

Title	MTX 15VRS Machine Parameters
Type of Documentation	Reference Book
Document Typecode	DOK-MTX***-MA*PAR**V15-RE05-EN-P
Internal File Reference	RS-7d76eec0a09eef1ac0a8028623337261-5-en-US-3
Change record	Edition 05, 2021-09 Refer to tab. 1-1 "Change record" on page 1
Copyright	© Bosch Rexroth AG 2021 All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Liability	The specified data is intended for product description purposes only and shall not be deemed to be a guaranteed characteristic unless expressly stipulated in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.
Editorial Department	Development Software and Control Firmware - Control Functions Motion and CNC, MaWe (TaDo/MePe)

Table of Contents

	Page
1 About this documentation.....	1
1.1 Validity of the documentation.....	1
1.2 Required and supplementing documentation MTX.....	2
1.2.1 Selection/compilation.....	2
1.2.2 Configuration.....	2
1.2.3 Commissioning.....	3
1.2.4 Operation.....	4
1.2.5 OEM Engineering.....	5
1.2.6 AddOns.....	5
1.3 Information representation.....	6
1.3.1 Using safety instructions.....	6
1.3.2 Names and abbreviations.....	8
1.3.3 Customer feedback.....	8
 2 Important instructions on use.....	 9
2.1 Intended use.....	9
2.1.1 Introduction.....	9
2.1.2 Areas of use and application.....	9
2.2 Unintended use.....	10
 3 NC configurator.....	 11
3.1 Basics.....	11
3.1.1 General information.....	11
3.1.2 NC Configurator function.....	11
3.1.3 Configuration data structure.....	11
3.1.4 Display sections of the NC Configurator.....	12
3.1.5 Icons and text colors used.....	17
3.2 How do I.....	18
3.2.1 Starting the NC Configurator.....	18
3.2.2 Closing the NC Configurator.....	18
3.2.3 Adapting the display areas according to the information requirements.....	19
3.2.4 Selecting the display areas.....	20
3.2.5 Selecting the data groups.....	21
3.2.6 Navigating in the display area "Edit parameters"	21
3.2.7 Adapting columns in the display area "Edit parameters"	23
3.2.8 Edit parameters.....	23
3.2.9 Undoing changes to parameters.....	24
3.2.10 Showing parameter attributes.....	24
3.2.11 Copying parameter objects.....	24
3.2.12 Filtering the display in the section "Edit parameters"	25
3.2.13 Finding parameter objects.....	27
3.2.14 Adding/removing channel and drive objects.....	28
3.2.15 Changing the drive functionality.....	29

Table of Contents

	Page
3.3	Importing/exporting configuration data..... 29
3.3.1	Exporting/importing the data group or the object node..... 29
3.3.2	Importing/exporting a data block..... 30
3.4	How to configure the NC..... 31
3.5	Troubleshooting MTX advanced NC configurator reports error 31 after change in parameterization..... 32
4	Explanation of parameters and terms used..... 33
4.1	Explanations..... 33
4.2	Terms..... 37
5	Optional parameter setup (SUP) - Setup of configuration data – 39
5.1	General information..... 39
5.2	Optional parameters..... 39
5.2.1	Intelligent Safety Technology (SAFE)..... 39
5.2.2	Electronic axis coupling (COUP)..... 40
5.2.3	Digitizing (DIG)..... 40
5.2.4	Measuring (MEAS)..... 41
5.2.5	Punching and nibbling (PUN)..... 41
5.2.6	Online corrections (ONL)..... 42
5.2.7	Move to positive stop (MPS)..... 42
5.2.8	Splines (SPL)..... 43
5.2.9	Tangential tool control (TTL)..... 43
5.2.10	Thread cutting (THRD)..... 43
5.2.11	Tool and workpiece management (TLWP)..... 44
5.2.12	Transformation (TRA)..... 44
5.2.13	Velocity-dependent output (VDO)..... 45
5.3	OEM-specific parameters..... 45
5.3.1	Prerequisites..... 45
5.3.2	OEM data group (OEM)..... 46
6	Basic NC configuration (MAIN) - Configuration data setup - 47
6.1	General information..... 47
6.2	Maximum number of channels..... 47
6.2.1	General information..... 47
6.2.2	Maximum number of channels (startup.xml)..... 47
6.3	Number of channels..... 48
6.3.1	General information..... 48
6.3.2	Number of channels..... 48
6.4	Maximum number of drives..... 48
6.4.1	General information..... 48
6.4.2	Maximum number of NC-controlled drives (startup.xml)..... 49
6.5	Number of drives..... 49
6.5.1	General information..... 49
6.5.2	Number of NC-controlled drives..... 50

	Page
6.5.3 Minimum number of configured drives.....	51
6.5.4 Number of configured drives.....	52
6.6 Maximum number of spindles.....	52
6.6.1 General information.....	52
6.6.2 Maximum number of NC-controlled spindles (startup.xml).....	52
6.7 Selection of drive function types.....	53
6.8 Drive parameterization.....	54
6.8.1 General information.....	54
6.9 NC-controlled drive (Dr[i]).....	55
6.9.1 Physical drive name.....	55
6.9.2 Enable drive.....	56
6.9.3 Enable Sercos communication.....	57
6.9.4 Enable virtual mode.....	57
6.9.5 Drive without command value input.....	58
6.9.6 Delayed reaction in case of non-fatal drive errors.....	58
6.10 Drive system functions (AxFunc).....	59
6.10.1 General information.....	59
6.10.2 Axis type.....	59
6.10.3 Channel assignment.....	60
6.11 Spindle system functions (SpFunc).....	61
6.11.1 General information.....	61
6.11.2 Spindle index.....	61
6.12 Sercos system functions (Scs).....	62
6.12.1 General information.....	62
6.12.2 Sercos address.....	62
6.12.3 Drive type.....	63
7 Channels (CHAN) - Basic parameters -	65
7.1 General information.....	65
7.2 Initialization of the channel status (Ini).....	65
7.2.1 State after startup (switched-on state).....	65
7.2.2 Status after channel reset.....	65
7.2.3 File name of the channel initialization program (init program).....	66
7.2.4 Delete SetPos Offset with a control reset.....	68
7.2.5 Automatic program reselection.....	68
7.3 Path data (Path).....	69
7.3.1 Velocity data (Vel).....	69
7.3.2 Feed override (Ovrd).....	70
7.3.3 Rapid traverse override (RapTravOvrd).....	72
7.3.4 Channel SHAPE (SHAPE).....	73
7.3.5 Acceleration/brake data (Acc).....	75
7.3.6 Control of the axis jump (Jump).....	76
7.3.7 Kinematic limitation (KinLim).....	77
7.4 Coordinates (Coord).....	79
7.4.1 Axis coordinate system (ACS).....	79
7.4.2 Machine coordinates (MCS).....	83

Table of Contents

	Page
7.4.3 Workpiece coordinates (WCS).....	84
7.5 Coupling (Couple).....	85
7.5.1 General information.....	85
7.5.2 Main spindle.....	85
7.5.3 Channel axis for G96.....	86
7.5.4 Coordinate system for G96 reference axis.....	86
8 Axes (AX) - Basic parameters -.....	89
8.1 General information.....	89
8.2 Limit switch monitoring (Pos).....	89
8.2.1 General information.....	89
8.2.2 Software limit switch (TrvLim[1..2]).....	90
8.3 Axis velocity data (Vel).....	92
8.3.1 Maximum axis velocity.....	92
8.3.2 Feed calculation.....	93
8.4 Override for asynchronous axes (Ovrd).....	94
8.4.1 Activating axis-specific override.....	94
8.5 Axis SHAPE (SHAPE).....	94
8.5.1 SHAPE order for axis.....	94
8.5.2 Minimum SHAPE order for axis.....	95
8.6 Feed forward (FeedFwd).....	96
8.6.1 Feed forward can be activated (via FFW/FFWA).....	96
8.7 Axis acceleration data (Acc).....	97
8.7.1 Maximum axis acceleration in feed mode.....	97
8.7.2 Maximum axis acceleration in rapid traverse.....	98
8.7.3 Axis acceleration reserve.....	98
8.8 Axis jerk data (Jerk).....	99
8.8.1 Maximum axis jerk in feed mode.....	99
8.8.2 Maximum axis jerk in rapid traverse.....	99
8.8.3 Axis jerk reserve.....	100
8.9 Jog mode (Jog).....	100
8.9.1 General information.....	100
8.9.2 Jog mode (increments and acceleration).....	101
8.9.3 Jog mode (jog velocity).....	102
8.10 Positioning logic/Enable modulo calculation (PosLog).....	105
8.10.1 General information.....	105
8.10.2 Enabling the modulo calculation.....	105
8.10.3 Positioning logic.....	106
8.10.4 Positioning logic can be switched via NC program.....	108
8.11 Location programming (LocProg).....	109
8.11.1 General information.....	109
8.11.2 Type of location programming.....	109
8.11.3 Splitting.....	110
8.11.4 Automatic location correction.....	110
8.12 Torque reduction (TorqRed).....	111
8.12.1 Reduced maximum torque when torque reduction enabled.....	111

	Page
8.12.2 Maximum standstill torque.....	112
8.13 Override for asynchronous axes (AsynchrAxOvrdf).....	112
8.13.1 General information.....	112
8.13.2 Max. override.....	112
8.13.3 Override value for step [1..32].....	113
9 Spindles (SP) - Basic parameters.....	115
9.1 General information.....	115
9.2 Spindle functions (SpFunc).....	115
9.2.1 General information.....	115
9.2.2 Definition of system spindle.....	115
9.2.3 Assignment of system spindle to channel spindle/spindle group.....	116
9.2.4 Functional overview.....	117
9.2.5 Basic data.....	117
9.2.6 Gears.....	120
9.2.7 Spindle override.....	128
9.2.8 Speed window.....	130
9.2.9 External spindle encoder.....	132
9.2.10 Jogging spindles.....	133
9.2.11 Spindle/Turret Axis.....	134
9.2.12 Spindle actuation for channel [k=1..n].....	135
9.2.13 Spindle syntax for system/channel Spindles.....	135
9.2.14 Spindle syntax[1..8].....	136
9.3 Spindle group functions (SpGrFunc).....	144
9.3.1 General information.....	144
9.3.2 Spindle group syntax[1..4].....	144
9.4 Spindle management (SpAdmin).....	152
9.4.1 General information.....	152
9.4.2 Spindles in the "ChSp" selection (channel spindle mode).....	153
9.4.3 Spindles in the "Sp" selection (compatibility mode).....	153
9.4.4 Block switching control.....	154
9.4.5 Waiting points in time for block switching control with strategy 2.....	155
9.4.6 Waiting points in time for block switching control with strategy 3.....	159
9.4.7 Strategy Determination for Block Switching Control.....	162
10 Sercos communication (SCS) - Basic parameters -	165
10.1 General information.....	165
10.2 Sercos Ring (Scslf).....	165
10.2.1 General information.....	165
10.2.2 Sercos Cycle Time (= Interpolation Cycle Time).....	165
10.2.3 Time monitoring for Sercos startup.....	165
10.3 Sercos Data Handling (Data Handl).....	166
10.3.1 Downloading Sercos Data in Phase 3.....	166
10.3.2 Sercos ID for NC command values.....	167
10.4 Virtual mode (VirtMode).....	167

Table of Contents

	Page
10.4.1 General information.....	167
10.4.2 Position of the reference point	167
10.5 Test mode.....	168
10.5.1 Enabling the test mode.....	168
11 Sercos parameters (SCSP) - Basic parameters -	171
11.1 General information.....	171
11.2 General Sercos settings.....	171
11.2.1 Storage mode.....	171
11.2.2 Encoder selection.....	172
11.2.3 Multiplex channel.....	172
11.3 Dynamic limitations.....	173
11.3.1 Position controller Kv factor.....	173
11.3.2 Torque limit value.....	173
11.4 Telegrams.....	174
11.4.1 User parameter display.....	174
11.4.2 Master data telegram.....	174
11.4.3 Drive telegram.....	175
11.4.4 Signal status word.....	175
11.4.5 Signal control word.....	176
12 NC/PLC interface (PLC) - Basic parameters.....	179
12.1 Overview.....	179
12.2 PLC selection (Plc).....	179
12.2.1 General information.....	179
12.2.2 IndraLogic.....	179
12.2.3 Resetting the CPL customer outputs to the PLC	180
12.3 Digital inputs/outputs (DigIO).....	182
12.3.1 General information.....	182
12.3.2 Hardware selection.....	182
12.3.3 Synchronization type of the high-speed I/Os.....	182
12.3.4 Initial address of digital inputs.....	183
12.3.5 Initial address of digital outputs.....	183
12.3.6 Function assignment of digital inputs.....	184
12.3.7 Function assignment of digital outputs.....	185
12.4 Analog inputs/outputs (AnaIO).....	185
12.4.1 General information.....	185
12.4.2 Initial address of digital outputs.....	186
12.4.3 Function assignment of analog outputs.....	186
12.4.4 Offset specification.....	187
12.4.5 Gain specification.....	187
12.4.6 Type selection.....	188
12.5 NC function-specific bit interface (NcFuncBitIf).....	189
12.5.1 General information.....	189
12.5.2 NC function for the bit interface iCh_ActFunc[1..24].....	189

