed) of spindle MD 35100 SPIND_VELO_LIMIT

2 Speed limited to maximum speed in the current gear stage MD 35130 GEAR_STEP_MAX_VELO_LIMIT

3 Speed limited due to position control to 90% of the minimum from MD 35100 and MD 35130 (SPCON, SPOS, poss. with COUPON,..)

4 Speed limited due to position control to MD 35132 GEAR_STEP_PC_MAX_VELO_LIMIT

5 Speed limited to SD 43220 SPIND_MAX_VELO_G26 (G26 S.. or specification from HMI)

6 Speed limited to MD 35160 SPIND_EXTERN_VELO_LIMIT based on the set VDI interface signal DB31,...DBX3.6

7 Speed limited to SD 43230 SPIND_MAX_VELO_LIMS at constant cutting speed (G96, G961, G962, G97, LIMS)

8 Speed limited to safe speed (SG) by Safety Integrated

9 Speed limited by preparation calculations

10 Limitation by drive parameter SINAMICS p1082 to maximum speed of the drive

11 Speed limitation to MD 36300 ENC_FREQ_LIMIT with functions that require a functioning measuring system, e.g. position control and G95, G96, G97, G973, G33, G34, G35 for the master spindle. The limitation takes into account the encoder speed, the MS arrangement (direct/indirect), MS limiting frequency and the current parameter set

12 Speed limited by axis mode. In the case of a synchronous spindle, axis mode is enforced by the leading spindle.

13 Speed of the superimposed motion of the following spindle limited to the residual dynamics remaining following the coupling. A larger proportion of the superimposed motion can be achieved by reducing the speed of the leading spindle, e.g. by programming G26 S, VELOLIM for the leading spindle or VELOLIMA for the following spindle. The coupling factor must be taken into account.

14 Speed of the leading spindle limited due to missing following spindle dynamics or a high transformation ratio

15 Speed of the master spindle limited to MD 35550 DRILL_VELO_LIMIT in the case of tapping with G331, G332

16 Speed limitation due to the programming of VELOLIM

17 Speed limitation by tool parameter $TC_TP_MAX_VELO

18 Not used

19 Not used

20 Speed limited due to NCU link

21 Speed limited by SD43235 SD_SPIND_USER_VELO_LIMIT, user-controlled speed limitation, e.g. tensioning device, chuck speed

22 Speed limited by the programming of VELOLIMA

23 Speed limited by the clamping state of the tool. In the case of a Weiss spindle, the clamping state can be read from $VA_MOT_CLAMPING_STATE[axn].

In oscillation mode (gear stage change), the variable returns the value for spindle mode (speed control mode).

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			

acSmaxVeloInfo (N, SSP2)		
Addressing		
Line index:	Max. line index:	
	$$maxnumGlobMachAxes	
	Read access	

acSminVelo (N, SSP2)				
Minimum spindle speed Minimum spindle speed This variable returns the minimum spindle speed for speed control mode. This is formed from the highest active speed increase, and cannot be undershot by speed programming or override < 100%. A speed increase is indicated by the VDI interface signal DB31..,DBX83.2 'Setpoint speed increased' and by $AC_SPIND_STATE, bit 11 (setpoint speed increased). The cause of the speed increase (machine, setting data, G code, VDI interface signal etc.) can also be determined with the system variable $AC_SMINVELO_INFO. If the spindle is in axis or positioning mode, then the speed is not increased by $AC_SMINVELO.				
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSminVeloInfo (N, SSP2)		
Min. identifier for speed-limiting data Identifier (index) for the speed-limiting data (machine/setting data, etc.) The system variable provides additional information about $AC_SMAXVELO, and returns the definitive data (machine, setting data, G code, VDI interface, etc.) as identifier/index. The speed-limiting data can be determined with the index from the following table of existing spindle speed limitations. The system variable provides additional information about $AC_SMINVELO, and returns the speed increasing data (machine, setting data) as identifier/index. The speed-increasing data can be determined with the index from the following table of existing spindle speed increases. 0 Not used 1 Not used 2 Lower speed limit (minimum speed) of the current gear stage MD 35140 GEAR_STEP_MIN_VELO_LIMIT 3 Not used 4 Not used 5 Lower speed limit (minimum speed) from SD 43210 SPIND_MIN_VELO_G25 (G25 S.. or specification from HMI) In oscillation mode (gear stage change) and axis mode, the variable returns the values from spindle mode.		

acSminVeloInfo (N, SSP2)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSmaxAcc (N, SSP2)				
Active acceleration of the spindle Active acceleration of the spindle This variable returns the active acceleration of the spindle for spindle mode. Bit 14 of $AC_SPIND_STATE (spindle accelerating) is set for the duration of the acceleration to the defined setpoint speed. Bit 15 of $AC_SPIND_STATE (spindle braking) is set for the duration of the braking to the defined setpoint speed. Apart from that, the acceleration-determining machine and setting data can be determined with the system variable $AC_SMAXACC_INFO. If the spindle is in axis mode, then $AC_SMAXACC does not return the current acceleration, instead the machine data (MAX_AX_VELO, MAX_AX_ACCEL, ...) typical for axis mode are active.				
Units and value ranges				
Physical unit:	Data type:			
Rev/s2, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSmaxAccInfo (N, SSP2)		

Identifier for active spindle acceleration data

Identifier for the active spindle acceleration data

The system variable provides additional information about $AC_SMAXACC and returns the definitive machine data as identifier/index. The index can be used to determine the active acceleration data on the basis of the following table of existing spindle accelerations.

The number range is oriented to the system variable $AC_SMAXVELO_INFO:

0 No acceleration limitation (SERUPRO)

1 Not used

2 Acceleration in speed control mode without position control in the current gear stage MD 35200 GEAR_STEP_SPEEDCTRL_ACCEL

3 Not used

4 Acceleration in the current gear stage based on position control MD 35210 GEAR_STEP_POSCTRL_ACCEL (SPCON, SPOS, poss. with COUPON,..)

5 Not used

6 Not used

7 Not used

8 Not used

9 Acceleration limited by preparation calculations

10 Not used

11 Not used

12 Acceleration limited by axis mode. In the case of a synchronous spindle, the axis mode is enforced by the leading spindle.

13 Acceleration of the superimposed motion of the following spindle limited to the residual dynamics remaining following the coupling

14 Acceleration of the leading spindle due to missing following spindle dynamics or a high transformation ratio

15 Acceleration of the master spindle MD 35212 GEAR_STEP_POSCTRL_ACCEL2 in the case of tapping with G331, G332 (only when the second data set is configured accordingly)

16 Acceleration limited by the configuration of ACC or ACCFXS (synchronized action)

17 Acceleration limited by tool parameter $TC_TP_MAX_ACCEL

18 Not used

19 Acceleration limited in JOG mode by MD 32301 MA_JOG_MAX_ACCEL

20 Acceleration limited due to NCU link

21 Not used

22 Acceleration limited by programming ACCLIMA

23 Not used

In oscillation mode (gear stage change), the variable returns the value for spindle mode (speed control mode).

Units and value ranges

Physical unit:	Data type:			
-	TYPE_DWORD			

Addressing

Line index:	Max. line index:

acSmaxAccInfo (N, SSP2)		
	$$maxnumGlobMachAxes	
	Read access	

acSpindState (N, SSP2)				
Spindle status in speed control mode				
This variable returns the selected states of the spindle. For positioning and axis modes, the variable $AA_IN-POS_STATE[Sn] can also be read.				
Bit 0: "Constant cutting speed active" (VDI interface signal DB31..,DBX84.0)				
Bit 1: "SUG active" (VDI interface signal DB31..,DBX84.1)				
Bit 2: "CLGON active" (VDI interface signal DB31..,DBX84.2)				
Bit 3: "Tapping without compensating chuck" (VDI interface signal DB31..,DBX84.3)				
Bit 4: "Synchronous mode" (following spindle with synchronous spindle coupling) (VDI interface signal DB31..,DBX84.4)				
Bit 5: "Positioning mode" (VDI interface signal DB31..,DBX84.5)				
Bit 6: "Oscillating mode" (gear stage change) (VDI interface signal DB31..,DBX84.6)				
Bit 7: "Speed control mode" (VDI interface signal DB31..,DBX84.7)				
Bit 8: "Spindle programmed" (e.g. M3, M4 S.., FC18, ..) (VDI interface signal DB31..,DBX64.4/5 or 6/7)				
Bit 9: "Speed limit exceeded" (VDI interface signal DB31..,DBX83.0)				
Bit 10: "Setpoint speed limited" (VDI interface signal DB31..,DBX83.1), active if the speed would be greater than the maximum speed as a result of programming or override ($AC_SMAXVELO)				
Bit 11: "Setpoint speed increased" (VDI interface signal DB31..,DBX83.2) active if the speed would be less than the minimum speed as a result of programming or override (system variable $AC_SMINVELO)				
Bit 12: "Spindle in setpoint range" (VDI interface signal DB31..,DBX83.5)				
Bit 13: "Actual direction of rotation right" (VDI interface signal DB31..,DBX83.7)				
Bit 14: "Spindle accelerating" remains active as long as the spindle is accelerating to the defined setpoint speed on the setpoint side.				
Bit 15: "Spindle braking" remains active as long as the spindle is braking to the defined setpoint speed or comes to a standstill on the setpoint side.				
Bit 16: "Spindle stopped" (VDI interface signal DB31..,DBX61.4)				
Bit 17: "Tool with dynamic limitation active" (VDI interface signal DB31..,DBX85.0)				
Bit 18: Reserved				
Bit 19: "Spindle in position" (VDI interface signal DB31..,DBX85.5)				
Bit 20: "Position control active" (VDI interface signal DB31..,DBX61.5)				
Bit 21: "Referenced/synchronized 1" (VDI interface signal DB31..,DBX60.4)				
Bit 22: "Referenced/synchronized 2" (VDI interface signal DB31..,DBX60.5)				
Bit 23: Direction of spindle rotation inverted by interface signal "Invert M3/M4" (DB31..,DBX17.6)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
			Read access	

acSVC (N, SSP2)				
Programmed, active cutting rate				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
			Read access	

acSType (N, SSP2)				
Type of spindle programming Type of spindle programming Spindle programmed as: 0 Spindle not programmed 1 Spindle speed, S in rev/min 2 Cutting rate, SVC in m/min or ft/min 3 Constant cutting rate, S in m/min or ft/min 4 Constant grinding wheel peripheral speed, S in m/s or ft/s				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm/min, inch/min, user defined	TYPE_DOUBLE	0	0	
			Read access	

vcSGear (N, SSP2)	$VC_SGEAR[spino]			
Currently active spindle gear stage Variable $VC_SGEAR[spino] determines the currently active spindle gear stage. $AC_SGEAR[spino] determines the defined gear stage in the main run. During search the actual gear stage may differ from the defined gear stage, because during search the gear stages are not changed. Using $VC_SGEAR[spino] and $AC_SGEAR[spino] it can be checked whether a gear stage change is to be performed after a search. The following values are possible: 1: 1st gear stage active .... 5: 5th gear stage active				
Return value	1: 1st gear stage active .... 5: 5th gear stage active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_WORD	0	0	5
			Read access	

acSGear (N, SSP2)		$AC_SGEAR[spino]		
Active spindle gear stage Variable $VC_SGEAR[spino] determines the currently active spindle gear stage. $AC_SGEAR[spino] determines the defined gear stage in the main run. During search the actual gear stage may differ from the defined gear stage, because during search the gear stages are not changed. Using $VC_SGEAR[spino] and $AC_SGEAR[spino] it can be checked whether a gear stage change is to be performed after a search. The following values are possible: 1: 1st gear stage active .... 5: 5th gear stage active				
Return value	1: 1st gear stage active .... 5: 5th gear stage active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_WORD	0	0	5
			Read access	

3.2.6.24 VSYN: NCK-specific user variable for synchronous actions

This block contains NCK-specific user variables for synchronous actions.

anMarker (N, VSYN)				
Dummy				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

3.2.6.25 Y: System data

The machine tool builder or user configures the control with the help of the machine data. Configuration can only be performed with certain access rights. The configuration of the NC can be read in the system data regardless of current access rights.

accessLevel (N, Y)				
Level of access rights Level of the access rights currently set. Can be changed by entering the password or turning the keyswitch.				
Return value	Value range: 0 = Access level SIEMENS 1 = Access level machine manufacturer 2 = Access level system start-up engineer (machine manufacturer) 3 = Access level end user with password 4 = Access level key switch 3 5 = Access level key switch 2 6 = Access level key switch 1 7 = Access level key switch 0			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

axisType (N, Y)				
Axis types for all machine axes Axis types for all machine axes (necessary for start-up): If a machine axis is addressed via the M module, the units and values are returned with reference to the axis type accessible via this variable. (The absolute machine axis index 1-N_Y_maxnumGlobMachAxes is specified via the line index)				
Return value	0 = Linear axis 1 = Rotary axis			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumGlobMachAxes	
Validity:	From NCK version 3.6		Read access	

basicLengthUnit (N, Y)				
Global basic unit				
Return value	0 = mm 1 = inch 4 = userdef			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

chanAssignment (N, Y)		MD 10010: ASSIGN_CHAN_TO_MODE_GROUP[x] x=ChannelNo		K1
Assignment of each channel to mode group				
Return value	0 = channel does not exist n = channel assigned to mode group n (n is maximum numBAGs (BAG = mode group))			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$maxnumChannels	
Validity:	From NCK version 3.6		Read access	

externCncSystem (N, Y)				
CNC system language CNC system whose part programs must be processed on the SINUMERIK control.				
Return value	0: No external language defined 1: System ISO Dialect0 Milling (obsolete) 2: System ISO Dialect0 Turning (obsolete) 3: External language via OEM application (from P6.2) 4: System ISO Dialect0 Milling (from P7.) 5: System ISO Dialect0 Turning (from P7.) etc.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

extraCuttEdgeParams (N, Y)				
Existing TO edge parameters Bit string that specifies which TO edge parameters are available in addition to the 25 standard parameters.				
Return value	Bit 0: Cutting edge parameter no. 26 valid (ISO Dialect Milling H No.) Bit 1: Cutting edge parameter no. 27 valid (Orientation of the cutting edge) Bit 2: Cutting edge parameter no. 28 valid (L1 of the orientation of the cutting edge) Bit 3: Cutting edge parameter no. 29 valid (L2 of the orientation of the cutting edge) Bit 4: Cutting edge parameter no. 30 valid (L3 of the orientation of the cutting edge) Bit 5: Cutting edge parameter no. 31 valid (L1 of the orientation of the cutting edge normal) Bit 6: Cutting edge parameter no. 32 valid (L2 of the orientation of the cutting edge normal) Bit 7: Cutting edge parameter no. 33 valid (L3 of the orientation of the cutting edge normal) Bit 8: Cutting edge parameter no. 34 valid (number of cutting edge teeth, always set) Bit 9: Cutting edge parameter no. 35 valid (basic angle of rotation of the cutting edge, always set) etc.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

kindOfSumcorr (N, Y)	$MN_MM_KIND_OF_SUMCORR			
Characteristics of total offsets in NC Characteristics of total offsets in NC: Bit No. Value Meaning 0 0 Total offsets are saved at the same time as the tool data. 1 Total offsets are not saved at the same time as the tool data. 1 0 Setup offsets are saved at the same time as the tool data. 1 Setup offsets are not saved at the same time as the tool data. 2 0 If the "Tool management" function is in use: The existing total/setup offsets are not affected when tool status "active" is set. 1 When tool status "active" is set, the existing total offsets are set to zero. The setup offsets are not affected. 3 0 If the "Tool management" function plus "Adapter" is in use: Transformation of total offsets 1 No transformation of total offsets 4 0 No setup offset data sets 1 Setup offset data sets are created additionally, in which case the total offset equals the product of total offset + "fine total offset".				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

maskToolManagement (N, Y)	$MN_MM_TOOL_MANAGEMENT_MASK			
Settings for NC tool management Settings for NCK tool management function Activation of tool management memory with "0" means: The set tool management data do not occupy any memory space. Bit 0=1: Memory for TM-specific data is made available Bit 1=1: Memory for monitoring data is made available Bit 2=1: Memory for user data (CC data) is made available Bit 3=1: Memory for "Consider adjacent location" is made available Bit 5=0: Parameters and function for tool wear monitoring are not available. Bit 5=1: Parameters and function for tool wear monitoring are available and, if bit 1 = 1, the wear monitoring function is also available. Bit 6=0: The wear group function is not available; i.e. parameters $TC_MAMP3, $TC_MAP9 cannot be programmed, $TC_MPP5 is not defined for magazine locations of type 1. Bit 6=1: The wear group function is available; i.e. parameters $TC_MAMP3, $TC_MAP9 can be programmed and wear groups defined. $TC_MPP5 contains the wear group number for location type 1. Bit 7=1: Tool adapter data sets are available. Bit 8=1: Total offsets are available. Bit 9=1: Tools in a turret are handled in OPI variable modules such that they are not "displayed" in tool half-locations, but always displayed in a turret location. Please note, therefore, that tools in a turret remain (in display terms) in their turret location in the event of a tool change. Bit 9=0: Default response; Tools in a turret are "displayed" in the OPI in their actual (according to data) location.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
			Read access	

maxCuttingEdgeNo (N, Y)	$MN_MAX_CUTTING_EDGE_NO			
Maximum value D number Maximum value of D number				
Return value	1 to 32000			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	9	1	32000
			Read access	

maxNumAdapter (N, Y)		$MN_MM_NUM_TOOL_ADAPT-ER, ...NUM_TOOL_ADAPTER_TYPE2		
Max. No. avail. tool adapt. data sets Maximum number of tool adapter data sets available in the NCK Line 1: Contains an image of MD 18104 $MN_MM_NUM_TOOL_ADAPTER. (Not for new developments) Line 2: Maximum number of all types of adapter Line 3: Maximum number of type 1 adapters Line 4: Maximum number of type 2 adapters				
Return value	>0: Maximum number of adapter data sets. If -1 is set as the number of type 1 adapters (in $MN_MM_NUM_TOOL_ADAPTER), an adapter is automatically assigned to each magazine location. Line 2: =0: No adapter data definitions possible. Line 3: =0: No type 1 adapter data definitions possible. Line 4: =0: No type 2 adapter data definitions possible.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	-1	4500
Addressing				
Line index:			Max. line index:	
			4	
			Read access	

maxNumNcusInNcuCluster (N, Y)				
Maximum number of NCUs in the NCU link				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

maxNumSumCorr (N, Y)		$MN_MM_NUM_SUMCORR		
Total number total offsets Total number of total offsets in the T/TOE and T/TOS modules				
Return value	The value -1 means that the number of total offsets equals the number of edges * number of total offsets per edge (MD18108 $MN_MM_NUM_SUM-CORR).			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
			Read access	

maxnumAlarms (N, Y)				
Size of NC alarm buffer Size of NCK alarm buffer (maximum number of pending alarms)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

maxnumChannels (N, Y)				
Maximum number of available channels				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

maxnumContainer (N, Y)				
Maximum number of axis containers Maximum number of available axis containers				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_UWORD		0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

maxnumContainerSlots (N, Y)				
Alternative name:	maxnumContainerPlaces			
Maximum number of slots per axis container Maximum number of available slots per axis container				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

maxnumCuttEdges_Tool (N, Y)	$MN_MAX_CUTTING_EDGE_PER_TOOL	
Max. number edges per tool Max. number of edges per tool		
Return value	1 to 12	

maxnumCuttEdges_Tool (N, Y)		$MN_MAX_CUTTING_EDGE_PER_TOOL		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	9		
			Read access	

maxnumDrives (N, Y)				
Maximum number of available drives				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

maxnumEdgeSC (N, Y)		$MN_MAX_SUMCORR_PERCUT-TING_EDGE		
Max. number total offsets per edge Maximum number of total and setup offsets per cutting edge. See T/TOS				
Return value	0 to 6			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
			Read access	

maxnumEventTypes (N, Y)				
Max. no. of event types for trace protocolling Maximum number of event types for the trace protocolling				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

maxnumGlobMachAxes (N, Y)				
Maximum number of available machine axes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

maxnumTraceProtData (N, Y)				
Max. no of data per data list for trace protocol. Maximum number of data per data list for trace protocolling				
Units and value ranges				
Physical unit:	Data type:			

maxnumTraceProtData (N, Y)				
-	TYPE_UWORD			
			Read access	

maxnumTraceProtDataList (N, Y)				
Max. no. of data per data list for trace protocol Maximum number of data per data list for trace protocolling				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

modeSpindleToolRevolver (N, Y)		MD $MN_MM_TOOL_MANAGE-MENT_MASK Bit 9		
Representation of active tool Representation of tool currently in use in modules magazine location data (T / TP, magazine data, location data) and tool data (T / TD, tool data, general data and T / TV, tool data, directory)				
Return value	0: Previous method: During operation, the tool is removed (in data terms) from its circular magazine location and loaded to the spindle location in the buffer magazine. 1: During operation, the tool remains in its circular magazine locations in the OPI modules. This applies to OPI modules magazine location data (T / TP, magazine data and location data) and tool data (T / TD, tool data, general data and T / TV, tool data, directory and T / AEV, working offsets, directory).			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

nckLogbookSeekPos (N, Y)				
NC logbook				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access and write access	

nckType (N, Y)				
NC type				
Return value	0: 840D pl 1000: FM-NC 2000: 810D pl 3000: 802S 4000: 802D pl 5000: 840Di pl (up to and including SW 6) 6000: SOLUTIONLINE 10700: 840D sl 14000: 802D sl T/M 14000: 802D sl N/G or C/U 14500: 808D 15000: 840Di sl 16000: 828D 21000: 840evo 30000: 840evo-cnci			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

nckVersion (N, Y)		$AN_NCK_VERSION		
NC version NC version Only the digits before the comma of the floating point number are evaluated, the digits after the comma may contain identifiers for development-internal intermediate releases. The digits before the comma includes the official NC identifier for the software release: For software release 3.4 the value of the variable is 34,....				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Validity:	From NCK version 3.6		Read access	

exportRestricted (N, Y)				
Export restriction Export restriction Identification of the software subject to export restriction according to the Federal Office of Economics and Export Control, and Export Control and Customs Compare OPI N/Y exportRestricted				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_BOOLEAN	1		
			Read access	

ncuPerformanceClass (N, Y)				
NCU power class				
Return value	0: No special power class 1: Powerline 2-n: Reserved			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	0	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numGlobalGFrames (N, Y)		$MN_MM_NUM_GLOBAL_G_FRAMES		
Number of channel-independent G frames				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numSearchRunToolParams (N, Y)				
Number of parameters in $$actToolDataBeforeSearch Number of parameters in the data $$actToolDataBeforeSearch in the area C, module S				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	3	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numAnalogInp (N, Y)		MD 10300: FASTIO_ANA_NUM_INPUTS		A2
Number of HW analog inputs				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numAnalogOutp (N, Y)		MD 10310: FASTIO_ANA_NUM_OUTPUTS		A2
Number of HW analog outputs				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numBAGs (N, Y)				
Number of available mode groups				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numBasisFrames (N, Y)		$MN_MM_NUM_GLOBAL_BASE_FRAMES		
Number of channel-independent basic frames				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numChannels (N, Y)				
Number of active channels				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numContainer (N, Y)				
Number of axis containers Number of currently available axis containers				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	$$maxnumContainer
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numContainerSlots (N, Y)				
Alternative name:	numContainerPlaces			
Number of slots per axis container Number of currently available slots per axis container				
Units and value ranges				
Physical unit:	Data type:			Maximum value:
-	TYPE_UWORD			$$maxnumContainer-Slots
Addressing				
Line index:			Max. line index:	
			$$numContainer	
			Read access	

numCuttEdgeParams (N, Y)				
Number of parameters of a cutting edge Number of parameters of a cutting edge in the T/TO module.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numCuttEdgeParams_tao (N, Y)		$MN_MM_NUM_CCS_TOA_PARAM		
Number of Siemens application cutting edge data Number of Siemens application data per cutting edge in the T/TAO module. MD18080 $MN_MM_TOOL_MANAGEMENT_MASK, bit2=1 for the "Magazine management (TMMG)" function is set.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	10
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numCuttEdgeParams_tas (N, Y)		$MN_MM_NUM_CCS_MON_PARAM		
No. of Siemens app. monitoring data/cutting edge Number of Siemens application monitoring data per cutting edge in the T/TAS module.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	10
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numCuttEdgeParams_ts (N, Y)				
Number of monitoring parameters of a cutting edge Number of monitoring parameters of a cutting edge in the T/TS module				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numCuttEdgeParams_tu (N, Y)	MD 18096: MM_CC_TOA_PARAM			
Number of OEM data of a cutting edge Number of OEM data of a cutting edge in the T/TUE module.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numCuttEdgeParams_tus (N, Y)	$MN_MM_NUM_CC_MON_PARAM			
Number of OEM monitoring data per cutting edge Number of monitoring data for the OEM per cutting edge in the T/TUS module.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	10
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numDigitInp (N, Y)	MD 10350: FASTIO_DIG_NUM_INPUTS			A2
Number of HW digital inputs				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numDigitOutp (N, Y)	MD 10360: FASTIO_DIG_NUM_OUT-PUTS			A2
Number of HW digital outputs				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numDrives (N, Y)				
Reserved				