	Page
12.6 Machine state display MZA (MachStatDiag).....	197
12.6.1 Cycle time.....	197
13 Auxiliary functions (AUXF) – Basic parameters -	199
13.1 Auxiliary function basics.....	199
13.2 Overview.....	201
13.3 Auxiliary function class definition (ClassDef).....	201
13.3.1 General information.....	201
13.3.2 Auxiliary function types.....	201
13.3.3 General parameters of auxiliary function classes class [1..30].....	202
13.3.4 Bit-coded auxiliary functions of auxiliary function classes class [1..30].....	205
13.3.5 BCD32-coded auxiliary functions of auxiliary function classes class [1..30].....	207
13.3.6 Integer-coded auxiliary functions of auxiliary function classes [1..30].....	210
13.3.7 Float-coded auxiliary functions of auxiliary function classes class [1..30].....	211
13.3.8 Double-coded auxiliary functions of auxiliary function classes class [1..30].....	212
13.4 Acknowledgement behavior for bit-coded auxiliary functions (BitAckn)	213
13.4.1 General information.....	213
13.4.2 Acknowledgement requirement from bit 0000 to bit 0127 (Byte000_015).....	213
13.4.3 Acknowledgement requirement from bit 0128 to bit 1535 (Byte016 to Byte191).....	215
13.5 Auxiliary function class definition (GrDef).....	216
13.5.1 General information.....	216
13.5.2 Channel dependency.....	216
13.5.3 Auxiliary function[1..15].....	217
13.6 Global output behavior (OutBehav)	217
13.6.1 General information.....	217
13.6.2 Output type.....	217
13.7 Number of usable auxiliary functions (NofAuxFunc).....	218
13.7.1 General information.....	218
13.7.2 Number of auxiliary functions.....	218
13.7.3 Channel-dependent auxiliary functions (ChDep).....	219
13.7.4 Channel-independent auxiliary functions (ChInDep).....	221
13.8 Addresses of auxiliary functions in the PLC interface (PlcIf).....	223
13.8.1 General information.....	223
13.8.2 Addresses in the channel interface[1..60] of channel-dependent auxiliary functions.....	223
13.8.3 Addresses of channel-independent auxiliary functions.....	226
13.9 Defining a new auxiliary function.....	228
13.10 Parameterizing example: Auxiliary functions.....	231
14 Graphical user interface (GUI) - Basic parameters -	239
14.1 General information.....	239
14.2 Measuring unit for display (DispUnit).....	239
14.3 Channel display (ChDisp).....	239
14.3.1 General information.....	239
14.3.2 Active channel displayed after switch-on	240
14.3.3 Hide channel within the interface.....	240

Table of Contents

	Page
14.4	Position display (PosDisp)..... 241
14.4.1	General information..... 241
14.4.2	Updating actual values in workpiece coordinates..... 241
14.4.3	Coordinate system selection..... 242
14.4.4	Display order of axes and channel coordinate packages..... 244
14.4.5	Display precision..... 251
14.5	Technology display for feed and speed..... 253
14.5.1	General information..... 253
14.5.2	Feed display..... 253
14.5.3	Speed display..... 254
14.6	Modal NC function group display (NcCodeDisp)..... 256
14.6.1	NC function for group[1..32]..... 256
14.7	Modal M-function group display (AuxFuncDisp)..... 257
14.7.1	M-function for group[1..32]..... 257
14.8	Display of zero point offsets (ZotDisp)..... 257
14.8.1	Number of zero point offset banks to be displayed..... 257
14.9	Number of placements (PmtDisp)..... 258
14.9.1	Number of placement banks to be displayed..... 258
14.10	Table display (TabDisp)..... 258
14.10.1	Decimal position for linear axes for metric table display..... 258
14.10.2	Decimal position for rotary axes for metric table display..... 259
14.10.3	Decimal position for linear axes for non-metric table display..... 259
14.10.4	Decimal positions for rotary axes for non-metric table display..... 259
15	NC programming (NCP) - Basic parameters..... 261
15.1	General information..... 261
15.2	Subroutine technology (SubProg)..... 261
15.2.1	General information..... 261
15.2.2	Non-modal G-functions GFunc[1..80]..... 261
15.2.3	Non-modal M-functions MFunc[1..8]..... 266
15.2.4	Modal subroutines SwiOn[1..63] / SwiOff..... 268
15.3	Polar coordinate programming (PolarCoord)..... 273
15.3.1	Syntax of angle 1..... 273
15.3.2	Syntax of angle 2..... 273
15.4	Corner rounding (CornRound)..... 274
15.4.1	Maximum allowed deviation from the contour..... 274
15.4.2	Minimum block length for corner rounding..... 274
15.4.3	Angle up to which the block transitions are seen as continuous..... 275
15.5	Diameter programming (DiaProg)..... 275
15.5.1	Effect of incremental specifications when programming the diameter..... 275
15.5.2	Channel axes/coordinates[1..8] that can be activated..... 276
15.6	Tool path correction (CutComp)..... 277
15.6.1	Starting and exiting correction build-up strategy..... 277

	Page
16 NC optimization (NCO) - Basic parameters.....	279
16.1 General information.....	279
16.2 Look-ahead.....	279
16.2.1 General information.....	279
16.2.2 Total number of blocks for block preparation and interpolation.....	280
16.2.3 Maximum number of function-related blocks.....	280
16.2.4 Percentage distribution of the total number of blocks.....	284
16.3 Memory management (Mem).....	288
16.3.1 General information.....	288
16.3.2 System memory.....	288
16.3.3 Channel-specific memory.....	289
16.4 File organization (FileOrg).....	291
16.4.1 General information.....	291
16.4.2 Search paths.....	291
16.4.3 Link process optimization for NC programs with the same name.....	294
16.4.4 Channel-specific setting of the file organization.....	294
16.5 Processor load (Perf).....	295
16.5.1 CPU time for block processing.....	295
16.6 Path velocity (PathVel).....	296
16.6.1 Keep velocity in motion-neutral NC blocks.....	296
16.7 In-Position evaluation (InPosEval).....	297
16.7.1 Mode of the In-Position evaluation.....	297
16.8 Position window (InPosWin).....	297
16.8.1 Inpos window, rough.....	297
16.9 Date/time setting (DateTime).....	298
16.9.1 General information.....	298
16.9.2 Winter.....	299
16.9.3 Summer.....	301
16.9.4 Clock change to winter time.....	303
16.9.5 Clock Change to Summer Time.....	304
16.10 Accuracy specification for circular programming (CircleAccur).....	306
16.10.1 General information.....	306
16.10.2 Center point programming.....	306
16.10.3 Radius programming.....	307
16.11 Scale factors (VelScaleFact).....	307
16.11.1 Channel-specific scaling factors for feed velocities and feed/revolution.....	307
16.11.2 Channel-specific scaling factors for rotary axis velocities.....	309
16.12 Precision programming (PrecProg).....	312
16.12.1 Contour deviation tolerance for "PrecProg" (PRP).....	312
16.13 Block transfer inhibit (BlkTransferLock).....	313
16.13.1 Standstill type at block end for block transfer inhibit.....	313
16.14 System units (CorrUnit).....	313
16.14.1 Units for tables and corrections.....	313
16.14.2 Default measuring unit for asynchronous linear axes.....	314
16.15 Database tables (DbTables).....	315
16.15.1 Buffer size for change event of the DB tables.....	315

Table of Contents

	Page
16.16 Communication server (CommServ).....	316
16.16.1 Enabling communication server.....	316
16.17 Web server (WebServ).....	316
16.17.1 Enable Web server.....	316
17 Operating functions (OPF) - Basic parameters	319
17.1 General information.....	319
17.2 Return to path (ReturnPath)	319
17.2.1 Return strategy.....	319
17.2.2 Return point.....	320
17.2.3 Maximum number of jog movements to be recorded.....	320
17.3 Re-entry (ReEntry).....	321
17.3.1 Function mode for re-entry.....	321
17.3.2 Directory for re-entry data.....	321
17.4 Handwheel functionality (Handw).....	322
17.4.1 General information.....	322
17.4.2 Number of increments/revolution.....	322
17.5 Jogging axes in workpiece coordinates (WcsJog).....	323
17.5.1 General information.....	323
17.5.2 Channel-specific data.....	323
17.5.3 Velocity data for the WCS jog interface.....	325
18 Safety functions (SPF) - Basic parameters	329
18.1 Overview.....	329
18.2 Waypoints (WayPoint).....	329
18.2.1 General information.....	329
18.2.2 Waypoint point[1..64].....	329
18.3 Area control (AreaCtrl).....	332
18.3.1 General information.....	332
18.3.2 Area monitoring Area[1..10].....	332
18.3.3 Dimension Dim[1..2].....	333
18.4 Distance control to programmed end point (AxDistCtrl).....	335
18.4.1 General information.....	335
18.4.2 Enabling the distance control.....	335
18.4.3 Distance to the programmed endpoint.....	335
18.5 One-dimensional collision monitoring (CollMon)	336
18.5.1 General information.....	336
18.5.2 Enabling the one-dimensional collision monitoring.....	336
18.5.3 Monitoring axis 1.....	337
18.5.4 Monitoring Axis 2.....	337
18.5.5 Traveling direction.....	338
18.5.6 Monitor upper limit.....	338
18.5.7 Monitor lower limit.....	339
18.5.8 Upper limit value.....	339
18.5.9 Lower limit value.....	340

	Page
19 Safety Engineering (SAFE) - Optional parameters -	341
19.1 General information	341
19.2 Drive-specific Safety parameters	341
19.2.1 Enabling intelligent Safety Engineering	341
19.2.2 Hiding Safety Engineering in the NC	341
19.2.3 Scaling factor for save velocities	342
20 Electronic axis coupling (COUP)Parameter-	343
20.1 General information	343
20.2 Axis coupling (AxCouple)	343
20.2.1 Maximum following error	343
20.2.2 Axis group index	343
20.2.3 Maximum standstill torque total of the group	344
20.2.4 Compensate Gantry offset command value	345
20.2.5 Upper limit for Gantry command value offset	345
20.2.6 Synchronous run monitoring on	346
20.2.7 Synchronous run monitoring window	347
20.2.8 Reference switch monitoring	347
20.2.9 Additional axes: System axis number of the master	348
21 Digitalizing (DIG) - Optional parameters	351
21.1 General information	351
21.2 Digitized data	351
21.2.1 Number of digitizing buffers	351
21.2.2 Number of reserve buffers	351
22 Measuring (MEAS) - Optional parameters	353
22.1 General information	353
22.2 Touch probes	353
22.2.1 Touch probe function available	353
22.2.2 Edge touch probe 1	354
22.2.3 Edge Touch Probe 2	354
23 Punching and nibbling (PUN) - Optional parameters	357
23.1 General information	357
23.2 Stroke monitoring	357
23.2.1 Enabling punching and nibbling	357
23.2.2 Stroke monitoring time	357
23.3 Axis-specific stroke release times	358
23.3.1 Reference of stroke release times	358
23.3.2 Stroke release times	359
24 Online corrections (ONL) - Optional parameters -	361
24.1 General information	361

Table of Contents

	Page
24.2 Online correction in MCS (distance control).....	361
24.2.1 General information.....	361
24.2.2 Enabling the online correction in MCS.....	361
24.2.3 Basic data.....	362
24.2.4 Sensor selection (Device).....	363
24.2.5 Device = Dr (Drive).....	364
24.2.6 Device = CplVar (permanent CPL variable).....	365
24.2.7 Increment calculation.....	365
24.2.8 Limits.....	366
24.2.9 Monitoring.....	368
24.3 Online correction in WCS.....	369
24.3.1 General information.....	369
24.3.2 Enabling the online correction in WCS.....	369
24.3.3 Sensor selection (Device).....	369
24.3.4 Device = Dr (Drive).....	370
24.3.5 Device = CplVar (permanent CPL variable).....	371
24.3.6 Device = Handw (Handwheel).....	371
24.3.7 Increment calculation.....	372
24.3.8 Limits.....	373
24.3.9 Axis interface for increment input.....	374
25 Traveling to positive stop (MPS) - Optional parameters.....	375
25.1 General information.....	375
25.2 Drive-dependent move to positive stop.....	375
25.2.1 Traverse to positive stop can be enabled.....	375
25.2.2 Maximum axis velocity.....	375
25.2.3 Maximum axis acceleration.....	376
25.2.4 Torque limit value.....	376
25.2.5 Maximum Permitted Position Deviation at Positive Stop.....	377
25.2.6 Position offset at positive stop.....	377
26 Splines (SPL) - Optional parameters.....	379
26.1 Overview.....	379
26.2 Spline parameters.....	379
26.2.1 Maximum permissible starting point/end point deviation.....	379
26.2.2 Maximum permissible relative path velocity error.....	379
26.2.3 Degree of connection spline.....	380
26.2.4 Overlapping of two partial sequences.....	381
26.2.5 Spline variant for vector orientation.....	381
26.2.6 Dimensional tolerance for B-spline approximation (default setting)	381
27 Tangential tool guidance (TTL) -Optional parameters.....	383
27.1 General information.....	383
27.2 Definition of tool axes.....	383
27.2.1 System axis number of the rotary axis.....	383

	Page
27.3	Number of tool edges..... 383
27.3.1	Number of edges..... 383
27.4	Angle settings for tool guidance..... 384
27.4.1	Rotate the tool by 180 degrees for change in direction..... 384
27.4.2	Intermediate block angle (after which an intermediate block is inserted)..... 385
27.4.3	Offset angle (between the symmetry axis of the tool and the path tangent)..... 386
27.4.4	Starting angle for the offset angle..... 387
27.5	Communication settings for PLC..... 387
27.5.1	NC-PLC communication before executing an intermediate block..... 387
28	Thread cutting (THRD) - Optional parameters..... 389
28.1	Overview..... 389
28.2	Cut mode (ModeSel)..... 389
28.2.1	Axis mode..... 389
28.2.2	Spindle mode..... 389
28.3	Velocity (Vel)..... 390
28.3.1	Velocity jump..... 390
28.4	Acceleration (Acc)..... 391
28.4.1	Path acceleration..... 391
28.5	Deceleration (Dec)..... 391
28.5.1	Path deceleration acceleration..... 391
28.6	Fast retract (Retr)..... 392
28.6.1	Activation fast retract..... 392
28.6.2	Axis number of retract axis..... 393
28.6.3	Retract vector of retract axis (main axis)..... 393
28.6.4	Axis number of the secondary axis..... 394
28.6.5	Retract vector of retract axis (secondary axis)..... 394
29	Tool and workpiece management (TLWP) - Optional parameters -..... 397
29.1	General information..... 397
29.2	Tool correction calculation..... 397
29.2.1	Tool corrections in the active WCS working plane (WcsActPlane)..... 397
29.2.2	Tool corrections in WCS coordinates-fixed assignment - (WcsFixed)..... 397
29.2.3	Channel axis/Logic axis..... 397
29.2.4	Correction direction..... 398
29.2.5	Tool corrections in TCS-coordinates - Fix assignment - (Tcs)..... 398
29.2.6	TCS coordinate..... 399
29.2.7	Correction direction..... 399
30	Axis transformation (TRA) - Optional parameters..... 401
30.1	General information..... 401
30.2	Axis transformation type (TrafoType)..... 401
30.3	System axes/coordinates of forward transformation (FwdInCoord)..... 403
30.3.1	System axis coordinate index (SACI)[1..8]..... 403
30.4	Joint length and angle parameters (JointPar)..... 404

Table of Contents

	Page
30.4.1 Length parameter/angle parameter.....	404
30.5 Axis/machine positions of reference position (RefPos).....	404
30.5.1 Reference position[1..8].....	404
30.6 Axis classification of system axes/coordinates (FwdInCoordClass).....	405
30.6.1 System axis/system coordinate[1..8].....	405
30.7 Epsilon environment.....	407
30.8 Reference orientation of the tool coordinate system TCS(TcsRefOri).....	407
30.8.1 TCS coordinate X.....	407
30.8.2 TCS coordinate Y.....	409
30.8.3 TCS coordinate Z.....	410
31 Velocity-dependent output (VDO) - Optional parameters.....	413
31.1 General information.....	413
31.2 Limit velocities (Vel).....	413
31.2.1 Upper velocity limit value.....	413
31.2.2 Lower velocity limit value.....	413
31.3 Command value at the analog output (Out).....	414
31.3.1 Command value when reaching the upper velocity limit value.....	414
31.3.2 Command value when reaching the lower velocity limit value.....	414
31.4 Coordinate selection for path velocity calculation (CoordSel).....	415
32 System data.....	417
32.1 General information.....	417
32.2 System data description.....	420
32.2.1 System Simulation "/Sys_Simulation".....	420
32.2.2 Status of bit events "/SysActBitEv".....	421
32.2.3 Control data of the function programmable distance-dependent signal "/SysAdvDistSig".....	421
32.2.4 Advanced control data of the function "Programmable distance-dependent signal"/SysAdv-DistSigE".....	428
32.2.5 Additional data of the function "programmable distance-dependent signal ""/SysAdvDistSigA".....	429
32.2.6 Automatic program selection after a system restart "/SysAPrg".....	431
32.2.7 Actual data of the system axes "/SysAxActData".....	434
32.2.8 System axis coupling command"/SysAxCoupleCmd".....	435
32.2.9 System axis coupling control"/SysAxCoupleCtr".....	453
32.2.10 System axis coupling status"/SysAxCoupleSta".....	456
32.2.11 System axis coupling tables"/SysAxCoupleTab".....	467
32.2.12 Block pre-run data for axes "/SysAxSRun".....	469
32.2.13 Data of the active axis transformation "/SysAxTrafo1" and "/SysAxTrafo2".....	472
32.2.14 Channel command data "/SysChCmd".....	477
32.2.15 SysCfgInfo "/SysCfgInfo".....	478
32.2.16 Active channel spindle data "/SysChSpActData".....	478
32.2.17 Spindle pre-run data "/SysChSpSRun".....	481
32.2.18 Block pre-run data"/SysChSRun".....	506
32.2.19 Test mode "/SysChTestMode".....	522
32.2.20 Waiting condition when waiting on bit events "/SysChWaitBitEv".....	523

	Page
32.2.21 Coordinate system display of the tool editor "/SysCoordSystem".....	524
32.2.22 System control "/SysCtrl".....	527
32.2.23 Control data for handwheel mode in WCS "/SysHandwWCS".....	527
32.2.24 System hardware status information "/SysHwStates".....	530
32.2.25 System interpolations axis positions "/SysIpoAcsPos".....	532
32.2.26 System interpolation basic workpiece positions "/SysIpoBcsPos".....	533
32.2.27 System interpolations machine positions "/SysIpoMcsPos".....	535
32.2.28 System interpolation path velocities "/SysIpoPathFeed".....	536
32.2.29 System interpolations workpiece positions "/SysIpoWcsPos".....	538
32.2.30 NC kernel diagnostics "/SysKernelDgn".....	540
32.2.31 Maximum number of NC kernel-controlled drives "/SysMaxAxNbr".....	542
32.2.32 Number of edges "/SysMaxEd" displayed.....	542
32.2.33 Control data for NC diagnostics "/SysNcDgnCtrl".....	543
32.2.34 Data for axis-parallel measurement.....	546
32.2.35 Return to path logic data "/SysReentry".....	552
32.2.36 Retrace logic data "/SysRetrace".....	557
32.2.37 List of the Sercos devices "/SysScsDevice" connected.....	559
32.2.38 IP addresses of the Sercos interface "/SysScsIpAdr".....	560
32.2.39 Settings for automatic Sercos parameterization "/SysScsDrPar".....	562
32.2.40 Control data to re-initialize selected SD elements after NC restart "/SysSDForce".....	563
32.2.41 Data of the memory space assignment of system data "/SysSdInfo".....	565
32.2.42 Actual data of system spindles "/SysSpActData".....	569
32.2.43 Command data for system spindles "/SysSpCmdData".....	570
32.2.44 Block pre-run data "/SysSRun".....	579
32.2.45 System status information "/SysStates".....	587
32.2.46 Data for cycle Swiveling G230 "/SysSwivel".....	589
32.2.47 Target information "/SysTargetInfo".....	593
32.2.48 Tool correction tool edge numbers "/SysTiActMaxEd".....	601
32.2.49 Active and preselected tool "/SysTool".....	602
32.2.50 Tool Corrections per edge "/SysToolEd".....	614
32.2.51 SysToolInsert "/SysToolInsert".....	617
32.2.52 Active and preselected tool at block pre-run "/SysToolSRun".....	618
32.2.53 Tool type data of tools in the tool catalog "/SysToolType".....	631
32.2.54 Variables for the tool change cycle TCH "/SysTch".....	633
32.2.55 Displaying the SRAM assignment "/SysSRamMemInfo".....	640
32.2.56 Checksum for Sercos configuration "/SysSercCfgInfo".....	641
32.2.57 Data of the zero point offsets (global) "/SysZOData".....	641
32.2.58 Data of the zero point offsets (axis-related) "/SysZODataA".....	643
32.2.59 Data of the zero point offsets (channel-related) "/SysZODataC".....	644
32.2.60 Interaction between simulation and NC program.....	645
33 Sercos initialization.....	649
33.1 General information.....	649
33.2 Sercos initialization sequence.....	649

	Page
34 Appendix.....	653
35 Service and support.....	675
Index.....	677

1 About this documentation

Editions of this documentation

Edition	Release date	Note
01	2019-09	First edition
02	2020-06	New: chapter 3.5 "Troubleshooting MTX advanced NC configurator reports error 31 after change in parameterization" on page 32 New: chapter "Reversed direction of the spindle rotation" on page 579 chapter 4.1 "Explanations" on page 33: Note on MTX advanced supplemented chapter 6.9 "NC-controlled drive (Dr[i])" on page 55, 6.9.1 Physical drive name: Note on MTX advanced supplemented chapter 6.10 "Drive system functions (AxFunc)" on page 59, 6.10.3 Channel assignment: Note supplemented chapter 32.2.43 "Command data for system spindles /SysSpCmdData" on page 570: ReverseRotation supplemented
03	2020-11	Chapter "Channels (CHAN) - Basic parameters -": chapter 7.2.4 "Delete SetPos Offset with a control reset" on page 68 modified (Defdb00203516) Chap. "NC/PLC interface (PLC) -Basic parameters-": chapter 12.5 "NC function-specific bit interface (NcFuncBitlf)" on page 189 corrected (Defdb00203516) Chapter "NC programming (NCP) - Basic parameters": chapter 15.2 "Subroutine technology (SubProg)" on page 261: Note added that G591 is not supported anymore from MTX 14VRS (Defdb00208668) Chap. "System data" revised: chapter 32.2.4 "Advanced control data of the function "Programmable distance-dependent signal"/SysAdvDistSigE" on page 428 added, chapter 32.2.18 "Block pre-run data/SysChSRun" on page 506 and chapter 32.2.44 "Block pre-run data/SysSRun" on page 579 changed or extended (Defdb00208292 , Defdb00208922)
04	2021-04	chapter 15.2.2 "Non-modal G-functions GFunc[1..80]" on page 261: Non-modal G-functions extended by G628 and G629 (WP-00082256) chapter "Formula for the master value calculation in the system axis coupling" on page 446, chapter "Incorrect master axis data when applying system axis coupling data " on page 455, chapter "Formula runtime error position in the system axis coupling" on page 459: SAC increase to 250 characters (WP-00082259)
05	2021	chapter "Selecting the permitted synchronization deviation of the system axis coupling" on page 440 supplemented (PBI 312413)

Tab. 1-1: Change record

1.1 Validity of the documentation

Overview on target groups and product phases

The following graphic refers to the bordered activities, product phases and target groups of the present documentation.

Example: The target group "Programmer" can "parameterize, program and configure" in the product phase "Engineering" using this documentation.

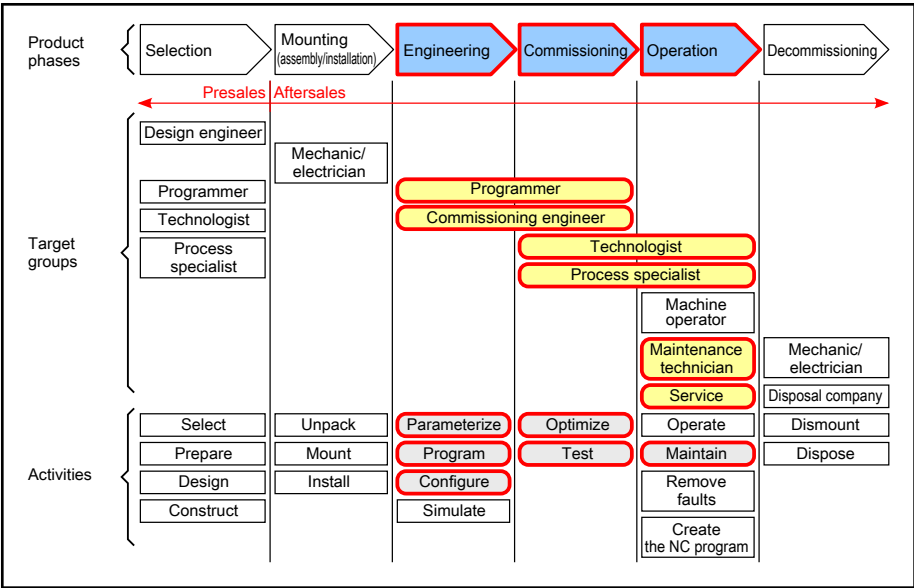


Fig. 1-1: Assigning this documentation to the target groups, product phases and target group activities

1.2 Required and supplementing documentation MTX

1.2.1 Selection/compilation

Documentation titles with type codes and part numbers

MTX 15VRS System Description DOK-MTX***-SYS*DES*V15-PRRS-EN-P, R911397742 This documentation describes the MTX control. It includes the designs, technical data, interfaces as well as the configuration of the control components.
MTX 15VRS SafeLogic System Overview DOK-MTX***-SL**SYS*V15-PRRS-EN-P, R911398637 This documentation describes the use of the safety control SafeLogic in the MTX.
IndraControl XM42 Controls DOK-CTRL-XM4X*CTRL**-ITRS-EN-P, R911345566 This documentation describes the IndraControl XM42 controls.