numDrives (N, Y)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numGCodeGroups (N, Y)				
Number of NC instruction groups				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numGCodeGroupsFanuc (N, Y)				
Number NC instruction groups (ISO Dialect) Number of NC instruction groups in ISO Dialect mode (the number for the turning and milling versions is not the same)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numGlobMachAxes (N, Y)				
Number of active machine axes				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numHandWheels (N, Y)				
Maximum number of handwheels				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numMagLocParams_tap (N, Y)		$MN_MM_NUM_CCS_MAGLOC_PARAM		
Number of Siemens app. data per magazine location Number of Siemens application data per magazine location in the T/TAPD module.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

numMagLocParams_tap (N, Y)		$MN_MM_NUM_CCS_MAGLOC_PARAM		
-	TYPE_UWORD	0	0	10
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numMagLocParams_u (N, Y)		$MN_MM_NUM_CC_MAGLOC_PARAM		
No. OEM data per mag. loc. in the T/TUPD module. Number of OEM data per magazine location in the T/TUPD module.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	10
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numMagParams_tam (N, Y)		$MN_MM_NUM_CCS_MAGAZINE_PARAM		
Number of Siemens application data per magazine Number of Siemens application data per magazine in the T/TAMD module.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	10
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numMagParams_u (N, Y)		$MN_MM_NUM_CC_MAGAZINE_PARAM		
Number of OEM data per magazine Number of OEM data per magazine in the T/TUMD module.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	10
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numMagPlaceParams (N, Y)				
No. parameters of a mag. loc. in the T/TP module Number of parameters of a magazine location in the T/TP module				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numMagPlacesMax (N, Y)		MD 18086: $MN_MM_NUM_MAGA-ZINE_LOCATION		FBW
Number of magazine locations the NC can manage. Number of magazine locations the NCK can manage. Important: The locations in the buffer and loading magazine are included in the count.				
Return value	Value range: o =0: The tool management cannot be active because no data can be created.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numMagsMax (N, Y)		MD 18084: $MN_MM_NUM_MAGAZINE		FBW
Number of magazines the NC can manage. Number of magazines that can be created in the NC in all TO areas. Buffer and loading magazines are included in the count. T/TMV modules				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numParams_Adapt (N, Y)				
No. of parameters of an adapter in the T/AD module Number of parameters of a type 1 adapter in the T/AD module. Not suitable for new developments.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	4		
			Read access	

numParams_SC (N, Y)				
No. total offset params per total offset set Number of parameters of a total offset or setup data set in the T/TOS and T/TOE modules respectively.				
Units and value ranges				
Physical unit:	Data type:	Initial value		

numParams_SC (N, Y)				
-	TYPE_UWORD	9		
			Read access	

numPlaceMulti (N, Y)				FBW
Number of parameters for a multiple assignment Number of distance relationships that a loading location or tool holder can have. See T/TPM module.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numPlaceMultiParams (N, Y)				FBW
Number of parameters of a multiple assignment Number of parameters for multiple assignments of a location to magazines. Two parameters ($TC_MDP1 and $TC_MDP2) are currently possible.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numToBaust (N, Y)		MD 18110: MM_NUM_TOA_MODULES		
Number of T areas				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numToolHolderParams (N, Y)				
Number of parameters in $$toolHolderData Number of parameters in the data $$toolHolderData in the area C, module S				
Return value	Number of parameters in $$toolHolderData. If the "flat D number" is active, the value = 0 will be returned.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	5	0	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numToolParams_tad (N, Y)		$MN_MM_NUM_CCS_TDA_PARAM		
Number of Siemens application tool data per tool Number of Siemens application data per tool in the T/TAD module,				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	10
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numToolParams_tu (N, Y)		MD 18094: MM_CC_TDA_PARAM		
Number of OEM data of a tool Number of OEM data of a tool in the T/TU module.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

numMultiToolParams (N, Y)				
Amount of multitool data Amount of multitool data in T/MTD module				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numMultiToolPlaceParams (N, Y)				
Amount of multitool place data Amount of multitool place data in the T/MTP module				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

maxNumPlacesPerMultitool (N, Y)				
Maximum number of multitool places Maximum number of places per multitool; specified by $MN_MAX_TOOLS_PER_MULTITOOL				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numMultiToolParams_mtud (N, Y)		$MN_MM_NUM_CC_MULTITOOL_PARAM		
Number of OEM data per MT in the T/MTUD module. Number of OEM data per multitool in the T/MTUD module.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numMultiToolParams_mtad (N, Y)		$MN_MM_NUM_CCS_MULTITOOL_PARAM		
No. Siemens app. data per multitool loc. in T/MTAD Number of Siemens application data per multitool location in the T/MTAD module.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numMultiToolPlaceParams_mtup (N, Y)		$MN_MM_NUM_CC_MTLOC_PARAM		
Number of MT location-specific OEM parameters Number of OEM data per multitool location in the T/MTUP module.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

numMultiToolPlaceParams_mtup (N, Y)	$MN_MM_NUM_CC_MTLOC_PARAM	
	1	
	Read access	

numMultiToolPlaceParams_mtap (N, Y)		$MN_MM_NUM_CCS_MTLOC_PARAM		
Amount of Siemens-specific multitool place data Number of Siemens application data per multitool location in the T/MTAP module.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numUserFrames (N, Y)		$MN_MM_NUM_GLOB- AL_USER_FRAMES		
Number of channel-independent user frames				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

toolChangeMFunc (N, Y)		MD 22560: TOOL_CHANGE_M_CODE		W1
Number of M function for tool change				
Return value	0 = change on T selection (standard for turning) 1 = change on selection M1.. 99999 = change on selection M99999 (standard for milling M06)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DWORD			
Validity:	From NCK version 3.6		Read access	

typeOfCuttingEdge (N, Y)			
Type of D-number programming Type of D-number programming see MD: MM_TYPE_OF_CUTTING_EDGE			
Return value	Value Meaning 0 no 'flat D number management' active 1 flat D numbers active		

typeOfCuttingEdge (N, Y)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

userScale (N, Y)				
User unit table User unit table with 13 elements (see Start-up Guide 2.4 and machine data)				
Return value	0 = table not active 1 = table active			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

handWheelNr (N, Y)				
Handwheel number PLC interface Handwheel number, as required for selection via the PLC user interface.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_UWORD		0	
Addressing				
Line index:			Max. line index:	
			$$numHandWheels	
			Read access	

maxNoOfChannels (N, Y)				
Maximum number of channels Maximum number of channels that can be activated. This defines the upper limit of the option data $ON_NUM_CHANNELS.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
-	TYPE_UWORD	1	1	
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

maxNoOfProgLevel (N, Y)				
Maximum number of program levels Maximum number of program levels present in the system.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

numOfISOCorr (N, Y)				
Number of tool offset values in ISO mode Number of offset values in the ISO offset memory in ISO2 and ISO3 modes.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	98		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

anLanguageOnHmi (N, Y)	$AN_LANGUAGE_ON_HMI	
Current language set on HMI Current language set on HMI -- [*] available languages		

anLanguageOnHmi (N, Y)	$AN_LANGUAGE_ON_HMI	
Return value	1 German [*] 2 French [*] 3 English (United Kingdom) [*] 4 Spanish [*] 5 Portuguese (Portugal) 6 Italian [*] 7 Dutch [*] 8 Chinese (Simplified) [*] 9 Swedish [*] 10 German (Austria) 11 German (Liechtenstein) 12 German (Luxembourg) 13 German (Switzerland) 15 Norwegian (Bokmål) 16 Norwegian (Nynorsk) 18 Hungarian [*] 19 Finnish [*] 20 French (Belgium) 21 French (Canada) 22 French (Luxembourg) 23 French (Principality of Monaco) 24 French (Switzerland) 26 Greek [*] 28 Czech [*] 30 English (United States) 31 English (Australia) 32 English (Belize) 33 English (Canada) 34 English (Caribbean) 35 English (India) 36 English (Ireland) 37 English (Jamaica) 38 English (Malaysia) 39 English (New Zealand) 40 Spanish (Argentina) 41 Spanish (Bolivia) 42 Spanish (Chile) 43 Spanish (Colombia) 44 Spanish (Costa Rica) 45 Spanish (Dominican Republic) 46 Spanish (Ecuador) 47 Spanish (El Salvador) 48 Spanish (Guatemala) 49 Spanish (Honduras)	

anLanguageOnHmi (N, Y)	$AN_LANGUAGE_ON_HMI	
	50 Portuguese (Brazil) [*] 53 Polish [*] 55 Danish [*] 57 Russian [*] 59 Albanian 60 Italian (Switzerland) 62 Bosnian (Latin, Bosnia and Herzegovina) 63 Bosnian (Cyrillic, Bosnia and Herzegovina) 65 Croatian (Croatia) [*] 66 Croatian (Latin, Bosnia and Herzegovina) 68 Slovak [*] 69 Slovenian [*] 70 Dutch (Belgium) 72 Romanian [*] 73 Romansh (Switzerland) 75 Bulgarian [*] 76 Estonian 77 Georgian 78 Latvian 79 Lithuanian 80 Chinese (Traditional) [*] 81 Chinese (Hong Kong S.A.R.) 82 Chinese (Macao S.A.R.) 83 Chinese (Singapore) 85 Korean [*] 87 Japanese [*] 88 Macedonian 89 Turkish [*] 90 Swedish (Finland) 92 Ukrainian 93 Afrikaans 94 Alsatian (France) 95 Amharic (Ethiopia) 96 Armenian 97 Azeri 98 Bashkir (Russia) 99 Belarusian 100 Arabic (Saudi Arabia) 101 Arabic (Algeria) 102 Arabic (Bahrain) 103 Arabic (Egypt) 104 Arabic (Iraq) 105 Arabic (Jordan) 106 Arabic (Kuwait)	

anLanguageOnHmi (N, Y)	$AN_LANGUAGE_ON_HMI	
	107 Arabic (Lebanon) 108 Arabic (Libya) 109 Arabic (Morocco) 110 Arabic (Oman) 111 Arabic (Qatar) 112 Arabic (Syria) 113 Arabic (Tunisia) 114 Arabic (U.A.E.) 115 Arabic (Yemen) 118 Assamese 119 Bengali 120 Gujarati 121 Hindi 122 Indonesian [*] 123 Kannada 124 Konkani 125 Malayalam 126 Marathi 127 Oriya 128 Punjabi 129 Sanskrit 130 English (Republic of the Philippines) 131 English (Singapore) 132 English (South Africa) 133 English (Trinidad and Tobago) 134 English (Zimbabwe) 137 Pashto (Afghanistan) 138 Dari (Afghanistan) 139 Urdu 140 Spanish (Mexico) 141 Spanish (Nicaragua) 142 Spanish (Panama) 143 Spanish (Paraguay) 144 Spanish (Peru) 145 Spanish (Puerto Rico) 146 Spanish (Spain) 147 Spanish (United States) 148 Spanish (Uruguay) 149 Spanish (Venezuela) 151 Serbian (Latin, Bosnia and Herzegovina) 152 Serbian (Cyrillic, Bosnia and Herzegovina) 155 Malay (Brunei Darussalam) 156 Quechua (Bolivia) 158 Inuktitut (Latin, Canada)	

anLanguageOnHmi (N, Y)	$AN_LANGUAGE_ON_HMI	
	159 Inuktitut (Syllabics, Canada) 160 Mohawk (Mohawk) 162 Mapudungun (Chile) 164 Tibetan (PRC) 165 Yi (PRC) 166 Mongolian (Traditional Mongolian, PRC) 167 Uighur (PRC) 169 Tamazight (Latin, Algeria) 171 Catalan 172 Basque 173 Galician 175 Sami, Northern (Finland) 176 Sami, Inari (Finland) 177 Sami, Skolt (Finland) 180 Breton (France) 181 Corsican (France) 182 Occitan (France) 184 Faroese 186 Tamil 187 Telugu 190 Welsh (United Kingdom) 192 Lower Sorbian (Germany) 193 Upper Sorbian (Germany) 195 Greenlandic (Greenland) 196 Icelandic 198 Irish 200 Persian 201 Syriac 203 Hebrew 204 Kazakh 205 Kyrgyz 206 Yakut (Russia) 207 Tatar 208 Uzbek 210 Khmer (Cambodia) 211 Lao 212 Thai [*] 213 Vietnamese [*] 214 Sinhala (Sri Lanka) 215 Filipino (Philippines) 216 Tajik (Cyrillic, Tajikistan) 217 Turkmen 220 Serbian (Cyrillic) 221 Serbian (Latin)	

anLanguageOnHmi (N, Y)		$AN_LANGUAGE_ON_HMI		
	224 K'iche (Guatemala) 225 Kiswahili 226 Luxembourgish 227 Divehi 228 Maltese 229 Mongolian 230 Malay [*] 231 Nepali (Nepal) 232 Hausa (Latin, Nigeria) 233 Igbo (Nigeria) 234 Yoruba (Nigeria) 235 Frisian (Netherlands) 236 Sami, Southern (Norway) 237 Sami, Northern (Norway) 238 Sami, Lule (Norway) 239 Maori (New Zealand) 240 Quechua (Peru) 241 Kinyarwanda (Rwanda) 242 Wolof (Senegal) 243 Sami, Southern (Sweden) 244 Sami, Northern (Sweden) 245 Sami, Lule (Sweden) 246 Sesotho sa Leboa (South Africa) 247 Setswana (South Africa) 248 isiXhosa (South Africa) 249 isiZulu (South Africa) 250 Quechua (Ecuador)			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	2	0	255
Addressing				
Line index:			Max. line index:	
			1	
			Read access and write access	

driveTypeSupport (N, Y)				
Type of supported drive				
Return value	0 = stepper 1 = digital			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

maxNumToolCarrOffsets (N, Y)				
Maximum number of ToolCarrOffsets Maximum number of tool carrier offset vectors				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

3.2.6.26 YFAFL: NCK instruction group list (Fanuc)

All G functions currently configured for the channels are made available for reading by the NCK. They are configured via machine data. Since the G functions are organized in groups, only one of which can be active at a time, this module is organized as a table.

There are two columns for each G group. The 1st column lists the number of G functions in a group ($$/N/YFAFL/Gruppe_NUM), this corresponds to the number of rows in each subsequent column. This second column contains all the G functions belonging to a group ($$/N/YFAFL/Gruppe).

As a result, the data for a certain G group are calculated via a column offset.

The column offset of each variable is:

2 * (G group number - 1)

The number of G groups is given in the variable "$$numGCodeGroupsFanuc" in area N / module Y. The resultant maximum column offset of the variables is thus 2 * $$numGCodeGroupsFanuc.

The G functions currently active are listed in area C / module SNCF.

Gruppe (N, YFAFL)				
Alternative name:	function			
Instruction group				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$/N/YFAFL/Gruppe_NUM	
			Read access	

Gruppe_NUM (N, YFAFL)				
Alternative name:	groupId			
Number of Fanuc-G functions Number of Fanuc-G functions in the relevant group				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	
-	TYPE_UWORD		0	

Gruppe_NUM (N, YFAFL)		
Addressing		
Line index:	Max. line index:	
	1	
	Read access	

3.2.6.27 YNCFL: NCK instruction group list

All G functions currently configured for the channels are made available for reading by the NCK. They are configured via machine data. Since the G functions are organized in groups, only one of which can be active at a time, this module is organized as a table.

There are two columns for each G group. The 1st column lists the number of G functions in a group ($$/N/YNCFL/Gruppe_NUM), this corresponds to the number of rows in each subsequent column. This second column contains all the G functions belonging to a group ($$/N/YNCFL/Gruppe).

As a result, the data for a certain G group are calculated via a column offset.

The column offset of each variable is:

2 * (G group number - 1)

The number of G groups is given in the variable "$$numGCodeGroups" in area N / module Y. The resultant the maximum column offset of the variables is thus 2 * $$numGCodeGroups.

The G functions currently active are listed in area C / module SNCF.

Gruppe (N, YNCFL)				
Alternative name:	function			
Instruction group (Function) Instruction group				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$/N/YNCFL/Gruppe_NUM	
Validity:	From NCK version 3.6		Read access	

Gruppe_NUM (N, YNCFL)				
Alternative name:	groupId			
Number of G functions (GroupID) Number of G functions in the relevant group				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

Gruppe_NUM (N, YNCFL)			
Addressing			
Line index:		Max. line index:	
		1	
Validity:	From NCK version 3.6	Read access	

3.2.7 Tool: Tool Data

3.2.7.1 AD: Adapter data (not for new developments)

Adapter data are used to define the dimensions of an adapter (L1, L2, L3) for each magazine location and the direction (transformation) of any loaded tools.

The transformation is applied when cutting edge data are processed in the OPI modules TOT, TOST and TOET if the tool is loaded in a magazine location to which adapter data are assigned.

Adapter data exist independently of magazine location data. The magazine location data contain a reference (see TP module, placeData) to the adapter data.

(Not for new developments.)

adaptData (T, AD)				
Adapter data The variable contains the adapter data specified by system variable $TC_ADPT: o Line 1: Adapter geometry length 1 ($TC_ADPT1) o Line 2: Adapter geometry length 2 ($TC_ADPT2) o Line 3: Adapter geometry length 3 ($TC_ADPT3) o Lines 4...n: Adaptation transformation number ($TC_ADPTT[n] Not suitable for new developments.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numParams_Adapt	
			Read access and write access	

3.2.7.2 ADN: Adapter data

Adapter data are used to define the dimensions of an adapter and the direction of any loaded tools of each magazine location.

The transformation is applied when cutting edge data are processed in the OPI modules TOT, TOST and TOET if the tool is loaded in a magazine location to which adapter data are assigned.

Adapter data exist independently of magazine location data. The magazine location data contain a reference (see TP module, placeData) to the adapter data.

Adapters can be inserted between a magazine location or a multitool location and a tool.

The adapters change the length and direction of the tool offsets concerned.

The transformation is applied when cutting edge data are processed in the OPI modules TOT, TOST and TOET if the tool is loaded in a magazine location or a multitool location with an adapter to which the adapter data are assigned.

The adapter data describe such an adapter, and exist independently of the use of the adapter in a magazine or multitool location.

The adapter used is determined by the magazine location or multitool location (see TP module, placeData or MTP module, placeData.

adaptType (T, ADN)		$TC_ADPT_TYPE[n]		
Adapter type Adapter type 0: No adapter defined by the number (=line). 1: Defines adapter type 1. 2: Defines adapter type 2.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
			Read access	

adapt1 (T, ADN)		$TC_ADPT1[n]		
Adapter geometry: Length 1 Adapter geometry: Length 1 Variable belongs to adapter type 1				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0.0		
			Read access and write access	

adapt2 (T, ADN)		$TC_ADPT2[n]		
Adapter geometry: Length 2 Adapter geometry: Length 2 Variable belongs to adapter type 1				
Units and value ranges				
Physical unit:	Data type:	Initial value		

adapt2 (T, ADN)		$TC_ADPT2[n]		
mm, inch, user defined	TYPE_DOUBLE	0.0		
			Read access and write access	

adapt3 (T, ADN)		$TC_ADPT3[n]		
Adapter geometry: Length 3 Adapter geometry: Length 3 Variable belongs to adapter type 1				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0.0		
			Read access and write access	

adaptt (T, ADN)		$TC_ADPTT[n]		
Adapter transformation number Adapter transformation number Variable belongs to adapter type 1				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
			Read access and write access	

adaptOff00 (T, ADN)		$TC_ADPT_OFF[n,0,0]		
Adapter type 2: Offset: 1st element, X component Adapter type 2: Offset 1st element of the angle head adapter Vector component in the X direction Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

adaptOff01 (T, ADN)		$TC_ADPT_OFF[n,0,1]		
Adapter type 2: Offset: 1st element, Y component Adapter type 2: Offset 1st element of the angle head adapter Vector component in the Y direction Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

adaptOff02 (T, ADN)		$TC_ADPT_OFF[n,0,2]		
Adapter type 2: Offset: 1st element, Z component Adapter type 2: Offset 1st element of the angle head adapter Vector component in the Z direction Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

adaptDir00 (T, ADN)		$TC_ADPT_DIR[n,0,0]		
Adapter type 2: 1st dir. comp. of 1st rotary axis Adapter type 2: Direction of rotary axis 1: Direction component of the rotary axis for the angle between the 1st and 2nd elements of the angle head adapter. Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

adaptDir01 (T, ADN)		$TC_ADPT_DIR[n,0,1]		
Adapter type 2: 2st dir. comp. of 1st rotary axis Adapter type 2: Direction of rotary axis 2: Direction component of the rotary axis for the angle between the 1st and 2nd elements of the angle head adapter. Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

adaptDir02 (T, ADN)		$TC_ADPT_DIR[n,0,2]		
Adapter type 2: 3st dir. comp. of 1st rotary axis Adapter type 2: Direction of rotary axis 3: Direction component of the rotary axis for the angle between the 1st and 2nd elements of the angle head adapter. Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

adaptAng0 (T, ADN)		$TC_ADPT_ANG[n,0]		
Adapter type 2: Angle 1 Adapter type 2: Angle 1 Angle between the 1st and 2nd elements of the angle head adapter Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
			Read access and write access	

adaptAngConst0 (T, ADN)		$TC_ADPT_ANG_CONST[n,0]		
Adapter type 2: Angle offset 1 Adapter type 2: Angle offset 1 Angle offset between the 1st and 2nd elements of the angle head adapter Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
			Read access and write access	

adaptOff10 (T, ADN)		$TC_ADPT_OFF[n,1,0]		
Adapter type 2: Offset: 2st element, X component Adapter type 2: Offset 2st element of the angle head adapter Vector component in the X direction Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

adaptOff11 (T, ADN)		$TC_ADPT_OFF[n,1,1]		
Adapter type 2: Offset: 2st element, Y component Adapter type 2: Offset 2st element of the angle head adapter Vector component in the Y direction Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

adaptOff12 (T, ADN)		$TC_ADPT_OFF[n,1,2]		
Adapter type 2: Offset: 2st element, Z component Adapter type 2: Offset 2st element of the angle head adapter Vector component in the Z direction Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

adaptDir10 (T, ADN)		$TC_ADPT_DIR[n,1,0]		
Adapter type 2: 1st dir. comp. of 2st rotary axis Adapter type 2: Direction of rotary axis 1: Direction component of the rotary axis for the angle between the 2st and 3nd elements of the angle head adapter. Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

adaptDir11 (T, ADN)		$TC_ADPT_DIR[n,1,1]		
Adapter type 2: 2st dir. comp. of 2st rotary axis Adapter type 2: Direction of rotary axis 2: Direction component of the rotary axis for the angle between the 2st and 3nd elements of the angle head adapter. Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

adaptDir12 (T, ADN)		$TC_ADPT_DIR[n,1,2]		
Adapter type 2: 3st dir. comp. of 2st rotary axis Adapter type 2: Direction of rotary axis 3: Direction component of the rotary axis for the angle between the 2st and 3nd elements of the angle head adapter. Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

adaptAng1 (T, ADN)		$TC_ADPT_ANG[n,1]		
Adapter type 2: Angle 2 Adapter type 2: Angle 2 Angle between the 2st and 3nd elements of the angle head adapter Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
			Read access and write access	

adaptAngConst1 (T, ADN)		$TC_ADPT_ANG_CONST[n,1]		
Adapter type 2: Angle offset 2 Adapter type 2: Angle offset 2 Angle offset between the 2st and 3nd elements of the angle head adapter Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
			Read access and write access	

adaptOff20 (T, ADN)		$TC_ADPT_OFF[n,2,0]		
Adapter type 2: Offset: 3st element, X component Adapter type 2: Offset 3st element of the angle head adapter Vector component in the X direction Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

adaptOff21 (T, ADN)		$TC_ADPT_OFF[n,2,1]		
Adapter type 2: Offset: 3st element, Y component Adapter type 2: Offset 3st element of the angle head adapter Vector component in the Y direction Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

adaptOff22 (T, ADN)		$TC_ADPT_OFF[n,2,2]		
Adapter type 2: Offset: 3st element, Z component Adapter type 2: Offset 3st element of the angle head adapter Vector component in the Z direction Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
			Read access and write access	

adaptDir20 (T, ADN)		$TC_ADPT_DIR[n,2,0]		
Adapter type 2: 1st dir. comp. of 3st rotary axis Adapter type 2: Direction of rotary axis 1: Direction component of the rotary axis for the angle between the 3rd element of the angle head adapter and the tool. Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

adaptDir21 (T, ADN)		$TC_ADPT_DIR[n,2,1]		
Adapter type 2: 2st dir. comp. of 3st rotary axis Adapter type 2: Direction of rotary axis 2: Direction component of the rotary axis for the angle between the 3rd element of the angle head adapter and the tool. Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

adaptDir22 (T, ADN)		$TC_ADPT_DIR[n,2,2]		
Adapter type 2: 3st dir. comp. of 3st rotary axis Adapter type 2: Direction of rotary axis 3: Direction component of the rotary axis for the angle between the 3rd element of the angle head adapter and the tool. Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
			Read access and write access	

adaptAng2 (T, ADN)		$TC_ADPT_ANG[n,2]		
Adapter type 2: Angle 3 Adapter type 2: Angle 3 Angle after the 2nd element of the angle head adapter Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
			Read access and write access	

adaptAngConst2 (T, ADN)		$TC_ADPT_ANG_CONST[n,2]		
Adapter type 2: Angle offset 3 Adapter type 2: Angle offset 3 Angle offset after the 2nd element of the angle head adapter Variable belongs to type 2 adapter (angle head adapter)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
			Read access and write access	

3.2.7.3 AEV: Working offsets: Directory

The active tool edges are sorted in consecutive ascending D number sequence in the AEV module. This module also contains the essential tool data for each D number entered. "Active" in this case refers to the replacement tools.

(If the "unique D numbers" option is not activated in the NC, the edges are sorted according to ascending ToolIdent and DuploNumber. The D number variable is then set to 0 on all lines in this module.)

The D number assignment is not necessarily unique for active tools. For this reason, the same D number may be entered in several lines (successively).

The line number is a serial number which is not related to the D number.

The number of active tool edges is stored in $$numActDEdges (module AEV), e.g. example 10,

i.e. module AEV contains entries for 10 tool edges. These are sorted in ascending D number sequence. The tool edge with the lowest D number has index (serial number) 1, the next-higher D number index 2, etc. and the edge with the highest D number index 10.

When tools are activated/deactivated and D numbers re-assigned, the entries for a D number change line dynamically.

Module T / AEV is organized as a 1-dimensional variable array and can be used for the following purposes:

- Display all tool edges, including D numbers, of active tools.
- Display associated tool data

The module contains the following information which can be addressed via a column index:

- Single column, in 1st line only. Number of D numbers (lines, tool edges) in the current list
- The other columns apply to all lines, each line contains tool edge data with the following information:
  - D number
  - Internal T number of associated tool
- Tool edge number relative to tool
  - Tool identifier
  - Duplo number
  - Magazine number and
  - Location number of tool

Individual values cannot be altered via this module.

Re-assignment of D numbers and changes in allocation to tools (deactivate, activate replacement tools) and other modifications to data cause changes to $$toolCounter in "C / S Channel-specific status data".

Variable $$modeSpindleToolRevolver (module N/Y, global system data) defines for circular magazines (T / TM, magazine data, general data, MagKind=3) whether the tool in OPI modules "T / TP, magazine data, location data", "T / TD, tool data, general data", "T/TV, tool data, directory" and "T / AEV, working offsets, directory" remains (new functionality) in its circular magazine location during operation or changes to the buffer magazine (earlier behaviour)).

DNo (T, AEV)				
D-number D number of the cutting edge located at the consecutive number (= line index) in the AEV module list. Note: Not to be confused with the cutting edge number (see variable "cuttEdgeNo"). Defined only in connection with the "unique D numbers" function.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numActDEdges	
			Read access	

cuttEdgeNo (T, AEV)				
Tool edge number Cutting edge number of the cutting edge located at the consecutive number (= line index) in the AEV module list. Note: This is not the D number of the cutting edge as used in the NC language (see variable "dNo").				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		1	$$maxnumCuttEdg-es_Tool
Addressing				
Line index:			Max. line index:	
			$$numActDEdges	
			Read access	

duploNo (T, AEV)				
Duplo number Internal tool number of the tool located at the consecutive number (= line index) in the AEV module list.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numActDEdges	
			Read access	

numActDEdges (T, AEV)				
Number of D numbers in the T_AEV module Number of D numbers in the AEV module Defined only in connection with the "Unique D numbers" function. For active tool management: Specifies the number of edges belonging to tools with "active" status (contained in the TO unit) When tool management function is not active: Specifies the number of all edges contained in the TO unit.				
Return value	=0: If the "Unique D number" function is not active, the value "=0" is always returned.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

toolIdent (T, AEV)				
Tool identifier Internal tool number of the tool located at the consecutive number (= line index) in the AEV module list.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numActDEdges	
			Read access	

toolInMag (T, AEV)				
Magazine in which the tool is currently located Magazine number of the magazine in which the tool is currently located, that is contained in the AEV module list with the consecutive number (=line index).				
Return value	o Number of entries > $$numActDEdges: error code is output.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numActDEdges	
			Read access	

toolInPlace (T, AEV)				
Current magazine location of the tool Magazine location number of the location in which the tool is currently located, that is contained in the AEV module list with the consecutive number (=line index).				
Return value	o Number of entries > $$numActDEdges: error code is output.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numActDEdges	
			Read access	

toolNo (T, AEV)				
T number of tool Internal tool number of the tool located at the consecutive number (= line index) in the AEV module list.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numActDEdges	
			Read access	

3.2.7.4 MTAD: Application-specific multitool data

This block contains application-specific multitool data.

siemData (T, MTAD)		$TC_MTPCSx[y] x=ParamNo y=Multitool-No		
Alternative name:	applData			
Multitool parameters for Siemens applications The Siemens application parameters for the multitools for all permissible data types, except the data type "String". The Siemens application data are made available with the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit0=1, bit2=1, bit10=1 and MD 18196 $MN_MM_NUM_CCS_MULTITOOL_PARAM>0. The data types of the individual parameters are specified by MD18197 $MN_MM_TYPE_CCS_MULTITOOL_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE. If there is a read or write access to a Siemens application parameter for multitools of the data type "String", an error string is returned.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0.0		

siemData (T, MTAD)	$TC_MTPCSx[y] x=ParamNo y=Multitool-No	
Addressing		
Line index:	Max. line index:	
	32000	
	Read access and write access	

3.2.7.5 MTAP: Application-specific multitool place data

This block contains application-specific multitool location data.

siemPlaceData (T, MTAP)		$TC_MTPPCSx[y,z] x=ParamNo y=MtNo z=MtPlaceNo		
Alternative name:	applPlaceData			
Siemens application multitool place data Siemens application multitool place data. These parameters can only be used if the machine data $MN_MM_NUM_CCS_MTLOC_PARAM and $MN_MM_TOOL_MANAGEMENT_MASK have been set accordingly. Reserved for SIEMENS applications. The data type in the NCK is specified by MD18199 $MN_MM_TYPE_CCS_MTLOC_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The data type for the OPI is changed to the data type TYPE_DOUBLE, and has to be changed back to the NCK data type by the application.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMultiToolPlaceParams_mtap * $$max-NumPlacesPerMultitool	
			Read access and write access	

3.2.7.6 MTD: Multitool data, general data

This block contains the general data of the multitool data.

multitoolNumLoc (T, MTD)	$TC_MTPN			
Number of locations in the multitool Number of locations that the multitool contains.				
Return value	o = 0: Error has occurred.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$MN_MAX_TOOLS_PER_MULTITOOL

multitoolNumLoc (T, MTD)	$TC_MTPN	
Addressing		
Line index:	Max. line index:	
	32000	
	Read access	

multitoolIdent (T, MTD)		$TC_MTP2		
Multitool identifier				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

multitoolInMag (T, MTD)				
Magazine in which the MT is currently located Magazine number of the magazine in which the multitool with T number "t" is currently located.				
Return value	o >0: Magazine number of the tool with T number = "t". o 0: Error is output, e.g. tool is not loaded			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access	

multitoolInPlace (T, MTD)				
Number of the magazine location of the MT Magazine location number of the multitool				
Return value	o >0: Magazine location number of the tool with T number = "t". o 0: Error is output, e.g. tool is not loaded			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access	

multitoolMyMag (T, MTD)				
Owner magazine of the multitool Number of the owner magazine from which the MT was loaded.				
Return value	o >0: Magazine number of the tool with T number = "t". o 0: Error is output, e.g. tool is not loaded			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	max. Nummer eines def. Magazins
Addressing				
Line index:			Max. line index:	
			32000	
			Read access	

multitoolMyPlace (T, MTD)				
Owner magazine location of the tool Magazine location from which the tool was loaded. A magazine location becomes the owner magazine location if the tool is loaded on a location of magazine location type 1 ($TC_MPP1=1). For fixed-location coded tools in buffers, the number states the location onto which the tool is loaded back into the magazine.				
Return value	o 0: Number of the owner magazine location o 0: Meaning depends on further variables: o $A_TOOLMLN >0: Manual tool o $A_TOOLMLN=0: Error is output, e.g. tool is not loaded			
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	max. Nummer def. Magazinplatz
Addressing				
Line index:			Max. line index:	
			32000	
			Read access	

multitoolStateL (T, MTD)	$TC_MTP8			
Multitool status The status of a multitool is defined by bits 0 - 17. Detailed information can be found in the "Tool Management" Function Manual.				
Return value	Value range: o Value 0: Not enabled o Bit 0: Active MT o Bit 1: Enabled o Bit 2: Disabled o Bit 3: Measured o Bit 4: Prewarning limit reached o Bit 5: MT is being changed o Bit 6: Fixed-location-coded o Bit 7: MT was in use o Bit 8: MT in buffer magazine with transport order o Bit 9: Ignore disabled status of the MT o Bit 10: MT must be unloaded o Bit 11: MT must be loaded o Bit 12: Master tool o Bit 13: Reserved o Bit 14: Marked for 1:1 replacement o Bit 15: Manual tool o Bit 16: MT is disabled if a tool is disabled in the MT o Bit 17: MT is at a disabled magazine location			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

multitoolplace_spec (T, MTD)	$TC_MTP7			
Magazine location type The magazine location type of the multitool is comparable to the magazine location type of a simple tool, and is needed for the empty location search and empty location check (magazine loading processes, tool change). 9999 = Location type not defined				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

multitoolsize_down (T, MTD)		$TC_MTP6		
Number of half locations downwards Size of the multitool downwards in half locations (= toward higher location number). Size 1 means that the half location is occupied in the direction "down" from where the multitool itself is mounted. Consequently, the minimum size is 1. The variable returns the number of half locations that are additionally occupied or reserved downwards.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	1	11
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

multitoolsize_left (T, MTD)		$TC_MTP3		
Number of half locations to the left Size of the multitool to the left in half locations (= toward higher location number). Size 1 means that the half location is occupied in the direction "to the left" where the multitool itself is mounted. Consequently, the minimum size is 1. The variable returns the number of half locations that are additionally occupied or reserved to the left.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	1	11
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

multitoolsize_right (T, MTD)		$TC_MTP4		
Number of half locations to the right Size to the right in half locations (= toward lower location number). Size 1 means that the half location is occupied in the direction "to the right" where the multitool itself is mounted. Consequently, the minimum size is 1. The variable returns the number of half locations that are additionally occupied or reserved to the right.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	1	11
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

multitoolsize_upper (T, MTD)		$TC_MTP5		
Number of half locations upwards Size of the multitool upwards in half locations (= toward lower location number). Size 1 means that the half location is occupied in the direction "upwards" where the multitool itself is mounted. Consequently, the minimum size is 1. The variable returns the number of half locations that are additionally occupied or reserved upwards.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	1	11
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

multitoolPosition (T, MTD)		$TC_MTP_POS		
Multitool posit. Number of the multitool location with a tool in a machining position.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$MN_MAX_TOOLS_PER_MULTITOOL
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

multitoolProtAreaFile (T, MTD)		$TC_MTP_PROTA		
Name of 1st protection area Name of the protection area or name of the file containing the description of the protection area. Reserved, do not use.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

multitoolKindOfDist (T, MTD)		$TC_MTP_KD		
Type of distance coding for MT locations The way in which the location distances (KindOfDistance) are defined in the multitool: o 1 = Distance coding with MT location number o 2 = Distance coding with length specifications o 3 = Distance coding with angle specifications				
Return value	Value range: o 0: Error is output o 1: Multitool with location coding o 2: Multitool with length coding o 3: Multitool with angle coding			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	0	3
Addressing				
Line index:			Max. line index:	
			32000	
			Read access	

multitoolAdaptNo (T, MTD)		$A_TOOLADAPT		
Adapter number at the mag. loc. of the multitool Number of the adapter assigned to the magazine location on which the multitool is mounted.				
Return value	>0: Adapter number 0: Multitool is not in any magazine location or no adapter is assigned to the magazine location.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	32000
Addressing				
Line index:			Max. line index:	
			32000	
			Read access	

3.2.7.7 MTP: Multitool data, place data

This block contains the location data of the multitool data.

mtPlaceData (T, MTP)	diverse, siehe Variablenbeschreibung			
Multitool place data Defines the multitool location data and location distances: o P1: Location distance length ($TC_MTPPL) o P2: Location distance angle ($TC_MTPPA) o P3: Location type (read access only) ($TC_MTPP2) o P4: Location status (bit array) ($TC_MTPP4) Bit 0=1: Disabled Bit 0=0: Disabled Bit 1=1: Free to hold a tool Bit 1=0: Occupied o P5: T number of the tool at this location ($TC_MTPP6) o P6: Number of the adapter at this location ($TC_MTPP7)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numMultiToolPlaceParams * $$maxNumPlacesPerMultitool	
			Read access and write access	

3.2.7.8 MTUD: Multitool data, user-defined data

This block contains the user-defined data of the multitool data.

userData (T, MTUD)	$TC_MTPCx[y] x=ParamNo y=MultitoolNo			
OEM-defined multitool parameters OEM-defined parameter for the multitools for all permissible data types, except the data type "String". The OEM data are made available with the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit0=1, bit2=1, bit10=1 and MD18192 $MN_MM_NUM_CC_MULTITOOL_PARAM>0. The data types of the individual parameters are specified by MD18193 $MN_MM_TYPE_CC_MULTITOOL_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE. If there is a read or write access to an OEM-defined multitool parameter of the data type "String", an error string is returned.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

3.2.7.9 MTUP: Multitool place user data

This block contains the user multitool location data.

userPlaceData (T, MTUP)		**$TC_MTPPCx[y,z] x=ParamNo y=MtNo z=MtPlaceNo**		
OEM-defined multitool location parameters OEM-defined data for the multitool locations for all permissible data types, except the data type "String". The OEM data are made available with the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit0=1, bit2=1, bit10=1 and MD18194 $MN_MM_NUM_CC_MTLOC_PARAM>0. The data type of the individual parameters is specified by MD18195 $MN_MM_TYPE_CC_MTLOC_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMultiToolPlaceParams_mtup * $$max-NumPlacesPerMultitool	
			Read access and write access	

3.2.7.10 MTV: Multitool data, directory

This block contains the directory of the multitool data.

MTnumWZV (T, MTV)				
Last assigned MT number The last assigned internal multitool number in the TOA area.				
Return value	0 = The "Multitool" function is inactive or the multitool has not been defined.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	32000
			Read access	

numMultiTools (T, MTV)		**$P_MTOOLN**		
Number of multitools in the TO area Specifies the number of multitools in the addressed TO area.				
Return value	o >0: Number of defined multitools o 0: Error has occurred, e.g. Multitool function not active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	32000
			Read access	

numLocations (T, MTV)				
Number of places in MT Number of places in the multitool				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$MN_MAX_TOOLS_PER_MULTITOOL
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_MULTITOOL	
			Read access	

multitoolIdent (T, MTV)				
Multitool identifier				
Return value	"" = No multitool, or TMMG is not active			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	""		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_MULTITOOL	
			Read access	

multitoolNo (T, MTV)		$P_MTOOLMT		
Multitool number of the i-th multitool Internal multitool number of the multitool located at the consecutive number (= line index) in the MTV module list .				
Return value	o > 0: Number of the multitool o 0 = The "Multitool" function is not active			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	32000
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_MULTITOOL	
			Read access	

multitoolKindOfDist (T, MTV)				
Type of distance coding for MT locations The location distances (KindOfDistance) within the multitool can be defined in various ways: o 1 = Distance coding with MT location number o 2 = Distance coding with length specifications o 3 = Distance coding with angle specifications The distance coding can only be changed for as long as no MT locations have been generated.				
Return value	Value range: o 0: No multitool, or magazine management (TMMG) is not active o 1: Multitool with location coding o 2: Multitool with length coding o 3: Multitool with angle coding			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	3
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_MULTITOOL	
			Read access	

multitoolInMag (T, MTV)				
Magazine in which the MT is currently located Magazine number of the magazine in which the multitool is currently located, that is contained in the MTV module list with the consecutive number (=line index).				
Return value	o Number of entries > "$$numMultiTool": Error is output.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	32000
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_MULTITOOL	
			Read access	

multitoolInPlace (T, MTV)				
Number of the magazine location of the MT Magazine location number of the location where the multitool is currently located, that is contained in the AEV module list with the consecutive number (=line index).				
Return value	o Number of entries > "$$numMultiTool": Error is output.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	32000

multitoolInPlace (T, MTV)		
Addressing		
Line index:	Max. line index:	
	$MN_MM_NUM_MULTITOOL	
	Read access	

3.2.7.11 TAD: Application-specific data

Data module TAD is organized as a 2-dimensional variable array. This module contains application-specific data for all tools. Every element can be addressed via a column and row index:

The column index is the number of the user-defined tool parameter. The number of tool parameters (columns) can be found in variable $$numToolParams_tad in area N / module Y.

The row index is the tool number. Attempts to access non-existent tools are negatively acknowledged.

Application-specific tool data are all of the same data type.

Application-specific tool data are reserved for SIEMENS applications.

siemData (T, TAD)		$TC_TPCSx[y]		
Alternative name:	applData			
Siemens application tool parameter The Siemens application data are made available by the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit2=1 and MD18204 $MN_MM_NUM_CCS_TDA_PARAM>0 The data types of the individual parameters are specified by MD18205 $MN_MM_TYPE_CCS_TDA_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

3.2.7.12 TAM: Application-specific magazine data

The TAM block contains application-specific information about the available tool magazines.

The application-specific magazine data are reserved for SIEMENS applications.

The TAM block is not intended for new developments.

siemData (T, TAM)				
Alternative name:	applData			
Siemens application magazine data Siemens application magazine data. These parameters can be used only if machine data $MN_MM_NUM_CCS_MAGAZINE_PARAM and $MN_MM_TOOL_MANAGEMENT_MASK are set accordingly. Reserved for SIEMENS applications. This block exists only for compatibility reasons. New developments should be made with the T/TAMD block. (The data format there is "TYPE_DOUBLE")				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMagParams_tam	
			Read access and write access	

3.2.7.13 TAMD: Application-specific magazine data double

The TAMD block contains application-specific information about the available tool magazines.
The application-specific magazine data are reserved for SIEMENS applications.

siemDataDouble (T, TAMD)		$TC_MAPCSx[y]		
Alternative name:	applDataDouble			
Siemens application magazine data Magazine parameters for Siemens applications for all permissible data types, except the data type "String". The Siemens application data are made available with the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit0=1, bit2=1 and MD18200 $MN_MM_NUM_CCS_MAGAZINE_PARAM>0>0. The data type of the individual parameters is specified by MD18201 $MN_MM_TYPE_CCS_MAGAZINE_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE. If there is a read or write access to a magazine parameter of data type "String" for Siemens applications, an error string is returned.				
Units and value ranges				
Physical unit:	Data type:	Initial value		

siemDataDouble (T, TAMD)		$TC_MAPCSx[y]		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMagParams_tam	
			Read access and write access	

3.2.7.14 TAO: Application-specific tool edge data

Data module TAO is organized as a 2-dimensional variable array. This module contains application-specific cutting edge data for all tools. Every element can be addressed via a column and row index. The column index is the tool number (T number), i.e. one column contains the application-specific data for all the cutting edges of a tool.

The assignments between tools and T numbers are listed in the Tool Directory (TV) module in the relevant T area.

A request is negatively acknowledged if a non-existent tool number is entered as the column index.

The number of rows is determined by the number of parameters per cutting edge and the number of cutting edges of a tool:

Max. number of rows = $$numCuttEdgeParams_tao * $$/T/TV/numCuttEdges (T number)

The number of parameters per cutting edge $$numCuttEdgeParams_tao can be found in area N / module Y. The number of tool-specific cutting edges is specified in area T / module TV.

Several rows can be addressed where necessary which means, for example, that all application-specific edge data of a tool can be read in one request.

Application-specific edge data are all of the same data type.

Application-specific cutting edge data are reserved for SIEMENS applications.

siemEdgeData (T, TAO)		$TC_DPCSx[y,z]		
Alternative name:	applEdgeData			
Cutting edge data for Siemens applications The Siemens application data are made available by the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit2=1 and MD18206 $MN_MM_NUM_CCS_TOA_PARAM>0 The data type of the individual parameters is specified by the MD18207 $MN_MM_TYPE_CCS_TDA_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	

siemEdgeData (T, TAO)	$TC_DPCSx[y,z]	
	$$numCuttEdgeParams_tao * $$/T/TV/numCuttEdges	
	Read access and write access	

3.2.7.15 TAP: Application-specific magazine location data

The TAP data block is organized as a 2-dimensional variable array. This block contains application-specific data of a T area. Each element can be addressed via a column and row index:

The column index is the magazine number, i.e. one column contains the application-specific magazine location data for all the locations of one magazine. The assignments between magazines and magazine numbers are listed in the appropriate Magazine Directory (TMV) block in the relevant T area. A request is negatively acknowledged if a non-existent magazine number is entered as the column index.

The number of rows is determined by the number of parameters per magazine location and the number of magazine locations:

Max. number of rows = $$numMagLocParams_tap * $$magNrPlaces

Application-specific magazine location data are reserved for SIEMENS applications.

The TAP block is not intended for new applications.

siemPlaceData (T, TAP)				
Alternative name:	applPlaceData			
Siemens application magazine data Siemens application magazine location data. These parameters can be used only if machine data $MN_MM_NUM_CCS_MAGLOC_PARAM and $MN_MM_TOOL_MANAGEMENT_MASK are set accordingly. Reserved for SIEMENS applications. This block exists only for compatibility reasons. New developments should be made with the T/TAPD block. (The data format there is "TYPE_DOUBLE")				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMagLocParams_tap * $$magNrPlaces	
			Read access and write access	

3.2.7.16 TAPD: Application-specific magazine location data double

The TAPD data block is organized as a 2-dimensional variable array. This block contains application-specific data of a T area. Each element can be addressed via a column and row index:

The column index is the magazine number, i.e. one column contains the application-specific magazine location data for all the locations of one magazine. The assignments between magazines and magazine numbers are listed in the appropriate Magazine Directory (TMV) block in the relevant T area. A request is negatively acknowledged if a non-existent magazine number is entered as the column index.

The number of rows is determined by the number of parameters per magazine location and the number of magazine locations:

Max. number of rows = $$numMagLocParams_tap * $$magNrPlaces

Application-specific magazine location data are reserved for SIEMENS applications.

siemPlaceDataDouble (T, TAPD)		**$TC_MPPCSx[y,z]**		
Alternative name:	applPlaceDataDouble			
Siemens application magazine data Siemens application magazine location data. Magazine location parameters for Siemens applications for all permissible data types, except the data type "String". The magazine location parameters for Siemens applications are made available by the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit0=1, bit2=1 and MD18202 $MN_MM_NUM_CCS_MAGLOC_PARAM>0. The data types of the individual parameters are specified by MD18203 $MN_MM_TYPE_CCS_MAGLOC_PARAM, and can be determined from the file _N_COMPLETE_TMA_ACX. The transfer in the OPI is always as TYPE_DOUBLE. If there is a read or write access to a magazine location parameter of data type "String" for Siemens applications, an error string is returned.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMagLocParams_tap * $$magNrPlaces	
			Read access and write access	

3.2.7.17 TAS: Application-specific monitoring data

Data module TAS is organized as a 2-dimensional variable array. This module contains application-specific monitoring data for all tools. Every element can be addressed via a column and row index:

The column index is the tool number (T number), i.e. one column contains the application-specific monitoring data for all the cutting edges of a tool. The assignments between tools and T numbers are listed in the Tool Directory (TV) module in the relevant T area. A request is negatively acknowledged if a non-existent tool number is entered as the column index.