RS Corresponding edition
Tab. 1-2: MTX documentation overview - Selecting/composing

1.2.2 Configuration

Documentation titles with type codes and part numbers

MTX 15VRS Machine Parameters DOK-MTX***-MA*PAR**V15-RERS-EN-P, R911400176 This documentation describes structure and the adjustment of available parameters of the MTX. It also includes the functions of the NC configurator and its operation.
MTX 15VRS PLC Interface DOK-MTX***-PLC*INT*V15-PRRS-EN-P, R911400172 This documentation describes the interface signals and the program function blocks for the integrated PLC.

MTX 15VRS Functional Description - Basics

DOK-MTX***-NC*F*BA*V15-RERS-EN-P, R911401401

This documentation describes the basic functions of the MTX. The basic commissioning steps and the control functions are provided as description and handling instruction.

MTX 15VRS Functional Description - Extension

DOK-MTX***-NC*F*EX*V15-RERS-EN-P, R911393316

This documentation describes the extended functions of the MTX. The basic commissioning steps and the control functions are provided as description and handling instruction.

MTX 15VRS Functional Description - Special Functions

DOK-MTX***-NC*F*SP*V15-RERS-EN-P, R911393309

This documentation describes the special functions of the MTX. The basic commissioning steps and the control functions are provided as description and handling instruction.

MTX Free Form Surface Milling

DOK-MTX***-FREEFORM***-APRS-EN-P, R911341435

This documentation describes the free form surface milling process with the MTX control. CNC programs generated by a CAD/CAM system are used as basis for the entire process. Overview over the following topics: Description of the MTX or CNC function for freeform surface milling, strategy during the NC parameterization process, boundary conditions when generating CNC programs with CAM software.

MTX Conversion of MTX Projects

DOK-MTX***-PROCONV****-PRRS-EN-P, R911342484

This documentation provides support during the conversion of MTX 1.x projects to MTX 2G. This project planning manual provides information on how to convert a project and identifies potential difficulties during the conversion.

RS

Corresponding edition

Tab. 1-3:

MTX documentation overview - Configuring

1.2.3 Commissioning

Documentation titles with type codes and part numbers

MTX 15VRS Commissioning

DOK-MTX***-STARTUP*V15-CORS-EN-P, R911393281

This documentation describes the commissioning of the MTX control. Apart from a complete overview, commissioning and configuration of the axes and the user interface as well as the PLC data are described.

IndraWorks 15VRS Basic Libraries, IndraLogic 2G

DOK-IL*2G*-BASLIB**V15-LIRS-EN-P, R911398633

This documentation describes the system-comprehensive PLC libraries.

IndraWorks 15VRS Field Buses

DOK-IWORKS-FB*****V15-APRS-EN-P, R911393284

This documentation describes the field bus and local periphery connections supported by the MLC and MTX systems. The focus of this documentation lies in the configuration, parameterization, commissioning and diagnostics of different periphery connections. It is the basis for the online help.

IndraWorks 14VRS WinStudio 7.4

DOK-IWORKS-WINSTUD*V14-APRS-EN-P, R911341585

This "User Manual and Technical Reference Book" supports the user in achieving the best results with the "Rexroth WinStudio"™ software. The document provides technical information and step-by-step instructions to create web-enabled HMI/SCADA programs.

IndraWorks 15VRS Software Installation DOK-IWORKS-SOFTINS*V15-CORS-EN-P, R911393450 This documentation describes the IndraWorks installation.
IndraWorks 15VRS Engineering DOK-IWORKS-ENGINEE*V15-APRS-EN-P, R911393303 This documentation describes the use of IndraWorks in which the Rexroth Engineering tools are integrated. It includes instructions on how to work with IndraWorks and how to operate the oscilloscope function.
IndraWorks 15VRS PLC Programming System IndraLogic 2G DOK-IWORKS-IL2GPRO*V15-APRS-EN-P, R911396137 This documentation describes the PLC programming tool IndraLogic 2G and its use. The documentation includes the basic use, first steps, visualization, menu items and editors.
IndraWorks 15VRS HMI DOK-MLC***-HMI*****V15-APRS-EN-P, R911399270 This documentation describes the functions, configuration and operation of the user interfaces IndraWorks HMI Engineering and IndraWorks HMI Operation.
MTX 15VRS Shape Cutting Technology DOK-MTX***-TECHCUT*V15-CORS-EN-P, R911399147 This documentation describes the shape cutting technology used by the machines to cut materials using an NC-controlled tool head.
MTX-integrated CAD/CAM System for Shape Cutting DOK-MTX***-CAD*TECHCUT-CO01-EN-P, R911393320 This documentation describes the mode of operation and the area of application of the CAD/CAM nesting software Lantek Expert Inside.

RS

Corresponding edition

Tab. 1-4:

MTX documentation overview - Commissioning

1.2.4 Operation

Documentation titles with type codes and part numbers

MTX 12VRS Block Pre-Run DOK-MTX***-BLK*RUN*V12-APRS-EN-P, R911334379 This documentation explains to the machine manufacturer how to setup the "Block pre-run" function at the machine for the end user.
MTX 15VRS Programming Manual DOK-MTX***-NC**PRO*V15-RERS-EN-P, R911393318 This documentation describes the standard programming of the MTX control. Apart from the basics of NC programming, the use of NC functions according to DIN 66025 as well as the NC functions with high-level language syntax and CPL functions are described.
MTX 15VRS Standard NC Operation DOK-MTX***-NC*OP***V15-APRS-EN-P, R911393314 This documentation describes the operation of the standard user interface of the NC control of the MTX. It includes the operation of the interface, the NC program development as well as the tool management.
MTX 15VRS Multitouch DOK-MTX***-MULTI***V15-APRS-EN-P, R911393311 The Multitouch user interface of the NC control MTX is described in this documentation.

MTX 15VRS Standard NC Cycles

DOK-MTX***-NC*CYC**V15-PRRS-EN-P, R911394940

This documentation describes the application of the standard cycles of the different technologies for the MTX control.

MTX 15VRS NC Simulation

DOK-MTX***-NC*SIM**V15-APRS-EN-P, R911393273

This documentation describes the NC simulation for the MTX control.

MTX 15VRS Measuring Functions

DOK-MTX***-MES*FUN*V15-APRS-EN-P, R91194938

This documentation describes the measuring cycles of the MTX control.

RS

Corresponding edition

Tab. 1-5:

MTX documentation overview - Operating

1.2.5 OEM Engineering

Documentation titles with type codes and part numbers

MTX 13VRS Automation Interface

DOK-MTX***-AUT*INT*V13-APRS-EN-P, R911337274

This documentation describes the script-based access to IndraWorks project data via the interface of the Automation Interface. Different objects including code examples are described. The Automation Builder is also described in this manual.

MTX 15VRS OPC Communication

DOK-MTX***-OPC*COM*V15-PRRS-EN-P, R911399272

This documentation describes the syntax and the structure of the items for the communication with Bosch Rexroth devices.

IndraWorks OPC UA Communication

DOK-IWORKS-OPC*UA*****-APRS-EN-P, R911379309

This documentation describes the OPC UA communication of the MLC and MTX control systems.

RS

Corresponding edition

Tab. 1-6:

MTX documentation overview - OEM engineering

1.2.6 AddOns

Documentation titles with type codes and part numbers

MTX 11VRS Action Recorder

DOK-MTX***-ACR*****V11-APxx-EN-P, R911329943

This documentation describes the MTX action recorder. It includes the installation and commissioning as well as interface signals, application and operation.

MTX 15VRS Efficiency Workbench MTX cta, MTX ega

DOK-MTX***-EWB*****V15-APRS-EN-P, R911400178

This documentation describes the mode of operation and the area of application of the analysis tool MTX cta and MTX ega.

MTX Remote Condition Monitoring DOK-MTX***-RCM*****V01-APRS-EN-P, R911334383 This documentation describes the operation of the Remote Condition Monitoring System.
MTX visIREC User Documentation DOK-MTX***-VISIREC*V01-APRS-EN-P, R911344242 This documentation describes the analysis tool visIREC used to optimize the free form surface milling process. 2D or 4D display of path-related data. 2D or 4D display of coordinate-related data. Analysis of critical areas (path and orientation deviation). Comparing the programmed NC blocks to the interpolated NC blocks.

xx / RS Corresponding edition
Tab. 1-7: MTX documentation overview - AddOns

1.3 Information representation

1.3.1 Using safety instructions

Structure of the safety instructions

The safety instructions are structured as follows:

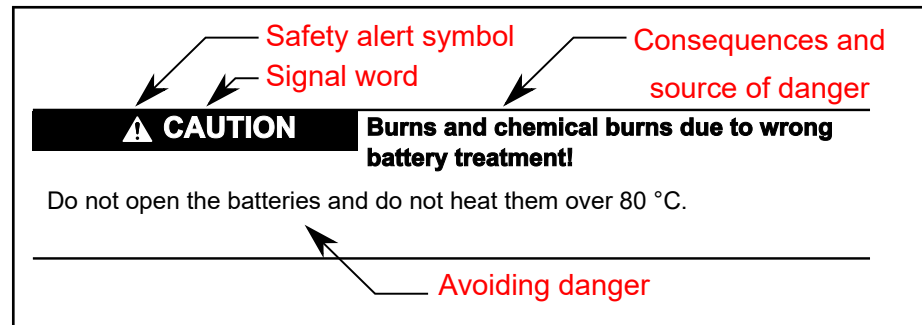


Fig. 1-2: Structure of the safety instructions

Explaining signal words and safety alert symbol

The safety instructions in this documentation contain specific signal words (danger, warning, caution, notice) and, if necessary, a safety alert symbol (according to ANSI Z535.6-2006).

The signal word draws attention to the safety instruction and indicates the risk potential.

The safety alert symbol (triangular safety reflector with exclamation marks), preceding the signal words Danger, Warning, Caution indicates hazards for persons.

DANGER

In case of non-compliance with this safety instruction, death or serious injury **will** occur.

WARNING

In case of non-compliance with this safety instruction, death or serious injury **can** occur.

⚠ CAUTION

In case of non-compliance with this safety instruction, minor or moderate injury can occur.

NOTICE

In case of non-compliance with this safety instruction, material damage can occur.

Symbols used

Pointers are displayed as follows:



This is a note.

Tips are displayed as follows:



This is a tip.

Explaining the signal alert symbol on the device



If this symbol is on your device, you have to observe the documentation on the device. The respective documentation informs on the type of hazard as well as the steps required to avoid this hazard.

1.3.2 Names and abbreviations

Term	Explanation
IWE	IndraWorks Engineering
IWO	IndraWorks Operation
NC	Numerical Control
OEM	Original Equipment Manufacturer
Profibus	Communication connection
Sercos	Sercos (Serial Realtime Communication System) is a world-wide standardized digital interface used for communication between controls and drives

Tab. 1-8: Terms and abbreviations

1.3.3 Customer feedback

Customer requests, comments or suggestions for improvement are of great importance to us. Please email your feedback on the documentations to Feedback.Documentation@boschrexroth.de. Directly insert comments in the electronic PDF document and send the PDF file to Bosch Rexroth.

2 Important instructions on use

2.1 Intended use

2.1.1 Introduction

Bosch Rexroth products are developed and manufactured according to the state-of-the-art. The products are tested prior to delivery to ensure operating safety and reliability.

The products may only be used as intended. If they are not used as intended, situations occur that result in damage to property or injury to persons.



Bosch Rexroth shall not assume any warranty, liability or payment of damages in case of damage resulting from a non-intended use of the products; the user shall solely bear all risks from unintended use of the products.

Before using Bosch Rexroth products, the following requirements have to be met to guarantee the intended use of the products:

- Anyone handling Bosch Rexroth products in any way is obliged to read and consent to the relevant safety instructions and the intended use.
- Hardware products may not be altered and have to remain in their original state; i.e. no structural changes are permitted. It is not permitted to decompile software products or alter source codes.
- Do not install damaged or defective products or use them in operation.
- It has to be ensured that the products have been installed as described in the relevant documentation.



Ensure that the data present in the control or entered or read in by the user is correct before applying it to exclude unwanted axis motion. It can be the following invalid or old data:

- Part programs
- ZO tables
- Compensation tables
- Tool tables
- Permanent CPL variables
- Remanent PLC data
- Permanent system data

2.1.2 Areas of use and application

The MTX control is used to

- program contour and machining technology (path feed, spindle speed, tool change) of a workpiece.
- guide a machining tool along a programmed path.

Feed drives, spindles and auxiliary axes of a machine tool are activated via Sercos interface.



This additionally requires I/O components for the integrated PLC which - together with the actual CNC - control the machining process as a whole and also monitors this process with regard to technical safety.

It may only be operated with the explicitly specified hardware component configurations and combinations and only with the software and firmware specified in the appropriate documents and functional descriptions.

The CNC system MTX provides the perfect CNC system solution for cutting and forming for the following technologies:

- Rotate
- Milling
- Drilling
- Grinding
- Bending
- Nibbling
- Punching
- Contour cutting
- Handling

2.2 Unintended use

Using the CNC system MTX outside the previously specified areas of application or under operating conditions other than the conditions described in the documentation and the specified technical data, is defined as "unintended use".