The number of rows is determined by the number of parameters per cutting edge and the number of cutting edges of a tool:

Max. number of rows = $$numCuttEdgeParams_tas * $$/T/TV/numCuttEdges (T number)

The number of parameters per cutting edge $$numCuttEdgeParams_tas can be found in area N / module Y. The number of tool-specific cutting edges ($$/T/TV/numCuttEdges) is specified in area T / module TV.

Several rows can be addressed where necessary which means, for example, that all application-specific monitoring data of a tool can be read in one request.

Application-specific monitoring data are all of the same data type.

Application-specific monitoring data are reserved for SIEMENS applications.

siemData (T, TAS)		**$TC_MOPCSx[y,z]**		
Alternative name:	applData			
Cutting edge monitoring for Siemens applications Cutting edge monitoring for Siemens applications. The data are made available by the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit1=1 (TMMO) and bit2=1 and MD18208 $MN_MM_NUM_CCS_MON_PARAM>0. The data type of the individual parameters is specified by MD18209 $MN_MM_TYPE_CCS_MON_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_tas * $$/T/TV/numCuttEdges	
			Read access and write access	

3.2.7.18 TC: Toolholder parameters

Module TC contains the data which define an orientatable toolholder (offset vectors, axis directions, rotation angle, type information).

It is also possible to read the current positions of the toolholder axes and the differences between the current and programmed axis values for the active toolholder.

tcCarr10 (T, TC)		$TC_CARR10		
X component of rotary axis v2 X component of the rotary axis v2 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr11 (T, TC)		$TC_CARR11		
Y component of rotary axis v2 Y component of the rotary axis v2 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr12 (T, TC)		$TC_CARR12		
Z component of rotary axis v2				
Z component of the rotary axis v2 of the orientable tool carrier or workpiece holder.				
Connected variables and machine data:				
o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT.				
o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr13 (T, TC)		$TC_CARR13		
Angle of rotation alpha1 (in degrees)				
Angle of rotation alpha1 of the orientable tool carrier or workpiece holder.				
Connected variables and machine data:				
o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT.				
o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr14 (T, TC)		$TC_CARR14		
Angle of rotation alpha2 (in degrees)				
Angle of rotation alpha2 of the orientable tool carrier or workpiece holder.				
Connected variables and machine data:				
o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT.				
o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
Degree	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	

tcCarr14 (T, TC)	$TC_CARR14	
	$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
	Read access and write access	

tcCarr15 (T, TC)		$TC_CARR15		
X component of offset vector l3				
X component of offset vector l3 of the orientable tool carrier or workpiece holder.				
Connected variables and machine data:				
o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on to the setting in MD28085 $MC_MM_LINK_TOA_UNIT.				
o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr16 (T, TC)		$TC_CARR16		
Y component of offset vector l3				
Y component of offset vector l3 of the orientable tool carrier or workpiece holder.				
Connected variables and machine data:				
o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT.				
o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr17 (T, TC)	$TC_CARR17	
Z component of offset vector l3		
Z component of offset vector l3 of the orientable tool carrier or workpiece holder.		
Connected variables and machine data:		
o All variables starting with "$TC_" can apply across channels or be channel-specific according to the setting in MD28085 $MC_MM_LINK_TOA_UNIT.		
o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.		

tcCarr17 (T, TC)		$TC_CARR17		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr18 (T, TC)		$TC_CARR18		
X component of offset vector l4				
X component of offset vector l4 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on to the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr19 (T, TC)		$TC_CARR19		
Y component of offset vector l4				
Y component of offset vector l4 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr1 (T, TC)		$TC_CARR1		
X component of offset vector l1				
X component of offset vector l1 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on to the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr20 (T, TC)		$TC_CARR20		
Z component of offset vector l4				
Z component of offset vector l4 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific according to the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr21 (T, TC)		$TC_CARR21		
Axis identifier of 1st rotary axis				
Axis identifier of the 1st rotary axis of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	

tcCarr21 (T, TC)	$TC_CARR21	
	$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
	Read access and write access	

tcCarr22 (T, TC)		$TC_CARR22		
Axis identifier of 2nd rotary axis Axis identifier of the 2st rotary axis of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr23 (T, TC)		$TC_CARR23		
Kinematic type Kinematic type of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Return value	Possible values and kinematic types: o P: Rotatable workpiece (part) o M: Rotatable tool and rotatable workpiece (mixed) o T: Rotatable tool			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr24 (T, TC)		$TC_CARR24		
Offset of 1st rotary axis in degrees Specifies the angle of the 1st rotary axis with which the axis takes its initial setting. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr25 (T, TC)		$TC_CARR25		
Offset of 2nd rotary axis in degrees Specifies the angle of the 1st rotary axis with which the axis takes its initial setting. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr26 (T, TC)		$TC_CARR26		
Offset of Hirth tooth system of 1st rotary axis Offset of Hirth tooth system in degrees of the 1st rotary axis of the orientable tool carrier or workpiece holder.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr27 (T, TC)		$TC_CARR27		
Offset of Hirth tooth system of 2st rotary axis Offset of Hirth tooth system in degrees of the 2st rotary axis of the orientable tool carrier or workpiece holder.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr28 (T, TC)		$TC_CARR28		
Increment of Hirth tooth system of 1st rotary axis Increment of Hirth tooth system in degrees of 1st rotary axis of the orientable tool carrier or workpiece holder.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr29 (T, TC)		$TC_CARR29		
Increment of Hirth tooth system of 2st rotary axis Increment of Hirth tooth system in degrees of 2st rotary axis of the orientable tool carrier or workpiece holder.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr2 (T, TC)	$TC_CARR2	
Y component of offset vector l1 Y component of offset vector l1 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.		

tcCarr2 (T, TC)		$TC_CARR2		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr30 (T, TC)		$TC_CARR30		
Minimum position of 1st rotary axis Minimum position of 1st rotary axis of the orientable tool carrier or workpiece holder.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr31 (T, TC)		$TC_CARR31		
Minimum position of 2nd rotary axis Minimum position of 2st rotary axis of the orientable tool carrier or workpiece holder.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr32 (T, TC)		$TC_CARR32		
Maximum position of 1st rotary axis Maximum position of 1st rotary axis of the orientable tool carrier or workpiece holder.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr33 (T, TC)		$TC_CARR33		
Maximum position of 2nd rotary axis Maximum position of 2st rotary axis of the orientable tool carrier or workpiece holder.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr34 (T, TC)		$TC_CARR34		
Name of the tool carrier/workpiece holder Contains a freely definable string provided as a freely definable identifier for the orientable tool carrier or workpiece holder. The identifier should not be used for other purposes as a sub-sequent expansion will allow an orientable tool carrier or workpiece holder to be activated via names as well as via numbers.				
Return value	Contains a freely definable string provided as a freely definable identifier for the orientatable toolholder. It has no meaning as yet within the NC and is not evaluated either. The identifier should not be used for other purposes since a later expansion will allow an orientatable toolholder to be activated via a name as well as via numbers			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr35 (T, TC)		$TC_CARR35		
Freely def. identifier (name) of 1st rotary axis. Contains a freely definable string.				
Return value	Contains a freely definable string provided as a free identifier for the first rotary axis. It has no meaning whatsoever within the NC, neither is it evaluated. It can therefore be used for any other purposes.			
Units and value ranges				
Physical unit:	Data type:	Initial value		

tcCarr35 (T, TC)		$TC_CARR35		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr36 (T, TC)		$TC_CARR36		
Name of the 2nd rotary axis. Contains a freely definable string for the 2nd rotary axis.				
Return value	Contains a freely definable string provided as a free identifier for the second rotary axis. It has no meaning whatsoever within the NC, neither is it evaluated. It can therefore be used for any other purposes.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr37 (T, TC)		$TC_CARR37		
Identifier of the tool carrier/workpiece holder Contains a freely definable integer value as the numeric identifier of the orientable tool carrier or workpiece holder.				
Return value	Contains an integer number for identifying the toolholder. It has no meaning whatsoever within the NC, neither is it evaluated.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr38 (T, TC)	$TC_CARR38
X component of the retraction position X component of the retraction position of the orientable tool carrier or workpiece holder.	
Return value	Contains a position (X component of return position). It has no meaning whatsoever within the NC, neither is it evaluated.

tcCarr38 (T, TC)		$TC_CARR38		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr39 (T, TC)		$TC_CARR39		
Y component of the retraction position Y component of the retraction position of the orientable tool carrier or workpiece holder.				
Return value	Contains a position (Y component of return position). It has no meaning whatsoever within the NC, neither is it evaluated.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr3 (T, TC)		$TC_CARR3		
Z component of offset vector l1 Z component of offset vector l1 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific according to the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr40 (T, TC)		$TC_CARR40		
Z component of the retraction position Z component of the retraction position of the orientable tool carrier or workpiece holder.				
Return value	Contains a position (Z component of return position). It has no meaning whatsoever within the NC, neither is it evaluated.			

tcCarr40 (T, TC)		$TC_CARR40		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr41 (T, TC)		$TC_CARR41		
X component of fine offset of offset vector l1				
X component of the fine offset of offset vector l1 of the orientable tool carrier or workpiece holder.				
Connected variables and machine data:				
o All variables starting with "$TC_" can apply across channels or be channel-specific depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT.				
o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr42 (T, TC)		$TC_CARR42		
Y component of fine offset of offset vector l1				
Y component of the fine offset of offset vector l1 of the orientable tool carrier or workpiece holder.				
Connected variables and machine data:				
o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT.				
o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr43 (T, TC)		$TC_CARR43		
Z component of fine offset of offset vector l1				
Z component of the fine offset of offset vector l1 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on to the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr44 (T, TC)		$TC_CARR44		
X component of fine offset of offset vector l2				
X component of the fine offset of offset vector l2 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr45 (T, TC)		$TC_CARR45		
Y component of fine offset of offset vector l2				
Y component of the fine offset of offset vector l2 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr46 (T, TC)		$TC_CARR46		
Z component of fine offset of offset vector l2				
Z component of the fine offset of offset vector l2 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on to the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr4 (T, TC)		$TC_CARR4		
X component of offset vector l2				
X component of offset vector l2 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on to the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr55 (T, TC)		$TC_CARR55		
X component of fine offset of offset vector l3				
X component of the fine offset of offset vector l3 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	

tcCarr55 (T, TC)	$TC_CARR55	
	$MN_MM_NUM_TOOL_CARRIER	
	Read access and write access	

tcCarr56 (T, TC)		$TC_CARR56		
Y component of fine offset of offset vector l3				
Y component of the fine offset of offset vector l3 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr57 (T, TC)		$TC_CARR57		
Z component of fine offset of offset vector l3				
Z component of the fine offset of offset vector l3 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on to the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr58 (T, TC)		$TC_CARR58		
X component of fine offset of offset vector l4				
X component of the fine offset of offset vector l4 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	

tcCarr58 (T, TC)	$TC_CARR58	
Addressing		
Line index:	Max. line index:	
	$MN_MM_NUM_TOOL_CARRIER	
	Read access and write access	

tcCarr59 (T, TC)		$TC_CARR59		
Y component of fine offset of offset vector l4				
Y component of the fine offset of offset vector l4 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr5 (T, TC)		$TC_CARR5		
Y component of offset vector l2				
Y component of offset vector l2 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr60 (T, TC)	$TC_CARR60	
Z component of fine offset of offset vector l4		
Z component of the fine offset of offset vector l4 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on to the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.		

tcCarr60 (T, TC)		$TC_CARR60		
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
mm, inch, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr64 (T, TC)		$TC_CARR64		
Fine offset of the offset of the 1st rotary axis Fine offset of offset ($TC_CARR24) of the 1st rotary axis in degrees of the orientable tool carrier or workpiece holder.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
Degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr65 (T, TC)		$TC_CARR65		
Fine offset of the offset of the 2st rotary axis Fine offset of offset ($TC_CARR25) of the 2st rotary axis in degrees of the orientable tool carrier or workpiece holder.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	
Degree, user defined	TYPE_DOUBLE	0	0	
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr6 (T, TC)		$TC_CARR6		
Z component of offset vector l2 Z component of offset vector l2 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific according to the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	

tcCarr6 (T, TC)	$TC_CARR6	
	$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
	Read access and write access	

tcCarr7 (T, TC)		$TC_CARR7		
X component of rotary axis v1 X component of the rotary axis v1 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr8 (T, TC)		$TC_CARR8		
Y component of rotary axis v1 Y component of the rotary axis v1 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr9 (T, TC)	$TC_CARR9	
Z component of rotary axis v1 Z component of the rotary axis v1 of the orientable tool carrier or workpiece holder. Connected variables and machine data: o All variables starting with "$TC_" can apply across channels or be channel-specific, depending on the setting in MD28085 $MC_MM_LINK_TOA_UNIT. o The maximum number of tool carriers is specified in MD18088 $MN_MM_NUM_TOOL_CARRIER.		

tcCarr9 (T, TC)		$TC_CARR9		
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr_KIN_TOOL_START (T, TC)		$TC_CARR_KIN_TOOL_START		
Start element of the TOOL chain Start element of the TOOL chain for parameterization from kinematic chains.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr_KIN_TOOL_END (T, TC)		$TC_CARR_KIN_TOOL_END		
End element of the TOOL chain. Start element of the TOOL chain for parameterization from kinematic chains.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr_KIN_PART_START (T, TC)		$TC_CARR_KIN_PART_START		
Start element of the part chain Start element of the part chain for parameterization from kinematic chains.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr_KIN_PART_END (T, TC)		$TC_CARR_KIN_PART_END		
End element of the part chain End element of the part chain for parameterization from kinematic chains.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr_KIN_CNTRL (T, TC)		$TC_CARR_KIN_CNTRL		
Data accepted from kinematic chains. Controls the acceptance of data from a kinematic chain.				
Return value	This system data defines which the geometry data of a tool carrier is read from the tool carrier data or from kinematic chains. The system data is bit-coded. At present, only Bit 0 is assigned. Bit 0 : If this bit is set, the offset vectors 1 to l4, the axis direction vectors v1 and v2 as well as the angular offsets of the rotary axis are determined from the kinematic chains, which are referred to with the system data $TC_CARR_KIN_TOOL_START, $TC_CARR_KIN_TOOL_END, $TC_CARR_KIN_PART_START and $TC_CARR_KIN_PART_END.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER	
			Read access and write access	

tcCarr_CORR_ELEM1 (T, TC)		$TC_TCARR_CORR_ELEM1		
Name of the offset vector l1 of a tool carrier. Name of the offset vector l1 of the tool carrier [i].				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr_CORR_ELEM2 (T, TC)		$TC_TCARR_CORR_ELEM2		
Name of the offset vector l2 of a tool carrier. Name of the offset vector l2 of the tool carrier [i].				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr_CORR_ELEM3 (T, TC)		$TC_TCARR_CORR_ELEM3		
Name of the offset vector l3 of a tool carrier. Name of the offset vector l3 of the tool carrier [i].				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr_CORR_ELEM4 (T, TC)		$TC_TCARR_CORR_ELEM4		
Name of the offset vector l4 of a tool carrier. Name of the offset vector l4 of the tool carrier [i].				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

tcCarr_KIN_ROTAX_NAME1 (T, TC)		$TC_CARR_KIN_ROTAX_NAME		
Name of first rotary axis of tool carrier. Name of first rotary axis of tool carrier [i].				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	

tcCarr_KIN_ROTAX_NAME1 (T, TC)	$TC_CARR_KIN_ROTAX_NAME	
	$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
	Read access and write access	

tcCarr_KIN_ROTAX_NAME2 (T, TC)		$TC_CARR_KIN_ROTAX_NAME		
Name of second rotary axis of tool carrier. Name of second rotary axis of tool carrier [i].				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	0		
Addressing				
Line index:			Max. line index:	
			$MN_MM_NUM_TOOL_CARRIER / $$numTo-Baust	
			Read access and write access	

3.2.7.19 TD: Tool data, general data

In addition to the tool offset data other tool characteristics are stored for managing the tools. The module TD contains the general data of the tools. The tool characteristics can be addressed via individual multiple-line variables. The variable line index corresponds to the T-number. If non-existent T-numbers are accessed, the request is acknowledged negatively. The module Tool directory (TV) in the associated T area shows which T-numbers are valid.

adaptNo (T, TD)		$A_TOOLADAPT		
Adapter number at the magazine location of tool Number of the adapter assigned to the magazine location where the tool is mounted. Adapters are described by system variables $TC_ADPTx.				
Return value	Value range: >0: Adapter number 0: No adapter assigned to the magazine location or the tool is not at a magazine location.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	32000
Addressing				
Line index:			Max. line index:	
			max. T-Nummer	
			Read access	

adaptNoOnMultitool (T, TD)		$A_TOOLMTADAPT		
Adapter number at the multitool loc. of the tool Number of the adapter assigned to the multitool location where the tool is mounted.				
Return value	>0: Adapter number at the multitool location 0: No adapter assigned to the multitool location or 0: the tool is not on a multitool.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	32000
Addressing				
Line index:			Max. line index:	
			max. T-Nummer	
			Read access	

duploNo (T, TD)		$TC_TP1		FBW
Duplo number The variable contains the duplo number. In the one TO unit, each tool is uniquely defined by both its identifier and its duplo number.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	T-Nummer		
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access	

numCuttEdges (T, TD)		$P_TOOLND[x] x = ToolNo		
Number of cutting edges of a tool Returns the number of cutting edges of the tool. A tool always has at least one cutting edge.				
Return value	o >0: Number of cutting edges of the tool. o 0: An error has occurred, e.g. tool does not exist.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			1	
Validity:	From NCK version 3.6		Read access	

toolIdent (T, TD)	$TC_TP2	FBW
Tool identifier Tool identifier (name) with up to 32 characters. If no identifier has been specified, the T number is used. If a tool is re-named, the NC checks whether the combination of duplo number and name is unique in the TO unit. If "No", the NC changes the duplo number of the tool so a conflict no longer occurs.		

toolIdent (T, TD)		$TC_TP2		FBW
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_STRING	"<T-Nummer>"		
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access	

toolInMag (T, TD)		$A_TOOLMN[x] x = ToolNo T		
Magazine in which the tool is currently located Magazine number of the magazine in which the tool with T number "t" is currently located.				
Return value	o >0: Magazine number of the tool with T no. = "t". o 0: Error has occurred, e.g. tool is not loaded or tool management is not active.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access	

toolInMultitool (T, TD)		$A_TOOLMTN[x] x = ToolNo T		
Multitool in which the tool is located. Multitool number of the multitool in which the tool with T number "t" is located.				
Return value	o > 0 Number of the multitool containing the tool with T number "t". o 0: Error has occurred, e.g. the tool is not loaded or the MT function or magazine management is not active.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access	

toolInMultitoolPlace (T, TD)		$A_TOOLMTLN[x] x = ToolNo T		
Multitool location number of the tool Multitool location number of the tool with T number "t" in the multitool.				
Return value	o > 0 Location number of the tool with T number "t" in a multitool. o 0: Error has occurred, e.g. the tool is not in the MT, magazine management or MT is not active.			
Units and value ranges				
Physical unit:	Data type:			

toolInMultitoolPlace (T, TD)		$A_TOOLMTLN[x] x = ToolNo T		
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access	

toolInPlace (T, TD)		$A_TOOLMLN[x] x = ToolNo T		
Current magazine location of the tool Magazine location number of the tool with T number "t"				
Return value	o >0: Magazine location number of the tool with T no. = "t". o 0: Error has occurred, e.g. tool is not loaded or the "Multitool" function is not active.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access	

toolInfo (T, TD)		$TC_TP11		FBW
Define tool subgroups Tools are assigned to bit-coded subgroups by variable ($TC_TP11 bit 0 - 7). This is done by assigning the same value ($TC_TP11) to each tool of a subgroup. $P_USEKT specifies which tool subgroups are considered during a tool change. For this purpose, $TC_TP11 and $P_USEKT are rounded bit-by-bit, the tool is only possible if the result is unequal to "0".				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

toolMon (T, TD)		$TC_TP9		FBW
Type of tool monitoring If a monitoring mode is activated for the tool, the current monitoring parameters are evaluated and, if necessary, the tool status is set to "disabled" or "prewarning limit reached". An existing tool disable is not cancelled. This variable is bit-coded to activate various monitoring modes simultaneously.				
Return value	o 0: no tool monitoring o 1: tool life monitoring o 2: no. of workpieces monitoring o 4: monitoring of edge wear parameters using wear limit o 8: monitoring of total offset parameters using wear limit			

toolMon (T, TD)		$TC_TP9		FBW
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

toolMyMag (T, TD)		$A_MYMN		
Owner magazine of the tool Magazine from which the tool was loaded. A magazine becomes the owner magazine if the tool is loaded on a location of magazine location type 1 ($TC_MPP1=1).				
Return value	o 0: Number of the owner magazine o 0: Meaning depends on further variables: o $A_TOOLMLN >0: Manual tool o $A_TOOLMLN=0: Error has occurred, e.g. tool is not loaded or "Multitool" function is not active.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	-	0	max. Nummer eines def. Magazins
Addressing				
Line index:			Max. line index:	
			max. T-Nummer	
			Read access	

toolMyMultitool (T, TD)		$A_MYMTN[x] x = ToolNo T		
Owner multitool of the tool				
Return value	o > 0: The tool with T number "t" is in multitool with the MT number o = 0: Error is output, e.g. tool is not loaded or the "Multitool" function is inactive.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access	

toolMyMultitoolPlace (T, TD)		$A_MYMTLN[x] x = ToolNo T		
Owner multitool place of the tool Owner multitool location of the tool on the multitool.				
Return value	o >0: Number of the owner multitool location of the tool o 0: Error is output, e.g. tool is not loaded or the "Multitool" function is not active.			

toolMyMultitoolPlace (T, TD)		$A_MYMTLN[x] x = ToolNo T		
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access	

toolMyPlace (T, TD)		$A_MYMLN		
Owner magazine location of the tool Magazine location from which the tool was loaded. A magazine location becomes the owner magazine location if the tool at a location of magazine location type 1 ($TC_MPP1=1) is set. For fixed-location coded tools in buffers, the number states the location onto which the tool is loaded back into the magazine.				
Return value	o 0: Number of the owner magazine location o 0: Meaning depends on further variables: o $A_TOOLMLN >0: Manual tool o $A_TOOLMLN=0: Error has occurred, e.g. tool is not loaded or "Multitool" function is not active.			
Units and value ranges				
Physical unit:	Data type:	Initial value		Maximum value:
-	TYPE_UWORD	-		max. Nummer def. Magazinplatz
Addressing				
Line index:			Max. line index:	
			max. T-Nummer	
			Read access	

toolSearch (T, TD)		$TC_TP10		FBW
Order of the replacement tools The value of the variable defines the order of selection of the replacement tools. The tool with the lowest number in $TC_TP10 is loaded. $TC_MAMP2 bit 2 = 1 must be set.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

toolState (T, TD)				FBW
Tool status				
Return value	o Value 0: Not enabled o Bit 0: Active tool o Bit 1: Enabled o Bit 2: Disabled o Bit 3: Measured o Bit 4: Prewarning limit reached o Bit 5: Tool is being changed o Bit 6: Fixed-location-coded o Bit 7: Tool was in use o Bit 8: Tool in buffer magazine with transport order o Bit 9=1: Ignore disabled status of the tool o Bit 9=0: Do not ignore o Bit 10: Tool is to be unloaded o Bit 11: Tool is to be loaded o Bit 12: Tool is master tool o Bit 13: Reserved o Bit 14: Tool marked for 1:1 replacement o Bit 15: Tool in use as manual tool			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

toolplace_spec (T, TD)	$TC_TP7			FBW
Magazine location type of the tool Magazine location type of the tool. Magazine location types are, for example, assigned for the empty location search. The magazine location type entered in the tool data ($TC_TP7) must match the magazine location type ($TC_MPP2) of the magazine. Any location type hierarchies present in $TC_MPTH are taken into account. An exception is magazine location type $TC_MPP2="0" in the magazine to which every tool matches = match all). (The default value = 9999 means location type not defined.)				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	9999		
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

toolsize_down (T, TD)		$TC_TP6		FBW
Tool size downwards in half locations				
Size of the tool downwards in half locations (= toward higher location number.). Size 1 means that the half location is occupied in the downwards direction from where the multitool is mounted. Consequently, the minimum size is 1. The variable returns the number of half locations that are additionally occupied or reserved downwards.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	1	11
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

toolsize_left (T, TD)		$TC_TP3		FBW
Tool size to the left in half locations				
Size of the tool to the left in half locations (= toward lower location number). Size 1 means that the half location is occupied in the direction "to the left" from where the multitool is mounted. Consequently, the minimum size is 1. The variable returns the number of half locations that are additionally occupied or reserved to the left.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	1	11
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

toolsize_right (T, TD)		$TC_TP4		FBW
Tool size to the right in half locations				
Size of the tool to the right in half locations (= toward higher location number.). Size 1 means that the half location to the right is occupied from where the multitool is mounted. Consequently, the minimum size is 1. The variable returns the number of half locations that are additionally occupied or reserved to the right.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	1	11
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

toolsize_upper (T, TD)	$TC_TP5	FBW
Tool size upwards in half locations		
Size of the tool upwards in half locations (= toward lower location number). Size 1 means that the half location is occupied in the upwards direction from where the multitool is mounted. Consequently, the minimum size is 1. The variable returns the number of half locations that are additionally occupied or reserved upwards.		