The CNC system MTX must not be used if...

- it is subjected to operating conditions not corresponding to the specified ambient conditions. Operation under water, under extreme temperature fluctuations or under extreme maximum temperatures is prohibited.
- Furthermore, the CNC system MTX shall not be used for applications not expressly approved of by Bosch Rexroth. Therefore, please read the information given in the general safety instructions!
- The CNC system MTX may not be used in systems or machines connected to the internet via an unsecure network connection. Otherwise, malfunctions or a control failure can result due to unauthorized access.

3 NC configurator

3.1 Basics

3.1.1 General information



This chapter contains **information on handling** the configuration files. For information on how the configuration files function, refer to the relevant chapters in the manual.

⚠ CAUTION

Danger due to arbitrary or unsolicited changes in the NC configuration!

With the NC Configurator, you can change the behavior of the control. Changes to the NC configuration, whether intentional or unintentional, can lead to severe damage to the machine, data loss or even personal injury!

⇒The NC may therefore only be configured by qualified personnel.

⇒The qualified personnel has to be able to detect and avoid the dangers involved in changing parameters.

3.1.2 NC Configurator function

The NC Configurator is used to:

- show and change the control parameters, consisting of the "configuration data"
- export and import the configuration data.



Saving/archiving/restoring the entire NC data or data limited to certain areas, such as configuration data, RAM file systems, user FEPROM, tool tables, perm. CPL variables or remanent PLC data (e.g. for backup or serial IB purposes) is functionally inaccessible within the NC Configurator and must be accessed at IndraWorks level in the context menu of the device "Motion"!

3.1.3 Configuration data structure

Optimal adjustability to any number of applications and, at the same time, diversity of functions typically require many parameters. It is therefore important to categorize the entire set of parameters (=configuration data) by topic.

Function groups are defined for this purpose, representing the order in which a control is to be configured. Every function group in turn contains topic areas (=data groups) into which logically related parameters are organized.

Function group	Data group	Note
Setup of configuration data	Selection "Optional Parameters" (SUP)	to select the data groups to be displayed in the function groups "Optional Parameters" and "OEM-specific parameters"
	Main NC configuration (MAIN)	for fundamental channel and drive configuration

Function group	Data group	Note
Basic parameters	Channels (CHAN)	for specific configuration of the respective topic area
	Axes (AX)	
	Spindles (SP)	
	Sercos communication (SCS)	
	Sercos parameters (SCSP)	
	NC/PLC interface (PLC)	
	Auxiliary functions (AUXF)	
	User interface (GUI)	
	NC programming (NCP)	
	NC optimization (NCO)	
	Operating functions (OPF)	
	Safety functions (SFU)	
Optional parameters	Intelligent Safety Technology (SAFE)	for specific configuration of the respective topic area Only those topic areas set to "Yes" as optional parameters under the "Setup (SUP)" data group are represented in this function group.
	Electronic couplings (COUP)	
	Digitizing (DIG)	
	Measuring (MEAS)	
	Punching and nibbling (PUN)	
	Online corrections (ONL)	
	Move to positive stop (MPS)	
	Splines (SPL)	
	Tangential tool control (TTL)	
	Thread cutting (THRD)	
	Tool and workpiece management (TLWP)	
	Transformations (TRA)	
	Velocity-dependent output (VDO)	
OEM-specific parameters	OEM data group (OEM)	to configure parameters defined by OEM. This function group is only displayed if the element "EnablOem" with the value "yes" is parameterized in the data group "Setup (SUP)".

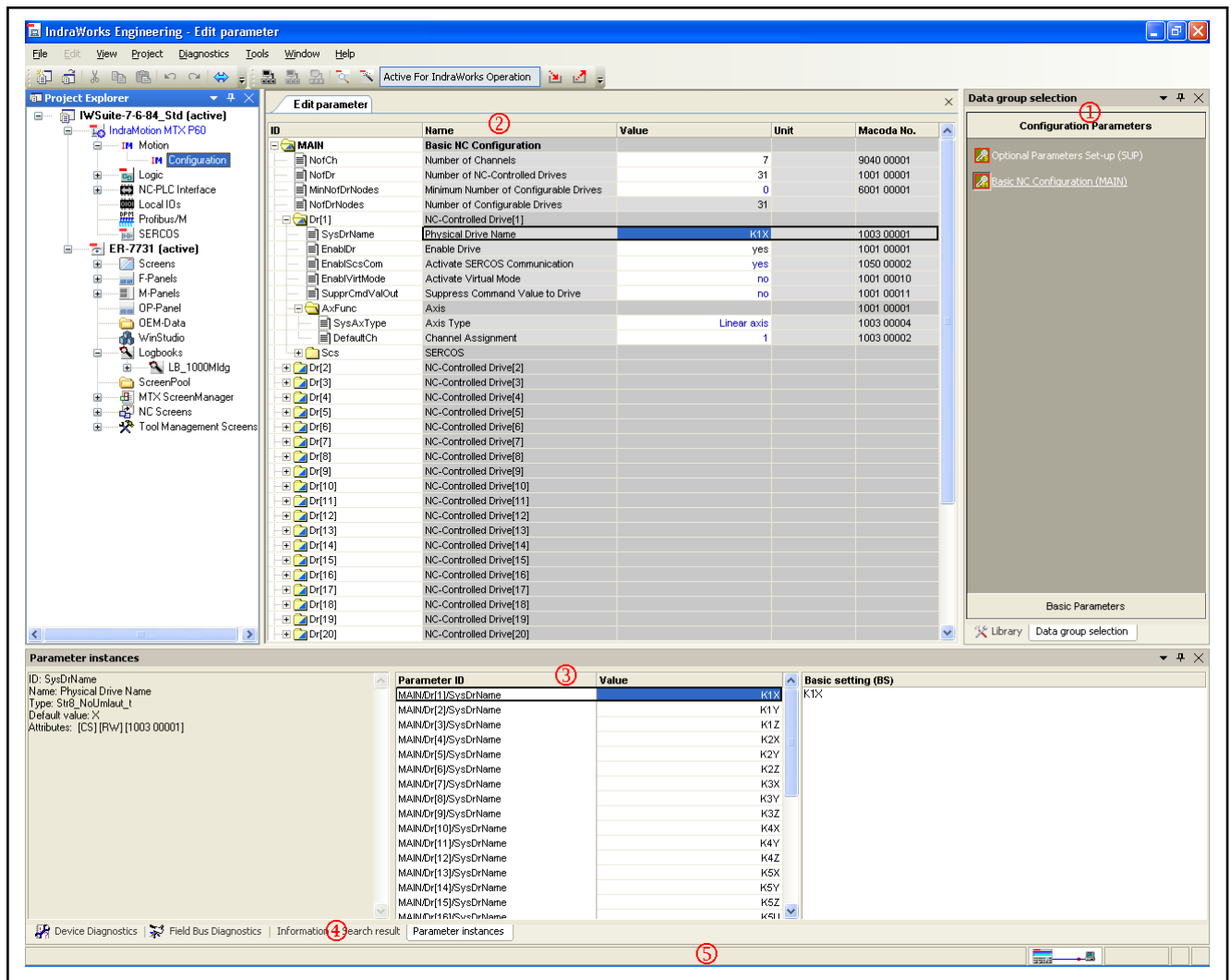
Tab. 3-1: Defined function and data groups

All single parameters can be arranged into a clear tree structure. The position in which a parameter is shown in this structure is described by the "parameter path". It contains the names, separated by "/", of all levels involved that you "go through" "on the way" to this parameter.

3.1.4 Display sections of the NC Configurator

General information

The GUI of the NC Configurator is completely embedded into the workbench of the application "IndraWorks Engineering" and uses the following display areas:



- 1 Display section "Data group selection"
- 2 Display section "Edit parameters"
- 3 Window for default display of the display areas "Information", "Search results" and "Parameter instances"
- 4 Tab to select the display areas "Information", "Search results" and "Parameter instances"
- 5 Display section "Status bar"

Fig. 3-1: Display sections of the NC Configurator

Within the workbench, you can distinguish different types of display areas:

- **Kernel display area**
of the embedded NC Configurator (here: "Edit parameters"). This type
 - exists only once within the workbench and
 - cannot be removed from the "IndraWorks Engineering" window.
- **Structure and additional information display areas**
(here: all other display areas identified in [fig. 3-1 "Display sections of the NC Configurator" on page 13](#) except for the status bar).
This type
 - can exist more than once within the workbench,

- can share the "IndraWorks Engineering" window with the kernel display area (selection with tab) and
- can also be placed on the desktop as a separate window.

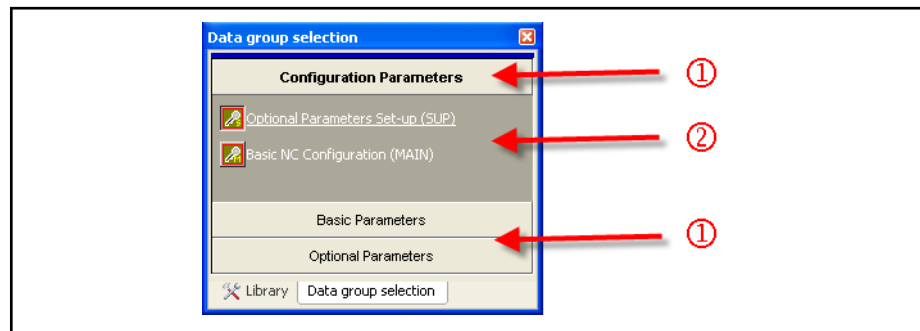


How to use the flexibility of the display area, refer to [chapter 3.2.3 "Adapting the display areas according to the information requirements"](#) on page 19.

Display area "Data group selection"

To **select a data group**. A selected data group is shown as parameter tree in the "Edit parameters" section.

Data groups are logically related configuration data. Function groups are logically related data groups. A summary of all function groups and data groups is given in [tab. 3-1 "Defined function and data groups"](#) on page 11.



1 Function groups

2 Data groups

Fig. 3-2: Display area "Data group selection"

A selected data group is depicted with as an elevated 3D data group symbol.

A selected data group (already shown in the "Edit parameters" section) is depicted by a deepened 3D icon and underlined name.



See [chapter 3.2.5 "Selecting the data groups"](#) on page 21 for choosing function groups and data groups.

Display area "Edit parameters"

For showing and editing all configuration data. The configuration data is visualized as tree structure starting with the name of the selected data group as root. The selected parameter object is shown with border or inversely.



- "Edit parameter" see from [chapter 3.2.6 "Navigating in the display area Edit parameters"](#) on page 21.
- Refer to [chapter 3.2.7 "Adapting columns in the display area Edit parameters"](#) on page 23 on how to configure the column display in the display area "Edit parameters".

Display area "Information"

The display area divided into two parts, with borders that can be moved with the mouse and containing the following information:

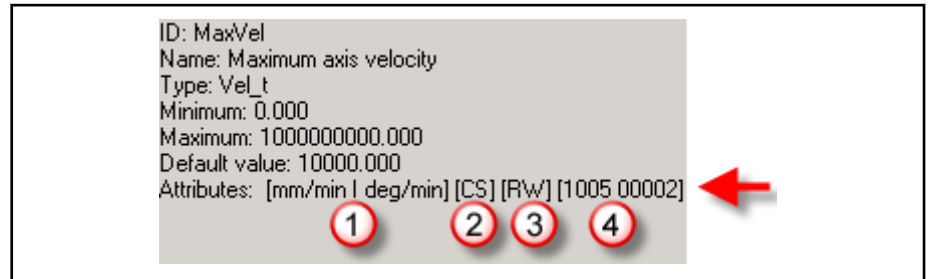
- **Detail data** (ID, name, type, minimum, maximum, default value, attributes) of the parameter object selected in the "Edit parameters" section and its

- relevant documentation section for the **parameter description** (if available).



By default, the display area uses a common window area shared with the display areas "Search results" and "Parameter instances". It can be brought to foreground by clicking on the corresponding tab.

The attribute values of a parameter are always displayed in a specific order in the detail data:



- 1 Unit of the parameter value
- 2 Time of enabling
- 3 Access rights
- 4 Machine parameter ID

Fig. 3-3: Order of parameter attributes within the detail data

Attribute	Value	Description
Unit of the parameter value		Physical unit
Time of enabling	CS	After NC cold start (System startup)
	SR	After system reset (System reset)
	MR	After module reset
	CR	After channel reset (Channel reset)
	EP	After main program end
	IM	Immediately
Access rights	H	Hidden (not visible)
	R	Read-only
	RW	Read and write
Machine parameter ID		Corresponding internal nine digit Maco- da no of the selected parameter

Tab. 3-2: Existing parameter attribute values

Display area "Search results"

Contains all objects found after a search. Double-clicking an entry (line) sets the focus to the corresponding parameter object in the "Edit Parameters" section.