toolsize_upper (T, TD)		$TC_TP5		FBW
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	1	11
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

toolProtAreaFile (T, TD)				
Reserved, do not use!				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			Read access	

toolMaxVelo (T, TD)		$TC_TP_MAX_VELO		
Maximum speed of tool The variable contains the value for the maximum speed of the tool with T no. = "t". The value must be written before the tool is loaded on the tool holder. The "Magazine management" function must be active.				
Return value	o >0: Value for the angular acceleration limit is evaluated if the tool is mounted on the spindle. o = 0: Speed limit is not active. No speed limit is defined.			
Units and value ranges				
Physical unit:	Data type:			
rev/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

toolMaxAcc (T, TD)		$TC_TP_MAX_ACC		
Maximum angular acceleration of the tool The variable contains the value for the maximum angular acceleration of the tool with T no. = "t". The value must be written before the tool is loaded on the tool holder. The "Magazine management" function must be active.				
Return value	o >0: Value for the angular acceleration limit is evaluated if the tool is mounted on the spindle. o = 0: Angular acceleration limit is not active. No limiting value is defined for the angular acceleration.			

toolMaxAcc (T, TD)		$TC_TP_MAX_ACC		
Units and value ranges				
Physical unit:	Data type:			
Rev/s2, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

toolStateL (T, TD)		$TC_TP8		FBW
Tool status Tool status of the tool with T number "t"				
Return value	Value range: Value 0: Not enabled o Bit 0 = Active tool o Bit 1 = Enabled o Bit 2 = Disabled o Bit 3 = Measured o Bit 4 = Prewarning limit reached o Bit 5 = Tool being changed o Bit 6 = Fixed-location-coded o Bit 7 = Tool was in use o Bit 8 = Identifier for tools in the buffer o Bit 9=1 = Ignore disabled state of tool o Bit 9=0 = Do not ignore o Bit 10 = Tool must be unloaded o Bit 11 = Tool must be loaded o Bit 12 = Reserved. o Bit 13 = Reserved o Bit 14 = 1:1 replacement o Bit 15 = Manual tool o Bit 17 = Tool is at a disabled magazine location			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UDWORD	0		
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

3.2.7.20 TDC: Tool parameters of the Siemens application

Tool parameters of the Siemens application

toolDataChangeInfo (T, TDC)				
Change service for tool data The service works for Siemens applications. The change service outputs changes to tool data. The ring buffer of the tool data change data is organized in columns. Each column contains a change notice The size of the ring buffer is defined in MD 28450 $MC_ MM_TOOL_DATA_CHG_BUFF_SIZE (= max. column index)				
Return value	o Line 1: Column of the last entry in the ring buffer o Line 2: Number of overruns of the ring buffer o Line 3: Block number of the changed date o Line 4: Parameter number of the changed date o Line 5: 1st index of the parameter of the changed date o Line 6: 2nd index of the parameter of the changed date o Line 7: 3rd index of the parameter of the changed date (Only with "ECP", "SCP", distance relationship and new CE number of the cutting edge)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

3.2.7.21 TF: Used as parameter and return value of the PI-Services _N_TMGETT and _N_TSEARC

This module is used for parameterizing as well as for the return parameters of PI services _N_TMGETT and _N_TSEARC. Access to this module must be T area specific and exclusive. It is up to the clients to guarantee this by using the semaphore mechanism (PI service _N_MMCSEM) with the function number for _N_TMSEARCH.

With _N_TMGETT, NO parameterizing elements (input parameters) are relevant; the only relevant one is the result parameter resultToolNr

parDataTAD (T, TF)				
Comparison value for parameter of the TAD module To store a comparison value (data type DOUBLE) for the "complex search" in the TAD module (_N_TSEARC). The value is linked to the corresponding parameter in TAD according to parMaskTAD. The size of the column matches the lines in the TAD module.				
Return value	See module TAD			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numToolParams_tad	
			Read access and write access	

parDataTAO (T, TF)				
Comparison value for parameter of the TAO module To store a comparison value (data type DOUBLE) for the "complex search" in the TAO module (_N_TSEARC). The value is linked to the corresponding parameter in TAO according to parMaskTAO. The size of the column matches the lines in the TAO module.				
Return value	See module TAO			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_tao	
			Read access and write access	

parDataTAS (T, TF)				
Comparison value for parameter of the TAS module To store a comparison value (data type DOUBLE) for the "complex search" in the TAS module (_N_TSEARC). The value is linked to the corresponding parameter in TAS according to parMaskTAS. The size of the column matches the lines in the TAS module.				
Return value	See module TAS			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_tas	
			Read access and write access	

parDataTD (T, TF)				
Alternative name:	parDataToolDuploNo parDataToolToolsize_left parDataToolToolsize_right parDataToolToolsize_upper parDataToolToolsize_down parDataToolToolplace_spec parDataToolToolState parDataToolToolMon parDataToolToolSearch parDataToolToolInfo parDataToolToolInMag parDataToolToolInPlace parDataToolNumCuttEdges			
Comparison value for parameter of the TD module To store a comparison value (data type DOUBLE) for the "complex search" in the TD module (_N_TSEARC). The value is linked to the corresponding parameter in TD according to parMaskTD. The size of the column matches the lines in the TD module.				
Return value	See module TD			

parDataTD (T, TF)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			17	
			Read access and write access	

parDataToolIdentTD (T, TF)				
Comparison val. for par. of TD module (identifier) For parameters with data type string[32] (tool identifier) of the TD module, a value can be stored here as a comparison value for the 'complex search' (_N_TSEARC). The size of the column matches the lines in the TF module.				
Return value	See module TD			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
			Read access and write access	

parDataTO (T, TF)				
Comparison value for parameter of the TO module To store a comparison value (data type DOUBLE) for the "complex search" in the TO module (_N_TSEARC). The value is linked to the corresponding parameter in TO according to parMaskTO. The size of the column matches the lines in the TO module.				
Return value	See module TO			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams * $$maxnumCuttEdg-es_Tool	
			Read access and write access	

parDataTS (T, TF)				
Comparison value for parameter of the TS module To store a comparison value (data type DOUBLE) for the "complex search" in the TS module (_N_TSEARC). The value is linked to the corresponding parameter in TS according to parMaskTS. The size of the column matches the lines in the TS module.				
Return value	See module TS			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

parDataTS (T, TF)		
Addressing		
Line index:	Max. line index:	
	$$numCuttEdgeParams_ts * $$maxnumCut-tEdges_Tool	
	Read access and write access	

parDataTU (T, TF)				
Comparison value for parameter of the TU module To store a comparison value (data type DOUBLE) for the "complex search" in the TU module (_N_TSEARC). The value is linked to the corresponding parameter in TU according to parMaskTU. The size of the column matches the lines in the TU module.				
Return value	See module TU			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numToolParams_tu	
			Read access and write access	

parDataTUE (T, TF)				
Comparison value for parameter of the TUE module To store a comparison value (data type DOUBLE) for the "complex search" in the TUE module (_N_TSEARC). The value is linked to the corresponding parameter in TUE according to parMaskTUE. The size of the column matches the lines in the TUE module.				
Return value	See module TUE			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_tu * $$maxnumCut-tEdges_Tool	
			Read access and write access	

parDataTUS (T, TF)				
Comparison value for parameter of the TUS module To store a comparison value (data type DOUBLE) for the "complex search" in the TUS module (_N_TSEARC). The value is linked to the corresponding parameter in TUS according to $$parMaskTUS. The size of the column matches the lines in the TUS module.				
Return value	See module TUS			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			

parDataTUS (T, TF)		
Addressing		
Line index:	Max. line index:	
	$$numCuttEdgeParams_tus * $$maxnumCut-tEdges_Tool	
	Read access and write access	

parMasksTAD (T, TF)				
Parameterization of the search screenform for TAD A search screenform can be defined for the "complex search" (_N_TUSEARC) for each parameter of the TAD module. The corresponding comparison values are stored in $$parDataTAD. If multiple parameters (i.e. search criteria) have been selected (#0), they are logically combined with AND.				
Return value	o Value 0 : Corresponding operand is not evaluated / Variable is not a criterion for comparison o Value 1 : == (equal) o Value 2 : < (less than) o Value 3 : > (greater than) o Value 4 : <= (less or equal) o Value 5 : >= (greater or equal) o Value 6 : && (bitwise AND, value only allowed for operands of the types WORD and DOU-BLEWORD) o For string operands "==" is the only operator allowed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			$$numToolParams_tad	
			Read access and write access	

parMasksTAO (T, TF)		
Parameterization of the search screenform for TAO A search screenform can be defined for the "complex search" (_N_TUSEARC) for each parameter of the TAO module. The corresponding comparison values are stored in $$parDataTAO. If multiple parameters (i.e. search criteria) have been selected (#0), they are logically combined with AND.		
Return value	o Value 0 : Corresponding operand is not evaluated / Variable is not a criterion for comparison o Value 1 : == (equal) o Value 2 : < (less than) o Value 3 : > (greater than) o Value 4 : <= (less or equal) o Value 5 : >= (greater or equal) o Value 6 : && (bitwise AND, value only allowed for operands of the types WORD and DOU-BLEWORD) o For string operands "==" is the only operator allowed	

parMasksTAO (T, TF)				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_tao	
			Read access and write access	

parMasksTAS (T, TF)				
Parameterization of the search screenform for TAS A search screenform can be defined for the "complex search" (_N_TUSEARC) for each parameter of the TAS module. The corresponding comparison values are stored in $$parDataTAS. If multiple parameters (i.e. search criteria) have been selected (#0), they are logically combined with AND.				
Return value	o Value 0 : Corresponding operand is not evaluated / Variable is not a criterion for comparison o Value 1 : == (equal) o Value 2 : < (less than) o Value 3 : > (greater than) o Value 4 : <= (less or equal) o Value 5 : >= (greater or equal) o Value 6 : && (bitwise AND, value only allowed for operands of the types WORD and DOUBLEWORD) o For string operands "==" is the only operator allowed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_tas	
			Read access and write access	

parMasksTD (T, TF)				
Parameterization of the search screenform for TD A search screenform can be defined for the "complex search" (_N_TUSEARC) for each parameter of the TD module. The corresponding comparison values are stored in $$parDataTD. If multiple parameters (i.e. search criteria) have been selected (#0), they are logically combined with AND.				
Return value	o Value 0 : Corresponding operand is not evaluated / Variable is not a criterion for comparison o Value 1 : == (equal) o Value 2 : < (less than) o Value 3 : > (greater than) o Value 4 : <= (less or equal) o Value 5 : >= (greater or equal) o Value 6 : && (bitwise AND, value only allowed for operands of the types WORD and DOUBLEWORD) o For string operands "==" is the only operator allowed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			17	
			Read access and write access	

parMasksTO (T, TF)				
Parameterization of the search screenform for TO A search screenform can be defined for the "complex search" (_N_TUSEARC) for each parameter of the TO module. The corresponding comparison values are stored in $$parDataTO. If multiple parameters (i.e. search criteria) have been selected (#0), they are logically combined with AND.				
Return value	o Value 0 : Corresponding operand is not evaluated / Variable is not a criterion for comparison o Value 1 : == (equal) o Value 2 : < (less than) o Value 3 : > (greater than) o Value 4 : <= (less or equal) o Value 5 : >= (greater or equal) o Value 6 : && (bitwise AND, value only allowed for operands of the types WORD and DOUBLEWORD) o For string operands "==" is the only operator allowed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	

parMasksTO (T, TF)		
	$$numCuttEdgeParams * $$maxnumCuttEdg-es_Tool	
	Read access and write access	

parMasksTS (T, TF)				
Parameterization of the search screenform for TS A search screenform can be defined for the "complex search" (_N_TUSEARC) for each parameter of the TS module. The corresponding comparison values are stored in $$parDataTS. If multiple parameters (i.e. search criteria) have been selected (#0), they are logically combined with AND.				
Return value	o Value 0 : Corresponding operand is not evaluated / Variable is not a criterion for comparison o Value 1 : == (equal) o Value 2 : < (less than) o Value 3 : > (greater than) o Value 4 : <= (less or equal) o Value 5 : >= (greater or equal) o Value 6 : && (bitwise AND, value only allowed for operands of the types WORD and DOU-BLEWORD) o For string operands "==" is the only operator allowed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_ts * $$maxnumCut-tEdges_Tool	
			Read access and write access	

parMasksTU (T, TF)				
Parameterization of the search screenform for TU A search screenform can be defined for the "complex search" (_N_TUSEARC) for each parameter of the TU module. The corresponding comparison values are stored in $$parDataTU. If multiple parameters (i.e. search criteria) have been selected (#0), they are logically combined with AND.				
Return value	o Value 0 : Corresponding operand is not evaluated / Variable is not a criterion for comparison o Value 1 : == (equal) o Value 2 : < (less than) o Value 3 : > (greater than) o Value 4 : <= (less or equal) o Value 5 : >= (greater or equal) o Value 6 : && (bitwise AND, value only allowed for operands of the types WORD and DOU-BLEWORD) o For string operands "==" is the only operator allowed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:

parMasksTU (T, TF)				
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			$$numToolParams_tu	
			Read access and write access	

parMasksTUE (T, TF)				
Parameterization of the search screenform for TUE A search screenform can be defined for the "complex search" (_N_TUSEARC) for each parameter of the TUE module. The corresponding comparison values are stored in $$parDataTUE. If multiple parameters (i.e. search criteria) have been selected (#0), they are logically combined with AND.				
Return value	o Value 0 : Corresponding operand is not evaluated / Variable is not a criterion for comparison o Value 1 : == (equal) o Value 2 : < (less than) o Value 3 : > (greater than) o Value 4 : <= (less or equal) o Value 5 : >= (greater or equal) o Value 6 : && (bitwise AND, value only allowed for operands of the types WORD and DOUBLEWORD) o For string operands "==" is the only operator allowed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_tu * $$maxnumCuttEdges_Tool	
			Read access and write access	

parMasksTUS (T, TF)				
Parameterization of the search screenform for TUS A search mascreenform sk can be defined for the "complex search" (_N_TUSEARC) for each parameter of the TUS module. The corresponding comparison values are stored in $$parDataTUS. If multiple parameters (i.e. search criteria) have been selected (#0), they are logically combined with AND.				
Return value	o Value 0 : Corresponding operand is not evaluated / Variable is not a criterion for comparison o Value 1 : == (equal) o Value 2 : < (less than) o Value 3 : > (greater than) o Value 4 : <= (less or equal) o Value 5 : >= (greater or equal) o Value 6 : && (bitwise AND, value only allowed for operands of the types WORD and DOUBLEWORD) o For string operands "==" is the only operator allowed			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	6
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_tus * $$maxnumCuttEdges_Tool	
			Read access and write access	

resultCuttingEdgeNrUsed (T, TF)	$A_USEDD			
D numbers of cutting edges since last count D numbers of the cutting edges used since the last workpiece count, that have been used on the tool holder previously defined by $$resultNrOfCutEdgesUsed. Various D offsets for a tool mean that there are multiple entries of the tool. Consequently, a T number can be present more than once. The variables "resultToolNrUsed" and "resultCuttingEdgeNrUsed" form a pair with the same line index consisting of the internal tool number and the cutting edge number for a tool used since the last cutting edge count.				
Return value	o >0: D number (can also be contained multiple times if D offsets of various tools were in use). o 0: No more cutting edges have been used since the last workpiece count. o -1: Magazine management (TMMO) is not active. If tool monitoring (TMMO) is not active, an error code for a missing variable is returned.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	max. Anzahl Schneiden in NC
Addressing				
Line index:			Max. line index:	

resultCuttingEdgeNrUsed (T, TF)	$A_USEDD	
	Zeile 1 * Zeile 2 von $$resultNrOfCutEdge-sUsed	
	Read access	

resultNrOfCutEdgesUsed (T, TF)	$A_USEDND			
Number of cutting edges since last workpiece count Number of different cutting edges used on the tool holder since the last count, including the cutting edge currently in use. Each tool used is included at least once. o Line 1: Number of possible tool holders o Line 2: Number of possible cutting edges. o Line i+2: Number of the i-th tool holder o i+3: Number of cutting edges used on i-th tool holder since the last count. This corresponds to $A_USEDND. The various lines are used to filter the cutting edges displayed.				
Return value	o > 0: Number of cutting edges used o 0: No more cutting edges have been used since the last count. If TMMO is not active, the value for index >2 = -2			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	max. Anzahl Schneiden in NCK
Addressing				
Line index:			Max. line index:	
			2*max.Anz. der Distanzbez. zw.Mag. und WZ-Haltern + 2 = 66	
			Read access	

resultNrOfTools (T, TF)				
Number of tools found The PI services TSEARC and TMGETT search for tools with specific properties and store the number of tools in resultN-rOfTools and the corresponding T numbers in resultToolNr.				
Return value	_N_TMGETT; o 0: No tool found o 1: Tool found _N_TSEARC o >0: Number of tools found o =0: No tool found			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$numTools
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

resultToolNr (T, TF)				
T numbers of the tools found The internal T numbers of the tools found are stored in the individual array elements. The order is random, so they are in the order in which the tools were found by the PI Service.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	31999
Addressing				
Line index:			Max. line index:	
			$$resultNrOfTools	
			Read access	

resultToolNrUsed (T, TF)		$A_USEDT		
T numbers of the tools since the last count The variables "resultToolNrUsed" and "resultCuttingEdgeNrUsed" form with the same line index a pair consisting of internal tool number and the cutting edge number for a cutting edge used since the last count. If a tool is used with different cutting edges, the tool is listed more than once. The two variables are linked to each other. $$resultNrOfCutEdgesUsed has to be read first, and then the individual T numbers with $$resultToolNrUsed.				
Return value	o >0: T number o 0: No more cutting edges have been used since the last count. o -1: Tool monitoring (TMMO) is not active. If tool monitoring (TMMO) is not active, an error code is returned for a missing variable. o -2: "toolHolder" has the value of an undefined tool holder.			
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_DWORD	0	0	max. Anzahl Schneiden in NC
Addressing				
Line index:			Max. line index:	
			Zeile 1 * Zeile 2 von $$resultNrOfCutEdgesUsed	
			Read access	

3.2.7.22 TG: Tool data, grinding-specific data

Special tool data are required for grinding tools. These data are contained in the module TG. They can be addressed via several multiple-row variables. The row index corresponds to the T number. If a non-existent T-number is addressed negative acknowledgment is returned. The module tool directory (TV) in the associated area T shows which T-numbers are valid.

actToolWide (T, TG)		$TC_TPG5		W4
Current wheel width The variable contains the width of the grinding wheel, e.g. after dressing.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

conntectPar (T, TG)		$TC_TPG2		W4
Chaining rule for grinding parameters The variable specifies which tool parameters of cutting edge 2 (D2) and cutting edge 1 (D1) are to be chained. If the value of a chained parameter is changed, the same parameters of the other cutting edge are automatically overwritten with the new value.				
Return value	If the following bits are set, the corresponding parameters of D1 and D2 are chained: Bit0: tool type Bit2: geometry length1 Bit3: geometry length2 Bit4: geometry length3 Bit11: wear length1 Bit12: wear length2 Bit13: wear length3 Bit20: base dimension/adapter dimension length1 Bit21: base dimension/adapter dimension length2 Bit22: base dimension/adapter dimension length3 The value is stored internally as an integer.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

drsPath (T, TG)		$TC_TPG_DRSPATH		
Path to the dressing program Path to the dressing program. Grinding wheels are brought into the correct geometric shape and calibrated with the dressing program.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			

drsPath (T, TG)	$TC_TPG_DRSPATH	
Addressing		
Line index:	Max. line index:	
	32000	
	Read access and write access	

drsProgname (T, TG)		$TC_TPG_DRSPROG		
Name of the dressing program. Name of the dressing program for grinding applications.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

inclAngle (T, TG)		$TC_TPG8		W4
Inclination angle of inclined grinding wheel Angle of inclination of the inclined grinding wheel in the current plane -90° < $TC_TPG8 < 90°				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
Degree	TYPE_DOUBLE		-90	90
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

maxRotSpeed (T, TG)		$TC_TPG6		W4
Maximum speed of the wheel Maximum rotary speed of the grinding wheel				
Units and value ranges				
Physical unit:	Data type:			
rev/min , m/min	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

maxTipSpeed (T, TG)	$TC_TPG7	W4
Maximum peripheral speed Maximum peripheral speed of the grinding wheel		

maxTipSpeed (T, TG)		$TC_TPG7		W4
Units and value ranges				
Physical unit:	Data type:			
mm/min, inch/min, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

minToolDia (T, TG)		$TC_TPG3		W4
Minimum radius of the wheel The minimum radius of the wheel is used for monitoring the geometry of the grinding wheel.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

minToolWide (T, TG)		$TC_TPG4		W4
Minimum width of the grinding wheel The minimum width of the grinding wheel is used for monitoring the geometry of the grinding wheel.				
Units and value ranges				
Physical unit:	Data type:			
mm, inch, user defined	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

paramNrCCV (T, TG)		$TC_TPG9		W4
Compensation parameters for SUG Compensation parameters for the function SUG ("constant perimeter speed of grinding wheel"). These parameters define which compensation value is to be used for SUG, tool monitoring and centerless grinding. The value always refers to cutting edge D1.				
Return value	Value range: o 3: Length 1 o 4: Length 2 o 5: Length 3 o 6: Radius The length consists of geometry + wear + basis, depending on the tool type.			
Units and value ranges				
Physical unit:	Data type:			

paramNrCCV (T, TG)		$TC_TPG9		W4
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

spinNoDress (T, TG)		$TC_TPG1		W4
Referenced spindle number Number of the spindle to be monitored and programmed. The variable contains the number of the spindle to be monitored (e.g. wheel radius and wheel width) and programmed (e.g. wheel peripheral speed).				
Return value	The value is stored internally as an integer.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			32000	
Validity:	From NCK version 3.6		Read access and write access	

3.2.7.23 TISO: ISO tool offset data

This block contains the ISO tool offset data.

isoCorrParam (T, TISO)		$TC_ISO_*		
Offset values for ISO modes This variable contains the offset values for the ISO2.2 and ISO3.2 modes. The column index contains the offset number.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			13	
			Read access and write access	

3.2.7.24 TM: Magazine data, general data

This module contains the information for the available tool magazines.

magActPlace (T, TM)		$TC_MAP8		
Current magazine loc. in relation to change pos. The current magazine position is updated with every magazine movement. The magazine position refers to the change position. Note: The writing of the value is not connected with any positioning of the magazine.				
Return value	The maximum value of the magazine position can be the number of magazine locations in the magazine. Higher and negative values are rejected.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access and write access	

magCmd (T, TM)				
Indicates type of current PI service for magazine. The variable indicates which of the following PI services are active: o _N_TMMVTL o _N_TMPOSM o _N_POSMT and this movement comes from a real magazine or leads to a real magazine. "magCmdState" can be used to determine whether the commands have been completed. Access to the status and results of the PI service can be controlled by semaphores.				
Return value	Value range: o 2 = _N_TMMVTL (move tool) o 3 = _N_TMPOSM (position tool) o 4 = _N_POSMT (position multitool)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magCmdPar1 (T, TM)				
Return parameter 1 to magCmd If there is movement from or to a real magazine, the variable contains the magazine number of the target. In the event of an error, the variable contains the wrong magazine number of the PLC acknowledgment.				
Return value	Value range: o Magazine number o Error number			

magCmdPar1 (T, TM)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magCmdPar2 (T, TM)				
Return parameter 2 to magCmd If there is movement from or to a real magazine, the variable contains the location number of the target. In the event of an error, the variable contains the wrong location number of the PLC acknowledgment.				
Return value	Value range: o Location number o Error number			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magCmdState (T, TM)				
Gives status of current PI service for magazine The processing state of the PI service can be queried with this variable and is stored in "magCmd".				
Return value	Values: o =0: Initial state (after the control has run up). o =1: The PI service has been accepted but not yet output to the PLC. o =2: The command has been output to the PLC, and the NC waits for (final) acknowledgment from the PLC. o =3: The PI service was terminated successfully. The tool has reached its destination location. o =4: The PI service was terminated with an error. This also occurs if the parameterization of the PI service is incorrect or the command with the acknowledgment status=3 (without alarm) was cancelled.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magDim (T, TM)		$TC_MAP6		FBW
Number of lines in the box magazine [n] The variable indicates the number of lines in a box magazine.				
Return value	Applies to box magazines (magKind = 5) number of lines. For all other magazine types the value is 1.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magDim2 (T, TM)		$TC_MAP7		
Number of columns in the box magazine [n] The variable indicates the number of columns in a box magazine.				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	1	1	600
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
			Read access	

magIdent (T, TM)		$TC_MAP2		FBW
Identifier of the magazine [n] The variable indicates the identifier of the magazine.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magKind (T, TM)		$TC_MAP1		FBW
Mag. type Returns the type of magazine.				
Return value	o1 = Chain o3 = Revolver o5 = Box magazine o7 =Tool buffer magazine o9 = Loading magazines			

magKind (T, TM)		$TC_MAP1		FBW
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magLink1 (T, TM)		$TC_MAP4		FBW
Reserved Function not implemented				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	-1		
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magLink2 (T, TM)		$TC_MAP5		FBW
Reserved Function not implemented				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	-1		
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magPlaceUserDataNumLimit (T, TM)		entfaellt		BTSS-Baustein T/TUP
Access to user magazine location data The line number for access to the user magazine location data in the TUP module is calculated as follows: $$numMagLocParams_u * $$magNrPlaces. (MD18092 $MN_MM_NUM_CC_MAGLOC_PARAM * $TC_MAP6[magNo] * $TC_MAP7[magNo]). However, the maximum possible line number is 32767, this means that not all user location data can be addressed. There are (bit-coded) status codes to indicate this state.				
Return value	Value range: o Bit0=1: The product of the current values from the number of magazine locations ($TC_MAP6[magNo] * $TC_MAP7[magNo]) and the number of user magazine location parameters (MD18092 $MN_MM_NUM_CC_MAGLOC_PARAM) exceeds the maximum line number. These values may not yet be effective. o Bit1=1: The product of the effective values from the number of magazine locations ($TC_MAP6[magNo] * $TC_MAP7[magNo]) and the number of user magazine location parameters (MD18092 $MN_MM_NUM_CC_MAGLOC_PARAM) exceeds the maximum line number. Therefore not all user data from all magazine locations of this magazine can be read on the OPI.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	0		
Addressing				
Line index:			Max. line index:	
			1	
			Read access	

magNo (T, TM)				
Number of the magazine Number of the magazine.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		1	$$numMagsMax
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magNrPlaces (T, TM)				
Number of locations in the magazine The variable returns the number of magazine locations. The value for chain magazines corresponds to system variable $TC_MAP7. For box magazines it is the product of TC_MAP6*TC_MAP7 (line times column).				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

magNrPlaces (T, TM)			
Addressing			
Line index:		Max. line index:	
		$$numMagsMax	
Validity:	From NCK version 3.6	Read access	

magState (T, TM)		$TC_MAP3		FBW
State of the magazine [n] The variable returns the current status of the magazine[n]: 1 = Active magazine 2 = Disabled 4 = Magazine in loading position 8 = Movement is active 16 = Enabled for loading				
Return value	1 = current magazine 2 = disabled 4 = magazine in loading position 8 = motion is active 16 = enabled for loading			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_UWORD	2		
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access and write access	

magToolSearchStrat (T, TM)	$TC_MAP10, Bits 0-7			
Tool search strategy Tool search strategy within the magazine. If bit7=1 + bit0=1 or bit2=1, if no "active tool" is found in the magazine, the active tool - if present - is selected from one of the other magazines linked to the tool holder.				
Return value	Search strategy: o Bit0=0 (Default): First available tool from the tool group. Begins the search in the magazine where the last change took place. o Bit0=1: Active tool from the magazine of the last change. Otherwise tool with the lowest duplo number. If there is no match in this magazine, the search is continued in the other linked magazines. o Bit1=1: Next replacement tool nearest to the current magazine position. o Bit2=1: Active tool with the lowest number contained in $TC_TP10. o Bit3=1: Search the tool in the group with the lowest actual value of the monitored variable. o Bit4=1: Search the tool in the group with the highest actual value of the monitored variable. o Bit5: Reserved o Bit6=1: Search in the current magazine (only if bit7=1). o Bit7=0: Begins the search in the magazine where the last tool change took place.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			320000	
			Read access	

magWearCompoundNo (T, TM)	$TC_MAP9			
Active wear group The variable sets the stated wear group in the magazine [n] to active. The tool locations of the magazine must have previously been assigned to a wear group via variable $TC_MPP5. Example: $TC_MAP9[1] = 1st wear group 1 in magazine "1" is activated.				
Return value	o >0: Number of the wear group with which the tool search begins. o 0: No wear group active o <0: Number of the wear group with which the tool search begins. However, if this wear group is disabled, the system changes to the next possible wear group for the next tool search.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	

magWearCompoundNo (T, TM)	$TC_MAP9	
	$$numMagsMax	
	Read access and write access	

magPlaceSearchStrat (T, TM)	$TC_MAP10, Bits 8-15			
Empty location search strategy The variable returns the current empty location search strategy of the magazine. It corresponds to system variable $TC_MAP10 bit 8-16.				
Return value	Value range: o Bit 8: Forward search as from location number 1 in ascending order. o Bit 9: Forward search as from the current location at the change position in ascending order. o Bit 10: Backward search, backwards from the last location number. o Bit 11: Backward search, backwards from the current location of the change position. o Bit 12: Symmetrical search beginning at the current location number at the change position (1st location left, 1st location right, 2nd location left 2nd location right, etc.). o Bit 13: 1:1 replacement (only significant with tool change): If the location types and sizes of the old and new tools are the same, the magazine location of the "new" tool being loaded is transferred to the "old" tool being removed, and vice versa. The 1:1 replacement is preferably checked. If a 1:1 replacement is not possible, the other settings are used for the search strategy. o Bit 14=0: Search first in the individual magazines. If no possible location is found, search in the next magazine for an available location for the tool. o Bit 14=1: Search in all magazines for the best location corresponding to the hierarchy for the tool. o Bit15=0: (Conventional type of hierarchy): With this type, the location type of the searching tool is sought in the table of system variables $TC_MPTH. If the location type is found, this hierarchy is taken and evaluated from this level to the end. Bit 15=1: (Alternative type of hierarchy): Location type hierarchies can be defined for the recorded location types 1, ..., $MN_MM_MAX_NUM_OF_HIERARCHIES. The hierarchy for location type 1 is defined by $TC_MPTH[0,n], for location type 2 by $TC_MPTH[1,n], and so on. (n: Index in a hierarchy). With this setting, a location type can be defined in various hierarchies.			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			Read access	

3.2.7.25 TMC: Magazine data, configuration data

Each tool magazine is configured with several parameters during start-up. These configuration data together with the state information are combined in the module TMC.

magBLMag (T, TMC)				W4
Number of the loading magazine The variable returns the number of the loading magazine in this TO unit.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

magCBCmd (T, TMC)				W4
Gives type of current PI service in the TO unit. The variable has the code "1" for the empty location search with the PI services: These are the PI services: o _N_TMFDPL o _N_TMFPBP o _N_FDPLMT With the PI services for moving a tool between locations of internal magazines, this variable contains the code for the type of PI service. If the movement comes from a real magazine or leads to a real magazine, the type of PI service is entered in "magCmd" in the T/TM module. "magCmdState" can be used to determine whether the commands have been completed. Access to the status and results of the PI services can be controlled by semaphores.				
Return value	Value range: o 1 = _N_TMFPPL, _N_TMFDPL and _N_FDPLMT (empty location search) o 2 = _N_TMMVTL (move tool) o 3 = _N_TMPOSM (position tool) o 4 = _N_POSMT (position multitool)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

magCBCmdState (T, TMC)				W4
Gives status of current PI service in the TO unit The processing state of the PI service can be queried with this variable, which is stored in "magCMCmd". Only the status is determined for PI service _N_POSMT_. No parameters (magCMCmdPar1...) are made available.				
Return value	Values: o =0: Initial state (after the control has run up). o =1: The PI service has been accepted but not yet output to the PLC. o =2: The command has been output to the PLC, and the NC waits for (end) acknowledgment from the PLC. o =3: The PI service was terminated successfully. The tool has reached its destination location. o =4: The PI service was terminated with an error. This also occurs if the parameterization of the PI service is incorrect or the command with the acknowledgment status=3 (without alarm) was cancelled.			

magCBCmdState (T, TMC)				W4
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

magCBIdent (T, TMC)		$TC_MAMP1		W4
Name of the magazine configuration				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Validity:	From NCK version 3.6		Read access	

magCMCmdPar1 (T, TMC)				W4
Return parameter 1 to MagCBCmd If there is movement between locations of internal magazines, the variable contains the magazine number of the target. In the event of an error, the variable contains the wrong magazine number.				
Return value	Value range: o Magazine number o Error number			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

magCMCmdPar2 (T, TMC)				W4
Return parameter 2 to MagCBCmd If there is movement between locations of internal magazines, the variable contains the magazine number of the target.				
Return value	Value range: o Location number o Error number			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

magRPlaces (T, TMC)				W4
Total number of real magazine locations The variable returns the number of all locations of all magazines in the TO unit. The locations in buffers and loading locations are counted.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

magSearch (T, TMC)	$TC_MAMP2	W4
Type of tool search and empty location search Type of tool search (bit 0...7) and type of empty location search (bit 8...16) Note: Individual settings in "magSearch" have priority in the T/TM module.		

magSearch (T, TMC)	$TC_MAMP2	W4
Return value	Tool search: o Bit0=0: (Default strategy) Take the first available tool found in the tool group. Search first in the magazine from which the last change was made. o Bit0=1: Select the "active" tool in the magazine of the previously changed tool, otherwise find the replacement tool with the lowest duplo number. If no tool is found in this magazine, the search is continued in the other linked magazines. o Bit1: Find the next replacement tool that is closest to the current magazine position. o Bit2: Select the "active" tool, otherwise the replacement tool with the lowest number contained in $TC_TP10. o Bit3: Find the tool in the group with the lowest actual value of the monitored variable. o Bit4: Find the tool in the group with the highest actual value of the monitored variable. o Bit5: Reserved o Bit6: Search first in the currently considered magazine (effective only in conjunction with bit 7=1). o Bit7=0: Start the tool search in the magazine from which the last changed tool came. o Bit7=1: Always start the search in the 1st magazine in the distance table. Note: Bit7=1 + bit0=1 or bit2=1, if no "active tool" is found in the magazine, the active tool - if present - is selected from one of the other magazines linked to the tool holder. Empty location search: Bit8: Search forwards. Search in ascending order from location no. 1. Bit9: Search forwards. Search in ascending order from the current location at the change position. Bit10: Search backwards. Search backwards from the last location no. Bit11: Search backwards. Search backwards from the current location at the change position. Bit12: Symmetrical search. The search starts at the current location no. at the change position (1st location left, 1st location right, 2nd location left, 2nd location right and so on). Bit13: 1:1 replacement (only significant with a tool change): If the old and new tools have the same location type and size, the magazine location of the "new" tool to be loaded is transferred to the "old" tool to be unloaded and vice versa. The 1:1 replacement is checked first. If the 1:1 replacement is not possible, the other settings become effective for the search strategy. Bit14=0: Search first in the individual magazines. If no possible location is found, search for a free location for the tool in the next magazine. Bit14=1: Search in all magazines for the best location corresponding to the hierarchy for the tool. Bit15=0: (Conventional type of hierarchy): With this type, the location type of the sought tool is sought in the table of system variables $TC_MPTH. If the location type is found, this hierarchy is accepted, and evaluated from this level to the end. Bit15=1: (Alternative type of hierarchy): Location type hierarchies can be defined for the selected location types 1, ..., $MN_MM_MAX_NUM_OF_HIERARCHIES. The hierarchy for location type 1 is defined by $TC_MPTH[0,n], that for location type 2 by $TC_MPTH[1,n], and so on. (n: index within a hierarchy). With this setting, a single location type can be defined in different hierarchies.	

magSearch (T, TMC)		$TC_MAMP2		W4
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

magVPlaces (T, TMC)				W4
Number of real locations The variable returns the number of real locations in all real magazines in the TO unit. The locations in buffers and loading locations are not counted.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

magZWMag (T, TMC)				W4
Number of the buffer magazine The buffers are the second internal magazine. Number of the buffer magazine in this TO unit.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

modeWearGroup (T, TMC)	$TC_MAMP3			
Strategy wear group Handling of the tools in a wear group (bit 0...7) Search strategies for wear groups (bit 8...15)				
Return value	o Bit 0=0: When a wear group is activated, the status of the tools remains unchanged o Bit 0=1: When a wear group is activated, the status of the tools changes. A tool is activated in each tool group. o Bit 1=0: When a wear group is disabled, the status of the tools remains unchanged o Bit 1=1: When a wear group is disabled, the tools are set to "disabled". o Bit 2...7 Reserved Search strategy for the next wear group o Bit 8=0: Find the next possible wear group o Bit 8=1: Find the wear group with the next higher, activatable group number o Bit 9...11 Reserved Search strategy within the tool group for the tool to be set to active (language command SETTA or PI_SETTST) o Bit 12=0: Lowest possible duplo number o Bit 12=1: Lowest possible magazine location number o Bit 13=1: Lowest number contained in $TC_TP10 (replacement order)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
			Read access	

magConfToolSearchStrat (T, TMC)	$TC_MAMP2			W4
Type of tool search Type of tool search. The variable has a bit-by-bit meaning. Bits 0-7, correspond to the variable "magSearch".				
Return value	A set bit has the following meaning: Bit0=0: (Default strategy) Take the first available tool found in the tool group. Search first in the magazine from which the last change was made. Bit0=1: Select the "active" tool in the magazine of the previously changed tool, otherwise find the replacement tool with the lowest duplo number. If no tool is found in this magazine, the search is continued in the other linked magazines. Bit1: Find the next replacement tool that is closest to the current magazine position. Bit2: Select the "active" tool, otherwise the replacement tool with the lowest number contained in $TC_TP10. Bit3: Find the tool in the group with the lowest actual value of the monitored variable. Bit4: Find the tool in the group with the highest actual value of the monitored variable. Bit5: Reserved Bit6: Search first in the currently considered magazine (effective only in conjunction with bit 7=1). Bit7=0: Start the tool search in the magazine from which the last changed tool came. Bit7=1: Always start the search in the 1st magazine in the distance table. Note: Bit7=1 + bit0=1 or bit2=1, if no "active tool" is found in the magazine, then - if present - the active tool is selected from one of the other magazines linked to the tool holder			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Validity:	From NCK version 3.6		Read access	

magConfMagSearchStrat (T, TMC)	$TC_MAMP2	W4
Type of empty location search Type of empty location search. The variable has a bit-by-bit meaning. In bits 8-15, it corresponds to the variable "magSearch".		
Return value	A set bit has the following meaning: Bit8: Search forwards. Search in ascending order from location no. 1. Bit9: Search forwards. Search in ascending order from the current location at the change position. Bit10: Search backwards. Search backwards from the last location no. Bit11: Search backwards. Search backwards from the current location at the change position. Bit12: Symmetrical search. The search starts at the current location no. at the change position (1st location left, 1st location right, 2nd location left, 2nd location right and so on). Bit13: 1:1 replacement (only with tool change of significance): If the old and new tools have the same location type and size, the magazine location of the "new" tool to be loaded is transferred to the "old" tool to be unloaded and vice versa. The 1:1 replacement is checked first. If the 1:1 replacement is not possible, the other settings become effective for the search strategy. Bit14=0: Search first in the individual magazines. If no possible location is found, search for a free location for the tool in the next magazine. Bit14=1: Search in all magazines for the best location for the tool corresponding to the hierarchy. Bit15=0: (Conventional type of hierarchy): With this type, the location type of the sought tool is sought in the table of system variables $TC_MPTH. If the location type is found, this hierarchy is accepted, and evaluated from this level to the end. Bit15=1: (Alternative type of hierarchy): Location type hierarchies can be defined for the selected location types 1, ..., $MN_MM_MAX_NUM_OF_HIERARCHIES. The hierarchy for location type 1 is defined by $TC_MPTH[0,n], that for location type 2 by $TC_MPTH[1,n], and so on. (n: index within a hierarchy). With this setting, one location type can be defined in different hierarchies. Bit16: The hierarchy consideration is canceled for the mini hierarchy that consists merely of the location type itself and location type 0. In this case, the empty location search does not distinguish between the suitable location type ($TC_TP7 == $TC_MPP2) and the general location type "0" of the magazine location.	

Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UDWORD			
Validity:	From NCK version 3.6		Read access	

3.2.7.26 TMV: Magazine data, directory

The data module TMV can be used for the following purposes:

1. To display all magazines. The most important magazine information is combined in the module TMV. The existing magazines are sorted in ascending order according to the magazine number without gaps. This means that variables that are defined in this module as one-dimensional arrays contain all magazine information without any gaps. The row index with which a specific array can be addressed does not refer to the magazine number, it is merely a serial number. Inserting/deleting a magazine dynamically changes the contents of a row.

2. To access magazine data in the modules TM, TP and TPM. Before accessing an element in the above modules, the module TV should be consulted to determine which tools have actually been defined.

magVIdent (T, TMV)				
Identifier of the magazine Identifier of the magazine at the consecutive number (= line index) in the TMV module list.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

magVNo (T, TMV)				
Number of the magazine Magazine number of the magazine at the consecutive number (= line index) in the TMV module list.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numMagsMax	
Validity:	From NCK version 3.6		Read access	

numActMags (T, TMV)				
Number of magazines Number of magazines in the TMV and TM modules Only returns values >0 if MD bit $MN_MM_TOOL_MANAGEMENT_MASK, bit0=1 (TMMG) and $MN_MM_NUM_MAGAZINE > 0 are set.				
Units and value ranges				
Physical unit:	Data type:			Maximum value:
-	TYPE_UWORD			$$numMagsMax
Validity:	From NCK version 3.6		Read access	

numActAdapters (T, TMV)		$P_MAGNA, $P_MAGNA1, $P_MAGNA2		
Number of adapters created Number of adapters created in the ADN module of the TO unit Line 1: Total number Line 2: Number of type 1 adapters Line 3: Number of type 2 adapters				
Units and value ranges				
Physical unit:	Data type:	Initial value	Minimum value:	Maximum value:
-	TYPE_UWORD	0	0	$$maxNumAdapter
Addressing				
Line index:			Max. line index:	
			3	
			Read access	

adaptVNo (T, TMV)		$P_MAGA		
Number of the adapter Number of the i-th adapter				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	32000
Addressing				
Line index:			Max. line index:	
			$$maxNumAdapter	
			Read access	

adaptVType (T, TMV)		$TC_ADPT_TYPE[$P_MAGA[i]]		
Adapter type Adapter type of the i-th adapter 1: Type 1 adapter defined. 2: Type 2 adapter defined.				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		1	2
Addressing				
Line index:			Max. line index:	
			$$maxNumAdapter	
			Read access	

3.2.7.27 TO: Tool offsets

The data module TO is organized as a 2-dimensional variable array.

The module contains the tool edge offset data for all tools. Each element can be addressed via a column and row index:

The column index is the tool number (T-number), i.e. the offset data for all cutting edges of a tool are located in one column. The assignment of a tool to a T-number is given in the module "Tool directory" (TV) in the related area T. If a non-existent tool number is entered for the column index the request is negatively acknowledged.

The number of rows is derived from the number of parameters per tool edge and the number of edges on a tool:

maxZeilenindex = $$numCuttEdgeParams * $$/T/TV/numCuttEdges (T-number)

The number of parameters per tool edge "$$numCuttEdgeParams" is given in module Y in area N. The number of cutting edges "$$/T/TV/numCuttEdges" is always tool-specific and is given in the module TV in associated area T.

If necessary, several rows can be addressed, so that in one request, for example, all tool edge offset data of a single tool can be read. The offset values of the tool edges are all of the same data type and have the same physical unit.

cuttEdgeParam (T, TO)		**$TC_DPCEx[y,z] x = ParamNo y = ToolNo z = EdgeNo**		
Replaced by $$edgeData				
Return value	The value for the tool type is stored internally as an integer.			
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			($$numCuttEdgeParams + 1) * $$maxnumCuttEdges_Tool	
			Read access and write access	

edgeData (T, TO)	$TC_DPx[y,z] x = ParamNo y = ToolNo z = EdgeNo	

Cutting edge offset data

The cutting edge parameters have 2 parts:

The first part runs to line $$numCuttEdgeParams + $$maxnumCuttEdges_Tool and contains the actual cutting edge parameters.