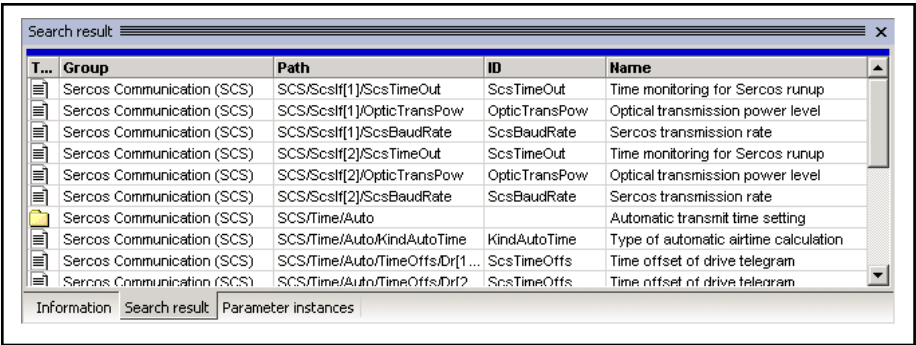


Fig. 3-4: Display section "Search results"



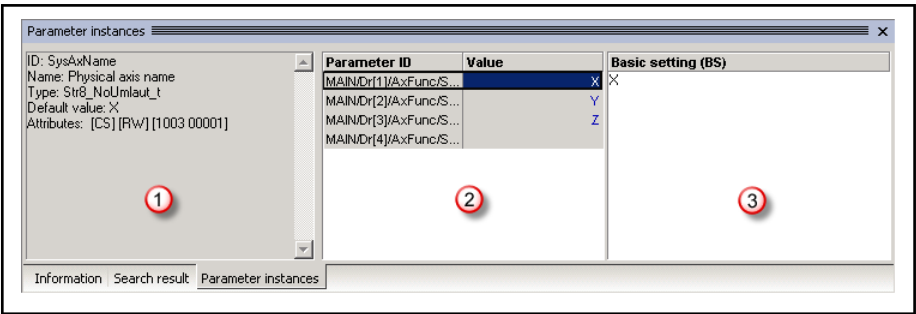
By default, the display area uses a window area shared with the display areas "Information" and "Parameter instances". It can be brought to foreground by clicking on the corresponding tab.

"Find parameter objects" see from [chapter 3.2.13 "Finding parameter objects"](#) on page 27 onwards.

Display section "Parameter instances"

The display area divided into three parts (see [fig. 3-1 "Display sections of the NC Configurator"](#) on page 13), with borders that can be moved with the mouse and containing the following information/functions:

- Display of **Detail data** (ID, Name, Type, Minimum, Maximum, Default value, Attributes) of the parameter object selected in the "Edit Parameters" section.
 - Display all existing **"Parameter instances"** of the selected parameter object in an editable list.
- "Parameter instances" are parameters that exist in the NC under the same ID but under different parameter paths. For example, the parameter "Maximum axis velocity" (MaxVel) exists individually for each applied axis. If you select such a parameter in the "Edit parameters" display area, it is possible to edit all existing instances easily using the list. That saves time and makes things clearer. Mostly, you encounter parameter instances in connection with drives, axes, spindles or transformations.
- Set all selected "Parameter instances" to **Basic setting** or to a user-defined value.



- 1 Detail data
- 2 Parameter instances of the selected parameter object
- 3 Display section "Parameter instances"

Fig. 3-5: Display section "Parameter instances"



By default, the display area uses a window area shared with the display areas "Information" and "Search output". It can be brought to foreground by clicking on the corresponding tab.

Display area "Status bar"

Displays (from left to right):

- the function and data group currently selected
- the number of hits after a search
- of the active filter criterion.

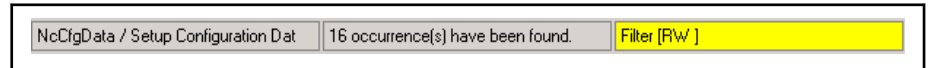


Fig. 3-6: NC Configurator: Information in the status bar

The status bar is an integral IndraWorks component.

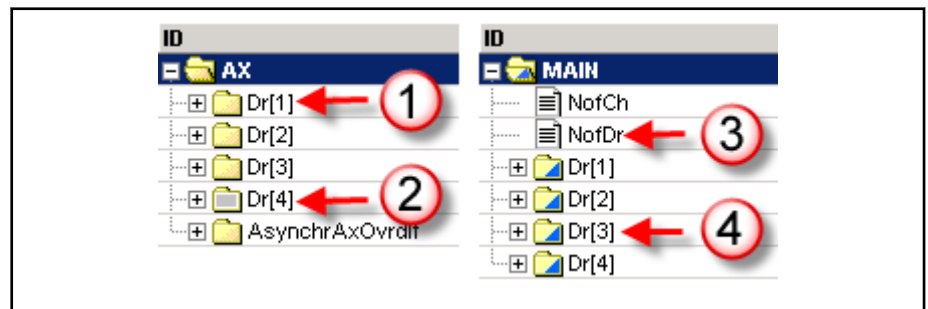
3.1.5 Icons and text colors used

Object icons (Icons)

The system displays the selected data groups in a hierarchy in the "Edit Parameters" display area.

Components of this structure can "object nodes" and parameters.

Object nodes represent places within the structure where the parameter path is split up into at least one more sublevel and/or at least one parameter.



1	Default object node
2	Locked object node
3	Parameter
4	Special object node

Fig. 3-7: Possible icons

Default object node	Object node that does not contain any of the menu items "New", "Convert to..." or "Delete" in its context menu (see also "Special object node").
Locked object node	Object node with sublevels that cannot be edited in the current data group due to the current parameter settings. For example, it is not possible to access a drive parameterized as spindle via the data group "Axis (AX)" because it is a spindle.
Parameter	Normal configuration parameter
Special object node	Object node that contains one of the menu items "New", "Convert to..." or "Delete" in its context menu.

Thus, the entire data structure under such an object node can be

- copied to create a new object node (menu item "New"; e.g. to create the parameter structure for another axis, more channels etc.)

- deleted, including the object node (menu item "Delete"; e.g. to delete the parameter structure of an existing axis, an existing channel etc.)
- converted to a different data structure (menu item "Convert to..."; e.g. to convert an axis to a spindle).

Text colors

Values of configuration parameters (displayed in the "Value" column) can be displayed in one of the following colors:

- blue:
parameter value corresponds to the system default value.
- red:
parameter value was edited in the current data group and saved to the internal database. The time of enabling depends on the parameter attribute "Time of enabling". Refer to [tab. 3-2 "Existing parameter attribute values"](#) on page 15.

3.2 How do I...

3.2.1 Starting the NC Configurator

Prerequisites:

The communication parameters of the control are set correctly. Specify the corresponding IP address of the NC and the relevant communication port correctly in the "Configuration" tab (default: Port 10099) in the context menu item "Properties..." of the control. The timeout period may not be set too short (default: 10 sec).

1. In the IndraWorks Project Explorer, expand the object tree of the control and its sub-object "Motion".

The object "NC configuration" is now visible.

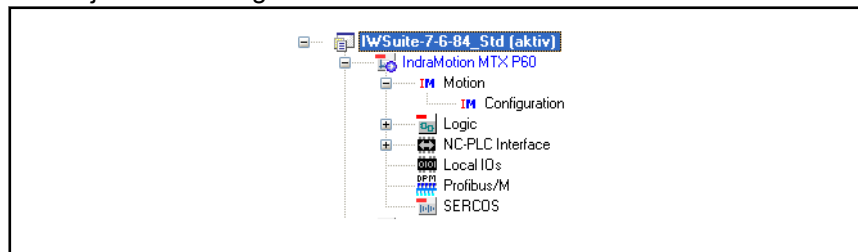


Fig. 3-8: Starting the NC configurator using the "NC configuration" object

2. Double-click on the "NC configuration" object or select the menu item "Open" from its context menu.

NC Configurator is started.

Since the proper establishment of a connection takes some time, the system shows a progress bar during this time.

3.2.2 Closing the NC Configurator

1. Select the display area "Edit parameters".
2. Close the display area "Edit parameters".

NC Configurator is closed.

Since the proper clearing of a connection takes some time, the system shows a progress bar during this period.

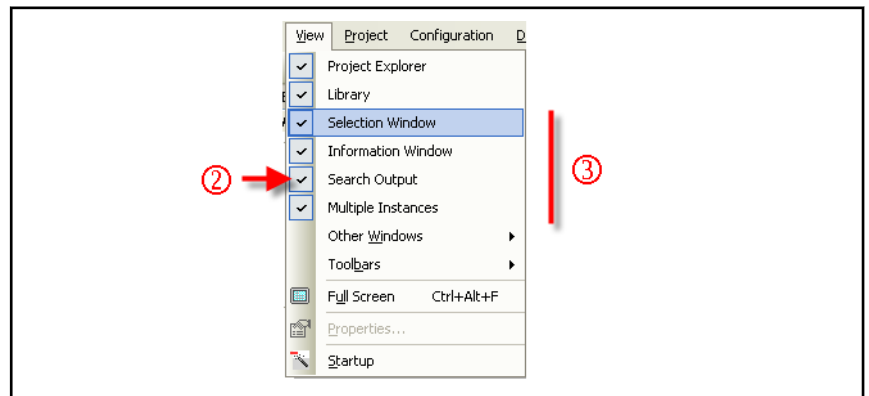
3.2.3 Adapting the display areas according to the information requirements

In order to adapt the default extent of information displayed to your special requirements, the display areas can be:

- switched on and off (exception: kernel display area)
- placed in the kernel display area.
A display area within the kernel display area is selected with a tab.
- placed in a separate window (exception: kernel display area).

Switching display areas on or off

1. Select "View" from the main IndraWorks menu.
2. Click on the appropriate menu item to switch a display area on or off. Display areas currently switched on are identified by a checkmark to the left of the menu item.



2 Area switched on

3 Display areas relevant to the NC Configurator

Fig. 3-9: Switching display areas on/off

Alternatives to switching a display area off:

1. Select the respective display area via the mouse.
2. Right-click in the header or on the tab of the selected display area and select the menu item "Hide".

Placing display areas into or removing display areas from the kernel display area

1. Select the respective display area via the mouse.
2. Right-click the header or the tab of the selected display area and select the menu item "Dockable".

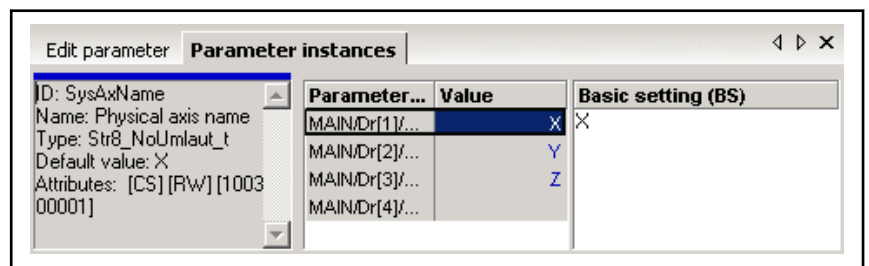


Fig. 3-10: Example: "Parameter instances" display area placed in the kernel display area



- The display area "Edit parameters" cannot be removed from the kernel display area.
- Display areas placed in the kernel display area cannot be placed as separate window on the desktop!

Placing display areas in separate windows or reintegrating them into the "IndraWorks Engineering" window

1. Select the respective display area via the mouse.
2. Right-click in the header or on the tab of the selected display area and select the menu item "Undocked".



If the menu item "Undocked" does not appear, the selected display area is currently in the kernel display area and must be removed from it first.

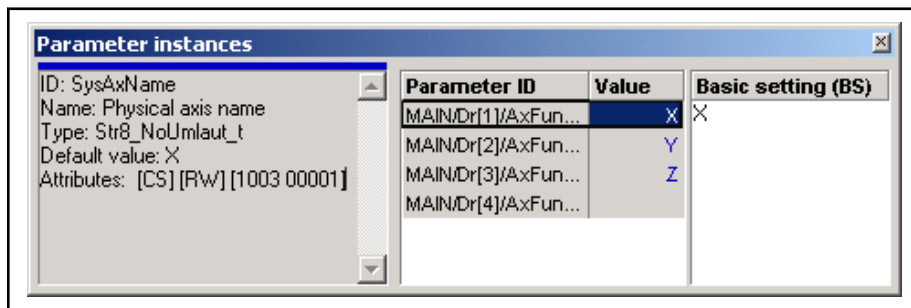


Fig. 3-11: Example: Display area "Parameter instances" as separate window



A display area as a separate window can also be placed back into the "IndraWorks Engineering" window via "drag & drop".

Using the "drag & drop" method, the target position is not restricted to the last position of the respective display area within the "IndraWorks Engineering" window and can be freely chosen.

1. Select the respective display area via the mouse.
2. Move the mouse cursor onto the header of the respective display area and hold it with the left mouse button.
The outline of the display area can now be freely moved.
3. Move the mouse cursor to the "IndraWorks Engineering" window.
The system indicates the allowed target positions using outlines.
4. To drop the display area into the indicated target position, let go of the left mouse button.

3.2.4 Selecting the display areas

For some actions, it can be necessary for the respective display area to be selected first.

With the mouse:

Left-click in the relevant display area.



- If several display areas have been placed in the kernel display area, you need to left-click the respective tab in the kernel display area.
- By default, the display areas "Information", "Search results" and "Parameter instances" share a common window area. In this case, select the display area by clicking on the corresponding tab.