Cutting edge offset data:

o x = 1: Tool type (INT)

o x = 2: Cutting edge position

o x = 3: Geometry length 1

o x = 4: Geometry length 2

o x = 5: Geometry length 3

o x = 6: Geometry radius

o x = 7: Geometry corner radius (tool type 700)

o x = 8: Geometry length 4 (tool type 700)

o x = 9: Geometry length 5

o x = 10: Geometry angle 1

o x = 11: Geometry angle 2 (tapered milling tools)

o x = 12: Wear length 1

o x = 13: Wear length 2

o x = 14: Wear length 3

o x = 15: Wear radius

o x = 16: Wear groove width b

o x = 17: Wear proj. length k

o x = 18: Wear length 5

o x = 19: Wear angle 1

o x = 20: Wear angle 2 for tapered

milling tools

o x = 21: Adapter length 1

o x = 22: Adapter length 2

o x = 23: Adapter length 3

o x = 24: Clearance angle

o x = 25: Manual Turn: Cutting speed

ShopMill: Status word of tool

o x = 26: H number, with ISO mode

o x = 27: Tool edge orientation

o x = 28: L1 component of the tool edge orientation

o x = 29: L2 component of the tool edge orientation

o x = 30: L3 component of the tool edge orientation

o x = 31: Standardized L1 component of the tool edge orientation

o x = 32: Standardized L2 component of the tool edge orientation

o x = 33: Standardized L3 component of the tool edge orientation

o x = 34: Number of teeth of a cutting edge

o x = 35: Basic angle of rotation of the cutting edge

edgeData (T, TO)		$TC_DPx[y,z] x = ParamNo y = ToolNo z = EdgeNo		
The second part starts at line ($$numCuttEdgeParams * $$maxnumCuttEdges_Tool) + 1 and ends at ($$numCuttEdgeParams + 1) * $$maxnumCuttEdges_Tool. It contains the D number of the cutting edge number(CE-No) 1 - $$maxnumCuttEdges_Tool. this is required for the function "Unique D number".				
Return value	Value range: o =-1: Cutting edge not present o = 1 .. maxDNo: cutting edge present o = 0: No D number assigned/assignment cancelled. $TC_DPCE = Cutting edge number, D = offset number D. If the D number of a cutting edge (variable of the TO module) is set to invalid, this does not affect value $TC_DPCE. The cutting edge number in the description of the line index is the same as that of parameter $TC_DPCE. The variable D number. defined in the module corresponds to the second index in the offset-specific parameters of type $TC_DPx[T,D],... and others with x=1,....35.) Notice: This variable is named "cuttEdgeParam" in non-Windows HMI and PLC.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			($$numCuttEdgeParams + 1) * $$maxnumCuttEdges_Tool	
Validity:	From NCK version 3.6		Read access and write access	

3.2.7.28 TOE: Edge-related location-dependent coarse total offsets, setup offsets

One set of edge-related coarse total offsets, setup offsets, exists for each tool edge and operating location. This module corresponds totally to module T / TOS, edge-related location-dependent fine total offsets.

edgeECData (T, TOE)	$TC_ECPx[t,d]			
Location-dependent setup offsets "coarse" Location-dependent (coarse) setup offsets are entered before machining, and taken into account in the total offset (fine). The location-dependent offsets relate to the geometric data of a cutting edge. DL means D location, in which location means the reference to the operating location. The function is activated by machine data MD18108 $MN_MM_NUM_SUMCORR, MD 10110 $MN_MM_MAX_SUM-CORR_PER_CUTTEDGE, MD 18112 $MN_MM_KIND_OF_SUMCORR, bit4 and MD 18080 $MN_MM_TOOL_MANAGE-MENT_MASK, bit8. The individual setup offset parameter sets are differentiated by the number ranges of system variable $TC_ECP: o x = 13-21, activatable with DL=1 o x = 23-31, activatable with DL=2 o x = 33-41, activatable with DL=3 o x = 43-51, activatable with DL=4 o x = 53-61, activatable with DL=5 o x = 63-71, activatable with DL=6 The setup offsets are activated in addition to the total offsets if MD $MN_MM_KIND_OF_SUMCORR, bit4=1 is set.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numParams_SC * $$maxnumEdgeSC * $ $maxnumCuttEdges_Tool	
			Read access and write access	

3.2.7.29 TOET: Edge-related location-dependent coarse total offsets, transformed setup offsets

One set of edge-related transformed total offsets exists for each tool edge and operating location.
This module corresponds totally to module T / TOE.

edgeECData (T, TOET)				
Transformed location-dep. setup offsets "coarse" Transformed, location-dependent setup offsets (coarse) are entered before machining, and taken into account in the total offset (fine). The location-dependent offsets relate to the geometric data of a cutting edge. DL means D location, in which location means the reference to the operating location. The function is activated by machine data MD18108 $MN_MM_NUM_SUMCORR, MD 10110 $MN_MM_MAX_SUMCORR_PER_CUTTEDGE, MD 18112 $MN_MM_KIND_OF_SUMCORR, bit4 and MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK, bit8. The individual setup offset parameter sets are differentiated by the number ranges of system variable $TC_ECP: o x = 13-21, activatable with DL=1 o x = 23-31, activatable with DL=2 o x = 33-41, activatable with DL=3 o x = 43-51, activatable with DL=4 o x = 53-61, activatable with DL=5 o x = 63-71, activatable with DL=6 The setup offsets are activated in addition to the total offsets if MD $MN_MM_KIND_OF_SUMCORR, bit4=1 is set.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numParams_SC * $$maxnumEdgeSC * $$maxnumCuttEdges_Tool	
			Read access and write access	

3.2.7.30 TOS: Edge-related positional fine total offsets

One set of edge-related total offsets exists for each tool edge and operating location.

The maximum number of operating locations is identical for all tool edges and defined by the new variable $ $maxnumEdgeSC ($MN_MAX_SUMCORR_PERCUTTING_EDGE) in "N / Y global system data".

$$numParams_SC (currently 9) offsets are provided (depending on location-independent wear values) for each total offset set: Length 1, length 2, length 3, radius and 5 others.

Each replacement tool has its own separate (different) data.

The NCK resets the data when the associated tool is activated if machine data ($MN_MM_KIND_OF_SUM-CORR, bit 1 = 1) is used for activation.

The total offsets of a tool edge are accessible via the internal T number of the associated tool, edge number, total offset number ("operating location").

PI Services may exist for selective creation and deletion of tool edge total offsets.

The existence of total offsets can be controlled selectively via the new machine data $MN_MM_NUM_SUM-CORR (OPI: $$maxNumSumCorr in N / Y).

The following applies:

When the MMC2 tool management function is in use, $MN_MM_NUM_SUMCORR = -1 must be set to ensure that the total offsets exist for all offset locations (number = $$maxnumEdgeSC) from creation of the tool edge until its deletion.

(The new PI Services for creation / deletion will not currently be used by the MMC2 tool management for turning applications). For the present, the new NC machine data $MN_MM_NUM_SUMCORR = -1 must be set to automatic creation / deletion.

The method of addressing in this module is analogous to accessing "Edge data / offsets" by column addressing with T number (using an array access operation to gain quick access to the total offsets of all tool edge operating locations or all edges of a tool).

The module contains the location-dependent total offsets for all tools. Each element is addressed via a column and line index:

The column index is the tool number (T number), i.e. all location-dependent total offsets of this tool (for all edges / locations) can be found in one column.

If a non-existent T number is specified as the column index, the request is acknowledged negatively.

The number of lines is determined by the number of total offset values, the number of operating locations and the maximum possible edge number of a tool:

maxZeilenindex = $$numParams_SC * $$maxnumEdgeSC * $$maxnumCuttEdges_Tool

These variables are stored in "N / Y global system data" and have the following meanings:

$$numParams_SC: No. of wear offsets per location (according to L1, L2, L3, radius and 5 others), currently 9

$$maxnumEdgeSC: Maximum number of locations (SC) per edge
$$maxnumCuttEdges_Tool: Max. permissible number of edges per tool

Several lines can be addressed simultaneously if necessary, allowing, for example, all location-dependent total offsets of all edges of one tool to be read in one request. The location-dependent total offsets of the tools are all of the same data type and have the same physical unit.

Module T / TOS has a 2-dimensional organization.

The following lines are provided for each T number (column index):

Edge 1, Location 1, L1
Edge 1, Location 1, L2
Edge 1, Location 1, L3
Edge 1, Location 1, Radius
Edge 1, Location 1, Par5
....
Edge 1, Location 1, Par $$numParams_SC
Edge 1, Location 2, L1
Edge 1, Location 2, L2
Edge 1, ...
Edge 1, Location $$maxnumEdgeSC, Par $$numParams_SC

Edge 2, Location 1, L1
....
Edge 2, Location $$maxnumEdgeSC, Par $$numParams_SC
....
Edge $$maxnumCuttEdges_Tool, Location $$maxnumEdgeSC, Par $$numParams_SC

Interrelationship between edge parameters, total offsets and variables:

Edge parameter	DL1	DL2	...	DL4	...
$TC_DP3	$TC_SCP13	$TC_SCP23	...	$TC_SCP43	...
$TC_DP4	$TC_SCP14	$TC_SCP24	...	$TC_SCP44	...
$TC_DP5	$TC_SCP15	$TC_SCP25	...	$TC_SCP45	...
....					
$TC_DP9	$TC_SCP19	$TC_SCP29	...	$TC_SCP49	...
$TC_DP10	$TC_SCP20	$TC_SCP30	...	$TC_SCP50	...
$TC_DP11	$TC_SCP21	$TC_SCP31	...	$TC_SCP51	...

with DLx, TC_DPy, TC_SCPz

x from 1 to 6 ($$maxnumEdgeSC = $MN_MAX_SUMCORR_PERCUTTING_EDGE) and maximum = 6
y from 3 to 11
z = (10 * x) + y

edgeSCData (T, TOS)		$TC_SCPx[t,d]		
Total offset - fine Total offsets (fine) - location-dependent offsets comprise the error values that make up the deviation of the workpiece from the setpoint dimension. The parameters of the location-dependent offsets relate to the geometric data of a cutting edge. DL means D location, in which location means the reference to the operating location. The function is activated by machine data MD18108 $MN_MM_NUM_SUMCORR, MD 10110 $MN_MM_MAX_SUM-CORR_PER_CUTTEDGE and MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK, bit8. The individual total offset parameter sets are differentiated by the number ranges of system variable $TC_SCP: o x = 13-21, activatable with DL=1 o x = 23-31, activatable with DL=2 o x = 33-41, activatable with DL=3 o x = 43-51, activatable with DL=4 o x = 53-61, activatable with DL=5 o x = 63-71, activatable with DL=6				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numParams_SC * $$maxnumEdgeSC * $ $maxnumCuttEdges_Tool	
			Read access and write access	

3.2.7.31 TOST: Edge-related location-dependent total offsets, transformed

One set of edge-related transformed total offsets exists for each tool edge and operating location.
This module corresponds totally to module T / TOS.

edgeSCData (T, TOST)				
Transformed total offset - fine Transformed total offsets (fine) - location-dependent offsets comprise the error values that make up the deviation of the workpiece from the setpoint dimension. The parameters of the location-dependent offsets relate to the geometric data of a cutting edge. DL means D location, in which location means the reference to the operating location. The function is activated by machine data MD18108 $MN_MM_NUM_SUMCORR, MD 10110 $MN_MM_MAX_SUM-CORR_PER_CUTTEDGE and MD 18080 $MN_MM_TOOL_MANAGEMENT_MASK, bit8. The individual total offset parameter sets are differentiated by the number ranges of system variable $TC_SCP: o x = 13-21, activatable with DL=1 o x = 23-31, activatable with DL=2 o x = 33-41, activatable with DL=3 o x = 43-51, activatable with DL=4 o x = 53-61, activatable with DL=5 o x = 63-71, activatable with DL=6				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			$$numParams_SC * $$maxnumEdgeSC * $ $maxnumCuttEdges_Tool	
			Read access and write access	

3.2.7.32 TOT: Edge data, transformed offset data

The HMI must be capable of displaying and modifying the offset data of the tool edges as both transformed and untransformed data. The transformation refers to the adapter data (if programmed) of magazine locations. The HMI can display and modify both transformed and untransformed data (of the same tool if necessary) "simultaneously" (in different applications or different HMIs).

To provide access to transformed data, a new module, T / TOT (edge data: transformed offset data), is provided which is identical to the existing module T / TO (edge data: Offset data), except that it supplies transformed data instead of untransformed data.

The information edge DNo (D numbers assigned to edges) is included under the offset ($$numCuttEdgeParams * $$maxnumCuttEdges_Tool) in both the T / TOT and T / TO modules.

Both modules have a 2-dimensional organization.

The T number is the column index.

Line numbers are calculated by the following method:

(EdgeNo -1) * $$numCuttEdgeParams +parameter No.

$$numCuttEdgeParams = parameter per edge (currently 25) (from Y in N area)

EdgeNo = edge number for tool

Example: with $$numCuttEdgeParams = 25 and $$maxnumCuttEdges_Tool = 9

Column: T number

Lines:

1 edge 1, parameter 1
2 edge 1, parameter 2
...
25 edge 1, parameter numCuttEdgeParams
26 edge 2, parameter 1
27 edge 2, parameter 2
...
50 edge 2, parameter numCuttEdgeParams
...
225 edge maxnumCuttEdges_Tool, parameter numCuttEdgeParams
226 edge 1, D No assigned to edge 1

Untransformed data: /Tool/Compensation/edgeData[uToa,cTNr,line_from,line_to]

Transformed data: /Tool/CompTransfor/edgeData[uToa,cTNr,line_from,line_to]

Values which can be displayed as transformed data are the 9 geo-data (corresponding to L1, L2, L3, radius, and generally 5 other values), wear and total offsets.

If tools which are not located in a magazine location with adapter data are accessed via the module for transformed data, then the data are treated as if they were untransformed.

cuttEdgeParam (T, TOT)				
Replaced by $$edgeData				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			($$numCuttEdgeParams + 1) * $$maxnumCuttEdges_Tool	
			Read access and write access	

edgeData (T, TOT)		
Transformed edge offset data Offset value parameters and cutting edge list with D numbers for a tool Part 1: Offset value parameters for a tool edge: Definition of the line index: (EdgeNo - 1) * $$numCuttEdgeParams + ParameterNo The meaning of each parameter depends on the corresponding tool type. At present, 35 parameters are reserved for each tool edge (however only some are allocated values). The valid parameters, some of which are only optional, can be found in OPI variable "$$extraCuttEdgeParams". To maintain flexibility for future extensions, the variable value '$$numCuttEdge-Params' should be used for calculation rather than the fixed value of 35 parameters. A detailed description of the tool parameters can be found in the 'Tool edge' chapter of the 'Tool Offset (W1)' documentation. The following list is a brief summary of the tool edge parameters: Parameter 1: Tool type ($TC_DP1) Parameter 2: Cutting edge position ($TC_DP2) Parameter 3: Geometry length 1 ($TC_DP3) Parameter 4: Geometry length 2 ($TC_DP4) Parameter 5: Geometry length 3 ($TC_DP5) Parameter 6: Geometry radius ($TC_DP6) Parameter 7: Geometry - corner radius (tool type 700; grooving saw) ($TC_DP7) Parameter 8: Geometry length 4 (tool type 700; grooving saw) ($TC_DP8) Parameter 9: Geometry length 5 ($TC_DP9) Parameter 10: Geometry angle 1 ($TC_DP10) Parameter 11: Geometry angle 2 (tapered milling tools ($TC_DP11) Parameter 12: Wear length 1 ($TC_DP12) Parameter 13: Wear length 2 ($TC_DP13) Parameter 14: Wear length 3 ($TC_DP14) Parameter 15: Wear radius ($TC_DP15) Parameter 16: Wear groove width b / rounding radius ($TC_DP16) Parameter 17: Wear projecting length k ($TC_DP17) Parameter 18: Wear length 5 ($TC_DP18) Parameter 19: Wear angle 1 ($TC_DP19) Parameter 20: Wear angle 2 for tapered milling tools ($TC_DP20) Parameter 21: Adapter length 1 ($TC_DP21) Parameter 22: Adapter length 2 ($TC_DP22) Parameter 23: Adapter length 3 ($TC_DP23) Parameter 24: Tool clearance angle ($TC_DP24) Parameter 25: ManualTurn: cutting speed ($TC_DP25) ShopMill: Bit-coded value for different states of tools of types 1xx and 2xx ($TC_DP25) Parameter 26: H number, with ISO mode Parameter 27: Tool edge orientation Parameter 28: L1 component of the tool edge orientation Parameter 29: L2 component of the tool edge orientation Parameter 30: L3 component of the tool edge orientation Parameter 31: Scaled L1 component of the tool edge orientation Parameter 32: Scaled L2 component of the tool edge orientation Parameter 33: Scaled L3 component of the tool edge orientation		

edgeData (T, TOT)				
Parameter 34: Number of teeth of the cutting edge Parameter 35: Basic angle of rotation of the cutting edges Part 2: edgeDNo, assigned optional D numbers of the cutting edges: Determination of the line index: (($$numCuttEdgeParams * $$maxnumCuttEdges_Tool) + EdgeNo)				
Return value	Value range: o =-1: Cutting edge not present o = 1 .. maxDNo: cutting edge present o = 0: No D number assigned/assignment cancelled. $TC_DPCE = Cutting edge number, D = offset number D. If the D number of a cutting edge (variable of the TO module) is set to invalid, this does not affect value $TC_DPCE. The cutting edge number in the description of the line index is the same as that of parameter $TC_DPCE. The variable D number. defined in the module corresponds to the second index in the offset-specific parameters of type $TC_DPx[T,D],... and others with x=1,....35.) Notice: This variable is named "cuttEdgeParam" in non-Windows HMI and PLC.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
mm, inch, user defined	TYPE_DOUBLE	0.0		
Addressing				
Line index:			Max. line index:	
			($$numCuttEdgeParams + 1) * $$maxnumCuttEdges_Tool	
			Read access and write access	

3.2.7.33 TP: Magazine location data

The data module TP is organized as a 2-dimensional variable array. The module contains the state and assignment of all magazine locations of a T area. Each element can be addressed by a column and line index:

The column index is the magazine number, i.e. the configuration data for all locations of a magazine are contained in one column. The assignment of a magazine to a magazine number is stated in the associated area T of the associated Magazine directory module (TMV). If a non-existent magazine number is specified for the column index, the request is negatively acknowledged.

The number of lines is derived from the number of parameters per magazine location and from the number of magazine locations:

maxLineindex = $$numMagPlaceParams * $$magNrPlaces

The number of parameters per magazine location "$$numMagPlaceParams" is given in area N of module Y.

The line indices are based on the following scheme:

1: Location type ($TC_MPP1) (read only)
  1: Magazine location
  2: Spindle
  3: Gripper
  4: Loader
  5: Transfer location
  6: Loading station
  7: Loading point
2: Location type ($TC_MPP2) (read only)
  >0: Location type for virtual location
  =0: "Match all" (buffer location)
  9999: Undefined (not a virtual location)
3: T number of tool in this location ($TC_MPP6)
4: Consideration of adjacent location on / off ($TC_MPP3)
  0: off
  1: on
5: Location status ($TC_MPP4)
  1: Disabled
  2: Free (<> occupied)
  4: Reserved for tool in buffer location
  8: Reserved for tool to be loaded
  16: Occupied in left half-location
  32: Occupied in right half-location
  64: Occupied in top half-location
  128: Occupied in bottom half-location
6: Physical magazine reference (read only)
  Magazine number of magazine to which location belongs
7: Type index ($TC_MPP5) (read only) and new: Wear group number as from SW 5.1

Type index/wear group number is read only in SW earlier than 5.1 but read/write as from SW 5.1 if it is assigned "Wear group" meaning.

Type index: The locations of a location type in a magazine are numbered in ascending order, e.g. type=2, type index=5; ==> Spindle5)

(previous meaning when location type = 1 before P5: Equals location number when location type=1)

Wear group number as from SW 5.1 ($TC_MPP5)

When location type = 1: Number of wear group to which this magazine location is assigned.

Value range: -32000, ..., -1, 0, 1, 2, ... 32000

=0: Not assigned to a wear group

>0: Number of assigned wear group, this wear group is enabled

<0: Number of assigned wear group, this wear group is disabled

By negating this system parameter, it is possible to disable or enable the whole assigned wear group.

See also $$magWearCompoundNo / $TC_MAP9 (active wear group number) and $$modeWearGroup / $TC_MAMP3 (general settings for wear grouping).

8: Adapter number as from SW 5.1 ($TC_MPP7)

Reference to adapter data set number.

Associated system data:

The number of parameters of this module changes accordingly:

N / Y, global system data, $$numMagPlaceParams = 8 as from SW 5.1

The number of magazine locations "$$magNrPlaces" is magazine specific and can be found in the associated area T of module TM.

The locations of the buffer magazine and the loading magazine are numbered in ascending order independently of the location type index.

If necessary, several lines can be addressed, so that, for example, all location data of a magazine can be read in a single request. The location data are all of the same type.

placeData (T, TP)	diverse, siehe Variablenbeschreibung	
Access to selected tool location data. The various parameters can be read out by selecting the lines: P1: Location type (read-only access) ($TC_MPP1) P2: Location type (read-only access) ($TC_MPP2) P3: T number of the tool at this location ($TC_MPP6) P4: Consider adjacent location On/Off ($TC_MPP3) P5: Location status (bit array) ($TC_MPP4) Description of the bit array, see value range (Disabled could mean the location must not be used or is possibly reserved for a fixed-location-coded tool) P6: Physical magazine reference (read-only access) P7: Location type index (numbering of a location type) ($TC_MPP5) P8: Number of the adapter at the magazine location ($TC_MPP7) P9: Tool number for which the location is reserved ($TC_MPP6) P10: Number of the spindle assigned to the buffer magazine location ($TC_MPP_SP) Only of significance if - Tool holders are used ($MC_TOOLHOLDER_MANAGEMENT > 0) - The magazine location "m" belongs to a buffer magazine "n" - The magazine location describes a tool holder ($TC_MPP1[n,m]=2) In this case the system variable contains the spindle number, the speed of which is to be monitored for maximum tool speed. If tool holders are not used ($MC_TOOLHOLDER_MANAGEMENT = 0), the variable contains the value of the spindle index from $TC_MPP5 If magazine location "n,m" is not a buffer magazine location for a spindle or tool holder, this variable contains the value = 0. P11: Type of T no. (tool or MT) ($P_TMNOIS)		

placeData (T, TP)		diverse, siehe Variablenbeschreibung		
Return value	Value range: o P1: Location type 1 = Magazine location 2 = Spindle 3 = Gripper 4 = Loader 5 = Transfer location 6 = Loading station 7 = Load position o P2: Location type >0 = Location type for virtual location = 0: "Match all" (buffer) 9999 = Undefined (no virtual location) o P3: T number o P4: Consider adjacent location 0 = Off 1 = On o P5 Location status 1 = Disabled 2 = Free (<> occupied) 4 = Reserved for tool in the buffer 8 = Reserved for tool to be loaded 16 = Occupies left half location 32 = Occupies right half location 64 = Occupies upper half location 128 = Occupies lower half location o P6 Magazine number of the magazine to which the location belongs (reference phys. magazine) o P7 Location type index buffer: 1 = Spindle Wear group number 0 = No wear group assigned >0 = Number of assigned wear group (enabled) <0 = Number of assigned wear group (disabled) o P8 Adapter number o P9 Magazine, T number of the tool, which has the identifier "reserved for tool in the buffer". o P10 Number of the spindle of the buffer magazine o P11 Type of T number (tool or MT)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

placeData (T, TP)		diverse, siehe Variablenbeschreibung	
		$$numMagPlaceParams * $$magNrPlaces	
Validity:	From NCK version 3.6	Read access and write access	

3.2.7.34 TPM: Magazine data, multiple assignment of place data

The data module TPM is organized as a 2-dimensional variable array.

ParameterNo = 1: Specifies the magazine number with which a relationship exists.

ParameterNo = 2: Distance (in locations) between the internal location and the magazine change position (cf. magazine number for 1st parameter) with which a relationship will be established.

It contains information about possible multiple assignments. The column index is the magazine number.

For location P with location number p in magazine MP (= column index) $$numPlaceMulti times the multiple assignments to other magazines which are possible are stored with the associated distances to the change positions in each of the magazines. The offset for row index zi for a location number p is calculated according to the following rule: zi = (p-1) * $$numPlaceMulti * $$numPlaceMultiParams + ParameterNo.

Determining the distance between the load position and the change position:

The value 9999 (magazine no. load position) must be specified for the variable $$multiPlace in the column. The LocationNo (p) for the line is the number of the load position. The line for the first assignment is calculated with ParameterNo = 1. When reading the variable, the system can thus read the magazine number linked to the intended change position. If this magazine number is correct, it is possible to read the number of locations between the load position and the change position with the variable $$multiPlace with the next higher line number. If the magazine number read was incorrect, the following magazine assignment must be read with the line number increased by $$numPlaceMulti.

This procedure has to be repeated a max. of $$numPlaceMultiParams times until the desired relationship has been found.

multiPlace (T, TPM)		diverse, siehe Variablenbeschreibung		
Distance to change position The current magazine position is required for tool change, loading and unloading. This position relates to the magazine zero point specified by the machine manufacturer. As a rule it is at the change position.				
Return value	o P1: Distance to the change position of magazine "n" to location "m" of the first internal magazine (9999). o P2: Distance to the change position of magazine n to the location of the 2nd internal magazine (loading magazine,9998)			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	

multiPlace (T, TPM)		diverse, siehe Variablenbeschreibung	
		$$numPlaceMulti * $$numPlaceMultiParams * $$magNrPlaces	
Validity:	From NCK version 3.6	Read access	

3.2.7.35 TS: Tool monitoring data

The module TS is organized as a 2-dimensional variable array. The module contains the tool edge monitoring data for all tools. Each element can be addressed via a column and line index:

The column index is the tool number (T-number), i.e. one column contains the monitoring data for all tool edges of a tool. The assignment of a tool to a T-number is given in the module Tool directory (TV) in the associated area T. If a non-existent tool number is specified for the column index, the request is acknowledged negatively.

The number of lines is derived from the number of parameters per tool edge and from the number of tool edges of a tool:

maxZeilenanzahl = $$numCuttEdgeParams_ts * $$/T/TV/numCuttEdges (T-number)

The number of parameters per tool edge "$$numCuttEdgeParams_ts" is given in the module Y in area N. The number of tool edges "$$/T/TV/numCuttEdges" is always tool specific and can be found in the module TV in associated area T.

If necessary, several lines can be addressed, so that in one request, for example, all tool edge monitoring data of a single tool can be read. The monitoring data of the tool edges are all of the same data type and have the same physical unit.

New tool monitoring modes "Monitoring of wear values" and "Monitoring of total offsets":

3 new parameters are provided for these modes:

P7 = Wear prewarning limit (SW 5.1 and later) ($TC_MOP6)

P8 = Remaining wear (actual value) (SW 5.1 and later) ($TC_MOP5)

P9 = Wear setpoint (SW 5.1 and later) ($TC_MOP15)

data (T, TS)		$TC_MOPx[y,z] x=ParamNo,y=T-Number,z=Edge		
Alternative name:	dummy			
Monitoring data of the tool edges Monitoring data of the tool edges Important: This is a 2-dimensional variable. The cutting edge number and the parameter number are addressed by the line index. 9 parameters are available for each tool edge. P1 = Prewarning limit service life in minutes ($TC_MOP1) P2 = Remaining service life in minutes ($TC_MOP2) P3 = Prewarning limit number of workpieces ($TC_MOP3) P4 = Remaining number of workpieces ($TC_MOP4) P5 = Desired service life ($TC_MOP11) P6 = Desired number of workpieces ($TC_MOP13) P7 = Prewarning limit wear (prewarning limit) ($TC_MOP5) P8 = Remaining wear (actual value) ($TC_MOP6) cannot be written P9 = Desired wear ($TC_MOP15) The data P1 - P6 are made available by the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit1=1 (TMMO). The data P7 - P9 are made available by the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit1=1 (TMMO) and bit5=1t.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_ts * $$maxnumCuttEdges_Tool	
Validity:	From NCK version 3.6		Read access and write access	

3.2.7.36 TT: Magazine location data

The module TT is organized as a 2-dimensional array where the variable with index (1/1) contains the maximum number of columns (corresponds to the location hierarchies) in this module. Each element can be addressed via a column and row index:

The column index is the number of the location hierarchy + 1. The row index is the number of the location type + 1. Row 1 contains the current T-number of rows for a specific location hierarchy as special information.

If all location types are to be read out for a location hierarchy, this must be defined in two steps:

1. The 1st line of each location hierarchy contains the number of assigned location types for this hierarchy
2. Lines 2 ... n can be read out in a single request.

placeType (T, TT)				
Magazine location hierarchy Search strategies for the empty location search within and between magazines can be derived from the hierarchization of location types. Different hierarchies can be defined. A prioritization of the magazine locations for the search can also be specified within a hierarchy.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			Wert aus Zeile 1	
Validity:	From NCK version 3.6		Read access	

3.2.7.37 TU: Tool data for OEM

(Previous designation: TUD)

The TUD data module is defined as a two-dimensional variable array. The module contains user-defined data for all tools. Each element can be addressed via a column and line index:

The column index is the number of the user-defined tool parameter. The number of tool parameters (columns) is given by the variable "$$numToolParams_tu" in module Y in area N.

The line index is the tool number. If non-existent tools are accessed, the request is acknowledged negatively.

The user-defined tool data are all of the same type.

data (T, TU)	$TC_TPCx[y] x = ParameterNo y = ToolNo			FBW
OEM-defined tool parameters OEM-defined parameter for the tools with all permissible data types, except the data type "String". The OEM data are made available with the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit2=1 and MD18094 $MN_MM_NUM_CC_TDA_PARAM>0. The data type of the individual parameters is specified by MD18095 $MN_MM_TYPE_CC_TDA_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE. If the data is the type "String", the variable can be read and written under "dataStr". If there is read access to an OEM-defined tool parameter of the data type "String", the value=0 is returned. If there is write access to an OEM-defined tool parameter of the data type "String", an error code is returned.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	

data (T, TU)		$TC_TPCx[y] x = ParameterNo y = ToolNo		FBW
			32000	
Validity:	From NCK version 3.6		Read access and write access	

dataStr (T, TU)		$TC_TPCx[y] x = ParameterNo y = ToolNo		
OEM-defined tool parameter (string type) OEM-defined tool parameters of type "String", column index = parameter number + 1000 If there is read access to an OEM-defined tool parameter that is not of the data type "String", an empty string is returned. If there is write access to an OEM-defined tool parameter that is not of the data type "String", an error code is returned.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			32000	
			Read access and write access	

3.2.7.38 TUE: Tool edge data for OEM

(Previous designation: TUO)

The data module TUE is organized as a two-dimensional variable field. The module contains user-defined edge data for all tools. Each element can be addressed via a column and line index:

The column index is the tool number (T number), i.e. the user-defined data for all edges of a tool can be found in one column. The assignment of a tool to a T number can be drawn from the module Tool Directory in te related area T. If a non-existant tool number is entered for the column index, then the job is acknowledged negatively.

The number of lines results from the number of parameters per edge and from the number of edges of a tool:

maxNumberof lines = $$numCuttEdgeParams_tu * $$/T/TV/numCuttEdges (T number)

The number of parameters per edge "$$numCuttEdgeParams_tu" can be drawn from module Y in area N. The number of edges "$$/T/TV/numCuttEdges" which can be tool-specific, can be drawn from module TV in the related area T.

If required, several lines may be addressed so that - for example - in one job all user-defined edge data of a tool can be read. The data are all of the same type.

edgeData (T, TUE)		$TC_DPCx[y,z] x=ParamNo,y=ToolNo z=EdgeNo		FBW
OEM data for tool edges OEM data for tool edges User-defined tool edge parameter, important: 2-dimensional variable, the column index is the T number. OEM-defined parameters for the cutting edges. They are made available by the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit2=1 and MD18096 $MN_MM_NUM_CC_TOA_PARAM>0. The data types of the individual parameters are specified by MD18097 $MN_MM_TYPE_CC_TOA_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_DOUBLE			
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_tu * $$maxnumCut-tEdges_Tool	
Validity:	From NCK version 3.6		Read access and write access	

3.2.7.39 TUM: Mgazine user data

This block contains the user magazine data

The TUM block is not intended for new developments.

userData (T, TUM)				
Magazine user data for a tool magazine Magazine user data for a tool magazine. These parameters can only be used if the machine data $MN_MM_NUM_CC_MAGAZINE_PARAM and $MN_MM_TOOL_MANAGEMENT_MASK have been set accordingly. This block exists only for compatibility reasons. New developments should be made with the T/TUMD block. (The data format there is "TYPE_DOUBLE")				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMagParams_u	
			Read access and write access	

3.2.7.40 TUMD: User magazine data double

This block contains the user magazine data (double)

userDataDouble (T, TUMD)		$TC_MAPCx[y] x = ParameterNo y = MagazineNo		
OEM-defined magazine parameters OEM-defined parameter for the magazines for all permissible data types, except the data type "String". The OEM data are made available with the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit0=1, bit2=1 and MD18090 $MN_MM_NUM_CC_MAGAZINE_PARAM>0. The data type of the individual parameters is specified by MD18091 $MN_MM_TYPE_CC_MAGAZINE_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE. If there is read or write access to an OEM-defined magazine parameter of the data type "String", an error string is returned.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMagParams_u	
			Read access and write access	

3.2.7.41 TUP: Magazine place user data

This block contains the user magazine location data

The TUP block is not intended for new developments.

userPlaceData (T, TUP)				
Magazine location user data for a tool magazine Magazine location user data for a tool magazine. These parameters can only be used if the machine data $MN_MM_NUM_CC_MAGLOC_PARAM and $MN_MM_TOOL_MANAGEMENT_MASK have been set accordingly. This block exists only for compatibility reasons. New developments should be made with the T/TUPD block. (The data format there is "TYPE_DOUBLE")				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DWORD	0		
Addressing				
Line index:			Max. line index:	
			$$numMagLocParams_u * $$magNrPlaces	
			Read access and write access	

3.2.7.42 TUPD: User magazine location data double

This block contains the user magazine location data (double)

userPlaceDataDouble (T, TUPD)		**$TC_MPPCx[y,z] x=ParamNo y=MagazineNo z=MagPlaceNo**		
Magazine place user data OEM-defined parameter for the magazine locations for all permissible data types, except the data type "String". The OEM data are made available with the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit0=1, bit2=1, and MD18092 $MN_MM_NUM_CC_MAGLOC_PARAM>0. The data type of the individual parameters is specified by MD18093 $MN_MM_TYPE_CC_MAGLOC_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE. If there is read or write access to an OEM-defined magazine location parameter of the data type "String", an error string is returned.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numMagLocParams_u * $$magNrPlaces	
			Read access and write access	

3.2.7.43 TUS: User monitoring data

This block contains the user monitoring data of the tool data

userData (T, TUS)		**$TC_MOPCx[y,z] x=ParamNo,y=T-Number,z=Edge**		
OEM parameters for cutting edge monitoring OEM-defined parameters for cutting edge monitoring. The data are made available by the MD settings MD bits $MN_MM_TOOL_MANAGEMENT_MASK, bit1=1 (TMMO) and bit2=1 and MD18098 $MN_MM_NUM_CC_MON_PARAM>0. The data type of the individual parameters is specified by the MD18099 $MN_MM_TYPE_CC_MON_PARAM, and can be determined from the file _N_COMPLETE_TOA_ACX. The transfer in the OPI is always as TYPE_DOUBLE.				
Units and value ranges				
Physical unit:	Data type:	Initial value		
-	TYPE_DOUBLE	0		
Addressing				
Line index:			Max. line index:	
			$$numCuttEdgeParams_tus * $$maxnumCuttEdges_Tool	
			Read access and write access	

3.2.7.44 TV: Tool data, directory

Data module TV can be used for the following purposes:

1. For displaying all tools of a magazine. The most important tool information is contained in module TV. Available tools are sorted consecutively in ascending order of T-number. This means that variables that are defined as one-dimensional arrays in this module contain all the tool information without any gaps. The line index with which a specific array is addressed has no connection with the tool number but is only a serial number. Inserting/deleting tools changes the contents of a line dynamically.

2. Access to tool data in modules TD, TG, TO, TS, TU and TUE. Before an element in one of the above modules is accessed, module TV should be consulted to ascertain which tools are actually defined.

SW 5.1 and later: Variable $$modeSpindleToolRevolver (module N/Y, global system data) defines for circular magazines (T / TM, magazine data, general data, MagKind=3) whether the tool in OPI modules "T / TP, magazine data, location data", "T / TD, tool data, general data", "T/TV, tool data, directory" and "T / AEV, working offsets, directory" remains (new functionality) in its circular magazine location during operation or changes to the buffer magazine (earlier behaviour).

Associated system data:

$$modeSpindleToolRevolver (module N / Y, global system data) with SW 5.1 and later.

TnumWZV (T, TV)				
Last assigned T-number for tool management Last assigned T-number for tool management The last assigned T number is the T number of the new tool last created in the NC through an NC language command or the PI service.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Validity:	From NCK version 3.6		Read access	

nrDuplo (T, TV)				
Duplo number The duplo number is automatically assigned when the tool is created. Its main purpose is to differentiate between tools with the same identifier (tool group). nrDuplo reads out the duplo number of the addressed tool.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numTools	
Validity:	From NCK version 3.6		Read access	

numCuttEdges (T, TV)		$P_TOOLND[tNo]		
Number of cutting edges of a tool Number of cutting edges of the tool at the consecutive number in the TV module list.				
Return value	o >0: Number of cutting edges of the tool. o 0: No cutting edge is defined.			
Units and value ranges				
Physical unit:	Data type:			Maximum value:
-	TYPE_UWORD			9
Addressing				
Line index:			Max. line index:	
			$$numTools	
Validity:	From NCK version 3.6		Read access	

numTools (T, TV)				
Number of tools in the area TO Contains the number of tools in the TO area				
Units and value ranges				
Physical unit:	Data type:		Minimum value:	Maximum value:
-	TYPE_UWORD		0	MD MM_NUM_TOOL
Validity:	From NCK version 3.6		Read access	

toolIdent (T, TV)		$TC_TP2[tNo]		
Tool identifier The tool identifier at the consecutive number in the TV module list.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_STRING			
Addressing				
Line index:			Max. line index:	
			$$numTools	
Validity:	From NCK version 3.6		Read access	

toolInMag (T, TV)		$A_TOOLMN		
Magazine in which the tool is currently located Magazine number of the magazine in which the tool is currently located, that is at the consecutive number in the TV module list.				
Return value	o >0: Magazine number of the tool or the magazine management is not active. o 0: Tool is not loaded			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			

toolInMag (T, TV)		$A_TOOLMN		
Addressing				
Line index:			Max. line index:	
			$$numTools	
Validity:	From NCK version 3.6		Read access	

toolInPlace (T, TV)		$A_TOOLMLN		
Current magazine location of the tool Magazine location number of the magazine containing the tool which is at the consecutive number in the TV module list.				
Return value	o >0: Magazine location number of the tool. o 0: Tool is not loaded or magazine management is not active (tool is not in any magazine).			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numTools	
Validity:	From NCK version 3.6		Read access	

toolNo (T, TV)				
T number of tool The internal tool number of the tool at the consecutive number in the TV module list. The internal number is assigned when a tool is created.				
Return value	Possible number range: 1-32000			
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			$$numTools	
Validity:	From NCK version 3.6		Read access	

numToolGroups (T, TV)				
Number of tool groups Tools with the same identifier but different D numbers are combined into a tool group. numToolGroups indicates the number of tool groups currently defined in the TOA unit.				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_UWORD			
Addressing				
Line index:			Max. line index:	
			Read access	

3.2.8 DriveVsa: Feed drive data

3.2.8.1 M: Machine data

V-M

r0020 (V, M)				
Speed setpoint smoothed The currently smoothed speed setpoint at the input of the speed controller resp. the U/f characteristic (after the interpolator)				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
			Read access	

r0021 (V, M)				
Actual speed smoothed Smoothed actual motor speed value				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
			Read access	

r0027 (V, M)				
Absolute actual current smoothed Smoothed actual current value				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
			Read access	

r0034 (V, M)				
Motor utilization thermal Thermal motor load				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
			Read access	

r0035 (V, M)				
Motor temperature Display and connector output for the actual temperature in the motor.				
Units and value ranges				
Physical unit:	Data type:			

r0035 (V, M)				
-	TYPE_FLOAT			
			Read access	

3.2.8.2 S: State data

During NC operation different internal states occur and system-specific data may change during operation. To distinguish those from system variables, they are classified as state data.

A distinction is made between:
- NCK-specific state data
- Mode-group-specific state data
- Channel-specific state data
- Drive-specific state data (FDD)
- Drive-specific state data (MSD)

No cyclic service may be set up on variables in this module. Only single variable access is permitted.

load611U (V, S)				
Load 611U				
Units and value ranges				
Physical unit:	Data type:			
-	TYPE_FLOAT			
Validity:	From NCK version 3.6		Read access	

Appendix A

A

A.1 List of abbreviations

Abbreviation	Derivation of the abbreviation	Meaning
ADI4	Analog Drive Interface for 4 Axis	
AC	Adaptive Control	
ALM	Active Line Module	Infeed module for drives
AP	User program	
AS	Automation system	
ASCII	American Standard Code for Information Interchange	American coding standard for the exchange of information
ASIC	Application Specific Integrated Circuit	User switching circuit
ASUP	Asynchronous subprogram	
AUTO		Operating mode "Automatic"
AUXFU	Auxiliary Function	Auxiliary functions
STL	Statement list	
BA	Operating mode	
Mode group	Mode group	
BERO	Proximity limit switch with feedback oscillator	
BI	Binector Input	
HHU	Handheld unit	
BICO	Binector Connector	Interconnection technology for the drive
BIN	Binary Files	Binary files
BIOS	Basic Input Output System	
BCS	Basic coordinate system	
BO	Binector Output	
OPI	Operator Panel Interface	
CAD	Computer-Auded Design	
CAM	Computer-Aided Manufacturing	
CC	Compile Cycle	Compile cycles
CI	Connector Input	
CF-Card	Compact Flash-Card	
CNC	Computerized Numerical Control	Computerized numerical control
CO	Connector Output	
COM Board	Communication Board	
CP	Communication Processor	
CPU	Central Processing Unit	Central processing unit
CR	Carriage Return	
CRC	Cyclic Redundancy Check	Checksum test
CRT	Cathode Ray Tube	Picture tube

Abbreviation	Derivation of the abbreviation	Meaning
CSB	Central Service Board	PLC module
CTS	Clear To Send	Signals that data is ready to be sent for serial data interfaces
CUTCOM	Cutter Radius Compensation	Tool radius compensation
DB	Data block	Data block in the PLC
DBB	Data-block byte	Data block-byte in the PLC
DBW	Data-block word	Data-block word in the PLC
DBX	Data-block bit	Data-block bit in the PLC
DDE	Dynamic Data Exchange	Dynamic data exchange
DDS	Drive Data Set	Drive data set
DIN	Deutsche Industrie Norm (German Industry Standard)	
DIR	Directory	Directory
DLL	Dynamic Link Library	
DO	Drive Object	Drive object
DPM	Dual Port Memory	
DRAM	Dynamic Random Access Memory	Dynamic memory block
DRF	Differential Resolver Function	Differential resolver function (handwheel)
DRIVE-CLiQ	Drive Component Link with IQ	
DRY	Dry Run	DRY run feedrate
DSB	Decoding Single Block	Decoding single block
DSC	Dynamic Servo Control / Dynamic Stiffness Control	
DSR	Data Send Ready	Signals the availability of serial data interfaces
DW	Data word	
DWORD	Double word (currently 32 bits)	
I	Input	
I/O	Input/Output	
ENC	Encoder	Actual value encoder
EPROM	Erasable Programmable Read Only Memory	Erasable, electronically programmable read-only memory
EQN		Type designation for an absolute encoder with 2048 sine signals per revolution
ESR	Extended stop and retract	
ETC	ETC key	Expansion of the softkey bar in the same menu
FB	Function block	
FBS	Flat screen	
FC	Function Call	Function block in the PLC
FDD	Feed Disable	Feed disable
FdStop	Feed Stop	Feed stop
FEPROM	Flash-EPROM	Read and write memory
FIFO	First In - First Out	Method of storing and retrieving data in a memory
FIPO	Fine interpolator	
FM	Function Module	

Abbreviation	Derivation of the abbreviation	Meaning
FM-NC	Function Module Numerical Control	Numerical control
FPU	Floating Point Unit	Floating-point unit
FRA	Frame block	
FRAME	Data set	Coordinate conversion with the components work offset, rotation, scaling, mirroring
CRC	Cutter radius compensation	
FST	Feed Stop	Feed stop
CSF	Control system flowchart (PLC programming method)	
FW	Firmware	
GC	Global Control	PROFIBUS: Broadcast telegram
GD	Global data	
GEO	Geometry, e.g. geometry axis	
GP	Basic program	
GS	Gear stage	
GUD	Global User Data	Global user data
HD	Hard Disk	Hard disk
HEX	Abbreviation for hexadecimal number	
AuxF	Auxiliary function	
HMI	Human Machine Interface	SINUMERIK user interface
MSD	Main spindle drive	
HT	Handheld Terminal	Handheld unit
HW	Hardware	
COMM	Commissioning	
IF	Drive module pulse enable	
IK (GD)	Implicit communication (global data)	
IKA	Interpolative Compensation	Interpolatory compensation
IM	Interface Modul	Interface module
INC	Increment	Increment
INI	Initializing Data	Initializing data
IGBT	Insulated Gate Bipolar Transistor	
IPO	Interpolator	
ISO	International Organization for Standardization	International Organization for Standardization
JOG	"Jogging" operating mode	
KD	Coordinate rotation	
KDV	Crosswise data comparison	Crosswise data comparison between the NC and PLC
K_V	Servo-gain factor	Gain factor of control loop
LAD	Ladder diagram	PLC programming method
LCD	Liquid Crystal Display	Liquid crystal display
LED	Light Emitting Diode	Light emitting diode
LF	Line Feed	
LMS		Position measuring system

Abbreviation	Derivation of the abbreviation	Meaning
LSB	Least Significant Bit	Least significant bit
LUD	Local User Data	User data
MAC	Media Access Control	
MAIN	Main program	Main program (OB1, PLC)
MB	Megabyte	
MCI	Motion Control Interface	
MCIS	Motion Control Information System	
MCP	Machine Control Panel	Machine control panel
MD	Machine data	
MDI	"Manual Data Automatic" operating mode	Manual input
MCS	Machine coordinate system	
MPF	Main Program File	Main program (NC part program)
MPI	Multi Point Interface	Multi-point interface
NC	Numerical Control	Numerical control
NCK	Numerical Control Kernel	Numerical control kernel
NCSD	NC Start Disable	NC start disable
NCU	Numerical Control Unit	NC hardware unit
IF	Interfaces	Interface signal
WO	Zero offset	
NX	Numerical Extension	Axis expansion module
OB	Organization block in the PLC	
OEM	Original Equipment Manufacturer	
OP	Operation Panel	Operator panel
OPI	Operation Panel Interface	Interface for connection to the operator panel
OSI	Open Systems Interconnection	Standard for computer communications
OPT	Options	Options
PIQ	Process Image Output	
PII	Process Image Input	
P bus	Peripheral bus	
PC	Personal Computer	
PCMCIA	Personal Computer Memory Card International Association	Standard for plug-in memory cards
PCU	Programmable Control Unit	
PI	Programm Instanz	
PG	Programming device	
PLC	Programmable Logic Control	Programmable Logic Controller
PN	PROFINET	
PO	POWER ON	
POU	Program organization unit	Unit in the PLC user program
PPU	Panel Processing Unit	Panel-based control
PTP	Point to Point	Point-to-point
PZD	Process data for drives	
QEC	Quadrant Error Compensation	Quadrant error compensation

Abbreviation	Derivation of the abbreviation	Meaning
QEC	Quadrant error compensation	
RAM	Random Access Memory	Program memory that can be read and written to
REF POINT		Function "Reference point approach" in JOG mode
REPOS		Function "Repositioning" in JOG mode
RID	Read In Disable	Read-in disable
RPA	R-Parameter Active	Memory area on the NC for R parameter numbers
RPY	Roll Pitch Yaw	Rotation type of a coordinate system
RTC	Real Time Clock	Real-time clock
RTS	Request To Send	RTS, control signal of serial data interfaces
SBL	Single Block	Single block
SBR	Subroutine	Subroutine (PLC)
SBT	Safe Brake Test	Safe Brake Test
SCC	Safety Control Channel	
SD	Setting-Datum	
SDB	System data block	
SDI	Safe Direction	Safe Motion Direction
SBT	Safe Brake Test	Safe Brake Control
SEA	Setting Data Active	Identifier (file type) for setting data
SERUPRO	Search–Run by Program Test	Search run by program test
SFC	System Function Call	
SGE	Safety-related input	
SGA	Safety-related output	
SH	Safe Stop	
SIC	Safety Info Channel	
SK	Softkey	
SKP	Skip	Skip block
SLM	Smart Line Module	
SLP	Safe Limited Position	Safely-Limited Position
SLS	Safely Limited Speed	Safely-Limited Speed
SM	Stepper Motor	
SOS	Safe Operating Stop	Safe Operating Stop
SS1	Safe Stop 1	Safe Stop 1 (time-monitored, ramp-monitored)
SS2	Safe Stop 2	Safe Stop 2
SPF	Subprogram file	Subprogram (NC)
SPL	Safe Programmable Logic	
PLC	Programmable Logic Controller	
SRAM	Static Random Access Memory	Static memory block
TNRC	Tool nose radius compensation	
LEC	Leadscrew error compensation	
SSI	Serial synchronous interface	Serial synchronous interface
STO	Safe Torque Off	Safe Torque Off
STW	Control word	
GWPS	Grinding wheel peripheral speed	

Abbreviation	Derivation of the abbreviation	Meaning
SW	Software	
SYF	System Files	System files
SYNACT	SYNACT Synchronized Action	Synchronized action
TB	Terminal Board (SINAMICS)	
TEA	Testing Data Aktive	Identifier for machine data
TCP	Tool Center Point	Tool tip
TCU	Thin Client Unit	
TEA	Testing Data Active	Identifier for machine data
TM	Terminal Module (SINAMICS)	
TO	Tool Offset	Tool offset
TOA	Tool Offset Active	Identifier (file type) for tool offsets
TRANSMIT	Transform Milling into Turning	Coordinate conversion on turning machines for milling operations
TTL	Transistor–Transistor–Logik	Interface type
UFR	User Frame	Zero offset
SR	Subroutine	
USB	Universal Serial Bus	
UPS	Uninterruptible Power Supply	
VDI		Internal communication interface between NC and PLC
FDD	Feed drive	
VPM	Voltage Protection Module	
VSM	Voltage Sensing Module	
WAB		Function "Smooth Approach and Retraction"
WCS	Workpiece coordinate system	
T	Tool coordinate system:	
TLC	Tool length compensation	
WPD	Work Piece Directory	Workpiece directory
T	Tool	
TM	Tool management	
TC	Tool change	
ZWS		Buffer location
ZOA	Zero Offset Active	Identifier (file type) for zero offset data
ZSW	Status word (of drive)	