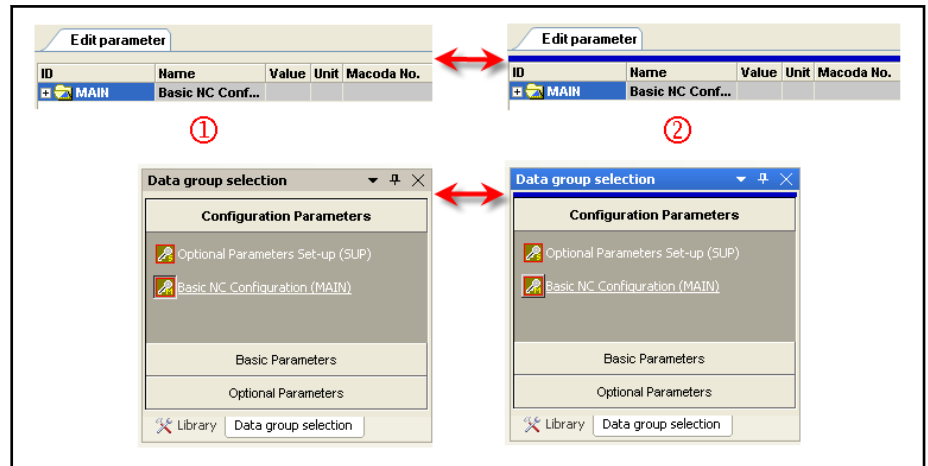
With the keyboard:

- Selecting in order:

keys<Ctrl>+<Tab>

- Selecting in reverse order:
keys<Ctrl>+<Shift>+<Tab>

Selected display areas (of the NC Configurator!) are indicated by a blue line under their header:



1 Not selected

2 Selected

Fig. 3-12: Selected/not selected display areas

3.2.5 Selecting the data groups

To **select a data group**, select first the function group and then the data group.

With the mouse:

Selecting function groups/data groups: Left-click

With the keyboard:

1. Select function group:
Key <Right arrow>, <Left arrow>.
2. Select data group:
Key <Down arrow>, <Up arrow>
3. Selecting data groups:
<Return> Key

A selected data group is depicted with as an elevated 3D data group symbol.

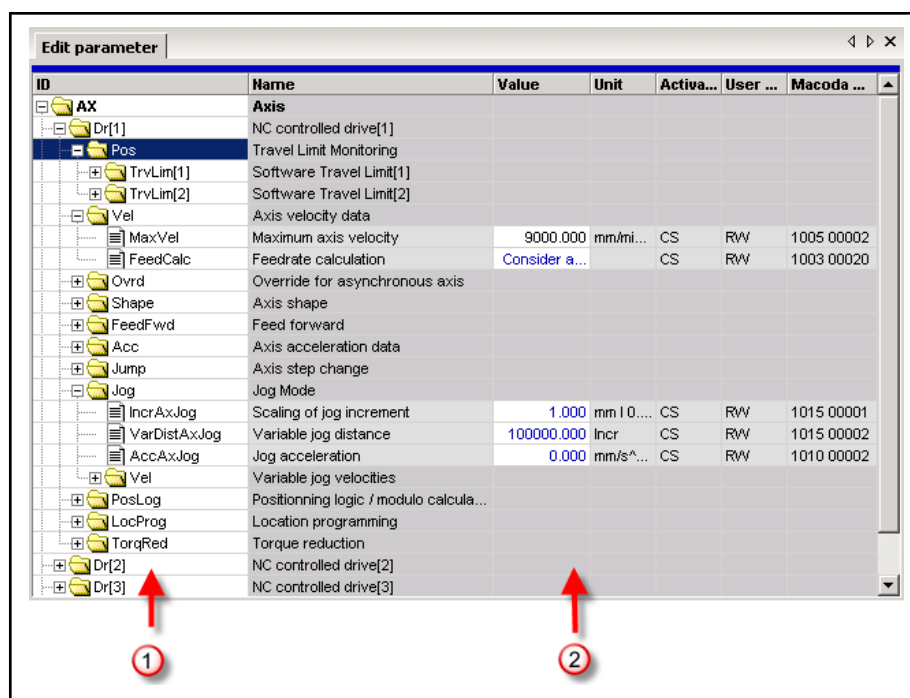
A selected data group is depicted by a depressed 3D icon and underlined name.

Selected data groups are shown in the "Edit Parameters" display area.

3.2.6 Navigating in the display area "Edit parameters"

The "parameter tree" is used to clearly display configuration data. It contains object nodes, which represent the branching into subordinate configuration levels as well as all individual parameters for the configuration. The path from the topmost object node to a single parameter is called the **parameter path**. Logically related parameters are grouped together into separate object nodes which, in turn, can be assigned to a superior object node.

A parameter is always edited in the "Parameter table".



- 1 Parameter tree
2 Parameter table

Fig. 3-13: Display section "Edit parameters"

Selecting and expanding/reducing single object nodes

With the mouse:

Click the plus or minus symbol to the left of the object node in question.

With the keyboard:

1. Select the object node with the key <Down arrow> or <Up arrow>.
2. Expand object nodes using the <Right arrow> key, reduce using the <Left arrow> key.

Expanding/reducing object nodes including all sublevels

With the mouse:

1. Open the context menu of the object node in question via right-click.
2. Select the menu item "Expand substructure" or "Reduce substructure".

Changing between parameter tree and parameter table

With the mouse:

- To change into the **parameter tree**, click within the parameter tree section.
- To switch to the **parameter table**, click on a line in the parameter table section that contains a parameter.

With the keyboard:

Via <Tab> key.

Selecting parameters

With the mouse:

1. Expand the object node that contains the parameter you want until the line with that parameter is shown.
2. Click the line in the parameter table section that contains the parameter you want.

With the keyboard:

1. Expand the object node that contains the parameter you want until the line with that parameter is shown.
2. Change to the parameter table.

3. Select the parameter with the key <Down arrow> or <Up arrow>.

3.2.7 Adapting columns in the display area "Edit parameters"

The display and order of the "Unit", "Enabling", "User rights" and "Macoda no." columns are configurable.

1. Select "Tools/Configure edit window..." from the IndraWorks main menu.

A dialog for column configuration is displayed:

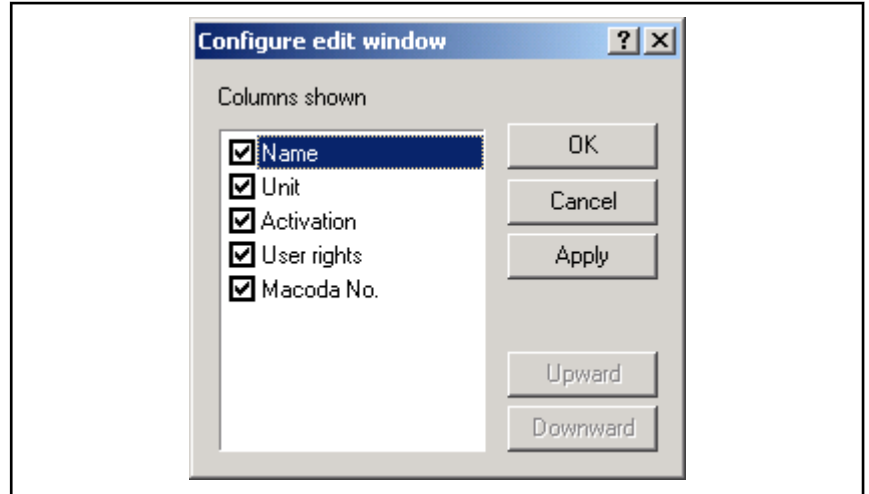


Fig. 3-14: Column configuration of the display area "Edit parameters"

2. Use the checkboxes to select which columns are to be hidden or visible.
3. Select the column to be moved. Define the display order via "Move up" or "Move down".



The order of the columns can also be changed in the display window by "drag & drop".

4. Apply your settings with "OK" (applies the changes and closes the dialog), "Apply" (applies the changes without closing the dialog) or cancel the dialog with "Cancel".

3.2.8 Edit parameters

With the mouse:

1. Expand the object node that contains the parameter you want until the line with that parameter is shown.
2. On the relevant line, double-click the field "Value".

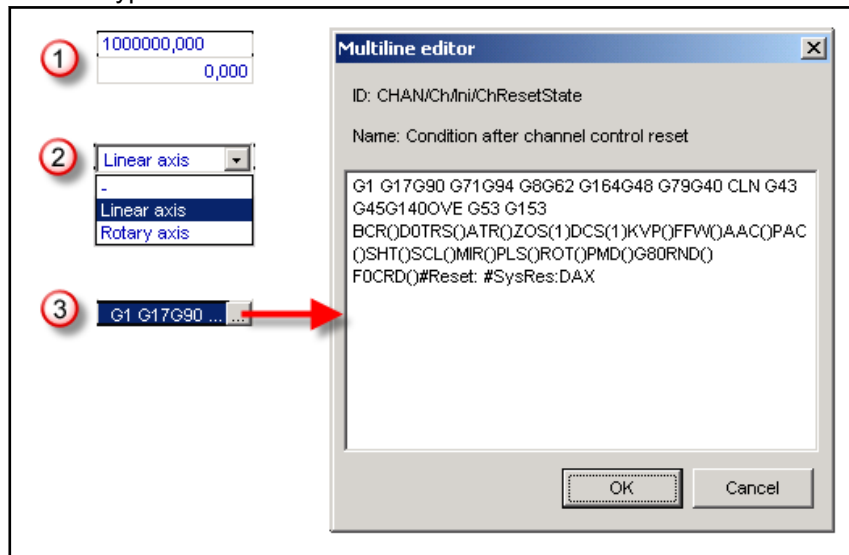
With the keyboard:

1. Expand the object node that contains the parameter you want until the line with that parameter is shown.
2. Using the <Tab> key, switch to the parameter table.
3. Select relevant parameters with the keys <Down arrow> or <Up arrow>.
4. The <Space> bar opens the "Value" field and it is used for editing purposes.
5. Apply any changed values with <Enter> or cancel the editing with <Esc>.



Parameter values that can be edited (text color: red) are automatically saved first by the system in the internal database. The time of enabling of the new values depends on the parameter attribute "Time of enabling" (see [tab. 3-2 "Existing parameter attribute values" on page 15](#)).

The "Value" field can be edited in different ways depending on the parameter type:



- 1 arbitrary one-line value input (input via keyboard)
- 2 predefined one-line value input (input from a list field)
- 3 arbitrary multiple-line value input (input via keyboard)

Fig. 3-15: Possible ways to edit

3.2.9 Undoing changes to parameters

Changes to parameters can be undone in steps of one action each. Select "Configuration/Undo" from the IndraWorks main menu.



In order to restore a parameter value to its default, select the parameter value and choose "Default Value" from its context menu. If the default value of the parameter is already entered, the menu item "Default Value" cannot be selected.

3.2.10 Showing parameter attributes

1. Select the respective parameter object in the "Edit parameters" display area.
2. If it is hidden, move the "Information" or "Parameter instances" display area to the foreground.

The attributes of the selected parameter object are shown in the detail data (see [chapter "Display area Information" on page 14](#)).

3.2.11 Copying parameter objects

Parameter objects and any sublevels contained can be copied to the clipboard. In return, copied data can be pasted back into appropriate parameter objects from the clipboard.

1. Select the parameter object in the "Edit Parameters" display area.

2. Select "Copy" from the context menu for that parameter object.

The data of the parameter object is copied to the clipboard and can then be pasted back into appropriate parameter objects using context menu item "Paste".

3.2.12 Filtering the display in the section "Edit parameters"

General information

In order to keep the amount of information in the "Edit parameters" display area limited to what is necessary, the following filter functions are available:

- Context filtering:
The filter criterion is the currently selected object node. Only this object node, including its sublevels, are shown when the filter is enabled. If several parameter instances of this object node exist, all parameter instances are also displayed.
- Optional filtering:
The filter criterion consists of the parameter attributes "User rights", "Unit" and "Time of enabling".

Enabling context filtering

1. Select the object node in the "Edit Parameters" display area.
2. Select "Context filter ON" from the context menu of that object node.
The message "Filter [Context Filter]" is displayed in the status bar.

Disabling context filtering

1. Select "Context filter OFF" from the context menu for any object node.
The message "Filter [Context Filter]" (in the status bar) disappears.

Enabling optional filtering

1. Select "Edit/Filter..." from the IndraWorks main menu.
The dialog "Filter" is displayed.
2. Select the checkbox "Enable filter" in the "Filter" dialog .

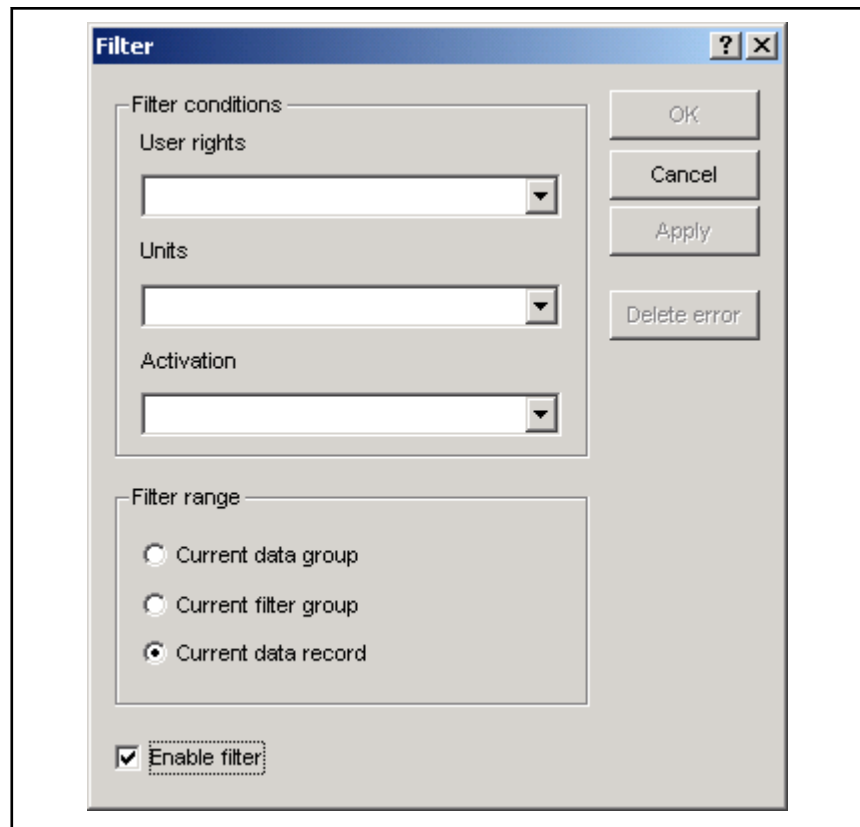


Fig. 3-16: Dialog "Filter"

- **"Filter conditions":**
List fields with parameter attributes to set the filter parameters. For possible attributes refer to [chapter "Display area Information" on page 14](#). If no entry is visible in a list field, that list field is no longer used for the filtering process. All conditions are linked by a logical AND. Only parameter objects for which all given conditions are true are shown in the "Edit Parameters" display area.
 - **"Filter range":**
Option to predefine the desired filter range.
3. Select the desired filter conditions and the filter range.
 4. Apply your settings with "OK" (applies the changes and closes the dialog), "Apply" (applies the changes without closing the dialog) or cancel the dialog with "Cancel".



The button "Delete filter" deletes all specified filter conditions.

The system shows the enabled filter conditions in the status bar.

Disabling optional filtering

1. Select "Edit/Filter..." from the IndraWorks main menu.
The filter dialog is displayed.
2. Disable the checkbox "Enable filter" in the Filter dialog.

3.2.13 Finding parameter objects

Search all or just some sections of the configuration data for defined criteria. The system collects all parameter objects found during a search in the "Search Results" display area and shows the number of hits in the status bar.

Double-clicking on an entry in the "Search results" display area shifts the focus to the corresponding parameter object in the "Edit parameters" section.

1. Select "Edit/Search..." from the IndraWorks main menu.

The dialog "Search" is displayed:

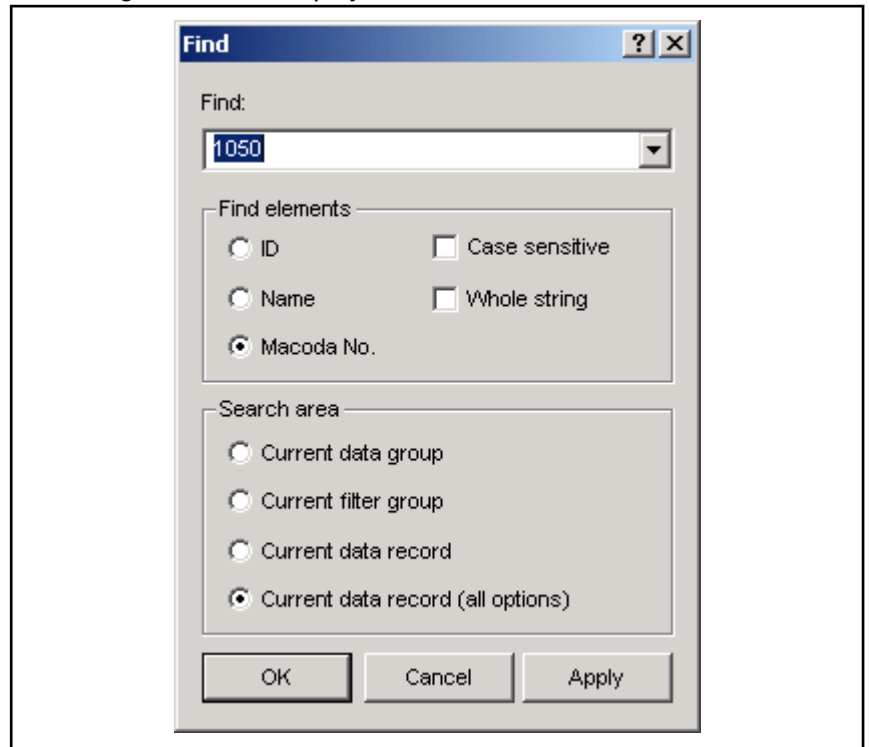


Fig. 3-17: "Search" dialog

- **"Find":**
Inputting the character string to be found.
The following placeholders are allowed in the string:
 - The character "*" for any string of any length
 - The character "?" for any single character.
- **"Find elements":**
Specifies whether ID, Name or Macoda No. is to be searched for in the configuration data. The search can be refined using the checkboxes.
- **Search area:**
Specifies where to search.
Note:
The setting "Current record (all options)" extends the search area to all data groups. This also includes data groups that could be parameterized in the function area "Optional parameters", but which do not exist there.

2. Specify the search pattern and select the desired search conditions and the search area.
3. Apply your settings with "OK" (this starts the search and closes the dialog), with "Apply" (this starts the search without closing the dialog) or cancel the dialog with "Cancel".

The display area "Search results" contains all findings discovered after a search.

3.2.14 Adding/removing channel and drive objects

The entire parameterization of a channel or drive is organized in the configuration data as a coherent channel data or drive data structure. If several channels/drives exist, there are also several instances of such data structures sequentially stored. They differ by the index n (channels: "Ch[n]"; drives: "Dr[n]").

When added, new channels or drive structures are always generated at the end of the existing instances.

When removing, always the last existing channels or drive structures are removed.

Example:

ID	Name	Value	Unit	Acti...	Use...	Mac...
MAIN	Basic Configuration					
NofCh	Number of channels	1	CS	RW	904...	
NofDr	Number NC controlled dri...	4	CS	RW	100...	
Dr[1]	NC controlled drive[1]					
Dr[2]	NC controlled drive[2]					
Dr[3]	NC controlled drive[3]					
Dr[4]	NC controlled drive[4]					

Fig. 3-18: Example: 4 drives exist

- Changing "NofDr" to 5: after "Dr[4]", "Dr[5]" is generated.
- Changing "NofDr" to 3: "Dr[4]" is deleted.



Data loss by reducing the current values of "NofCh" or "NofDr"!

The entire parameterization of the respective instance is deleted!

⇒ Ensure that the channel or drive instances to be deleted are no longer used.

⇒ If necessary, export the parameterization of the instances to be deleted using "Export" from the context menu. You can re-import the parameters after adding a new instance (of the same type) if necessary.

Procedure to change the number of channels or drives in the system:

1. In the display area "Data group selection", select the data group "Basic NC Configuration (MAIN)" from the function group "Setup of configuration data".
2. In the "Edit parameters" display area, edit the required number of channels (ID "NofCh") or drives (ID "NofDr").



Some object nodes can be recreated or deleted from the context menu of the superior object node (e.g. system spindles, from the context menu of the special object node "Spindle functions" (SpFunc). For "special object nodes", also see [fig. 3-7 "Possible icons" on page 17](#)).

The following applies:

- When adding, new structures are always generated at the end of existing instances.
- When removing, always the last of the existing instances are removed.

3.2.15 Changing the drive functionality

After adding a drive object, for example, it is necessary to define its drive type (axis, spindle, C--axis). The drive type specifies which parameters the data structure must include:

- AxFunc: Only axis-relevant parameters
- SpFunc: Only spindle-relevant parameters
- AxFunc/SpFunc: axis and spindle-relevant parameters (C-axis)



By default, the system always creates new drives as "AxFunc" (axis) drive type.

To change the drive functionality of a drive, proceed as follows:

1. In the display area "Data group selection", select the data group "Basic configuration (MAIN)" from the function group "Setup of configuration data".
2. Select the relevant drive object "Dr[...]" in the "Edit parameters" display area.
3. From the context menu for the drive object, select "Convert to..." and the desired drive type.

3.3 Importing/exporting configuration data

3.3.1 Exporting/importing the data group or the object node

For backup purposes or for quick duplication of object nodes (including all sublevels contained), the context menu for every object node contains the functions "Export" and "Import".

"Export" allows you to export the data group or a object node in ASCII format to an NC configuration file (*.cfg) desired.



File name + extension may not exceed 30 characters in the NC kernel (26 characters for the file name + four characters for point and extension).

"Import" loads the content of the saved NC configuration files back into the selected object node and its sublevels.



We recommend not to manually edit the content of NC configuration data. When changing the data structures, the import might not be possible anymore.

3.3.2 Importing/exporting a data block

For backup purposes of the complete data block (i.e. all data groups), the following functions are available under the menu item "NC Configurator":

- Import data block (default)
- Export data block (default)
- Import data block (alternatively)
- Export data block (alternatively)

Default mode When exporting and importing in the default mode, the data format used is the XML format defined by the World Wide Web Consortium (W3C). When exporting, a new directory is created under a freely selectable directory path. The name can be selected freely. An NC configuration file (*.cfg) is created for each individual data group in the directory mentioned above. All files form one data block. Additionally, another file called "<DirectoryName>.cds" is created representing the data block during import.

Alternative mode When exporting and importing in the alternative mode, a format especially defined by Rexroth is used as data format.



If you are not sure in which mode to export/import your data, select the default mode.

When exporting in the alternative mode, the configuration data of all data groups is stored in a single export file (*.cdt, cdt = configuration data table). This format allocates the parameters and their values line by line and in the display sequence of the data groups in the NC Configurator as well as in the display sequence within each data group. Each line contains single parameters with a complete path in the so-called XPath syntax and - separated by semicolon - the corresponding parameter value.

Example:

Format of an *.cdt export file

```
/SUP/FuncGr3/safe;0
/SUP/FuncGr3/coup;1
...
/SUP/FuncGr3/vdo;0
/SUP/FuncGr4/oem;0
/MAIN/NofCh;1
/MAIN/NofDr;4
/MAIN/MinNofDrNodes;0
/MAIN/NofDrNodes;4
/MAIN/Dr[1]/SysDrName;X
...
/VDO/Ch[1]/Out/AnaOutUpVelLim;100.00
/VDO/Ch[1]/Out/AnaOutLowVelLim;10.00
/VDO/Ch[1]/CoordSel;2
...
/OEM/Plc/AddrCplDialogForegr;-1
/OEM/GenPurpose/SIntValues/SIntVal[1];0
```


...

/OEM/GenPurpose/DoubleValues/DoubleVal[49];0.00000

/OEM/GenPurpose/DoubleValues/DoubleVal[50];0.00000

Use cases of the alternative mode

If two different data blocks of this format are compared to each other using a program to compare files, the difference between the two NC configurations can be determined easily. Thus, all differences from the default state can be identified for example. If you store these differences in a separate difference file, this file can be imported into another control for example.

Another use case can be the modularization of the NC configuration; that means the fragmenting of an export file into several files with partial data blocks and the sequential importing of these sets at a later time.

3.4 How to configure the NC



The following section solely explains the basic approach to configure the NC. It does not account for actions that could be necessary due to the peculiarities of a specific application, or when configuring specific NC functions.

1. In the display area "Data group selection", select the data group "Optional parameters setup (SUP)" from the function group "Setup of configuration data".
2. In the "Edit parameters" display area, set the required functions under the parameter object "FuncGr[1]" (optional parameters) to "Yes".

Thus, the system shows the required parameter structures you later select from the function group "Optional Parameters" for the parameterization.

3. In the display area "Data group selection", select the data group "Basic NC configuration (MAIN)" from the function group "Setup of configuration data".
4. In the "Edit parameters" display area, set the required number of channels (ID "NofCh") and drives (ID "NofDr"). The number of drives parameterized is the sum of all machining axes, auxiliary axes and spindles your application uses.

The system creates the necessary data structures. After this, the number of objects "Dr[...]" compatible with the ID "NofDr" must appear in the parameter tree (Dr: drive).

5. For each "Dr[...]" object, ensure its drive type is correct and change the drive type if necessary. Select the menu item "Convert to..." from the context menu of each drive object.

Drive type	Description
AxFunc	Drive with axis functionality
SpFunc	Drive with spindle functionality
AxFunc/SpFunc	Drive with axis and spindle functionality (C-axis)

Tab. 3-3: Available drive types



By default, the system always creates new drives as "AxFunc" (axis) drive type. The currently parameterized drive type cannot be selected under "Convert to..."

6. Set the remaining parameters of each drive, starting with the object "Dr[1]". We recommend to use the display area "Parameter instances" for these settings.
7. In the display area "Data group selection", select the individual data groups in sequence from the function group "Basic Parameters". Check all parameter settings and adjust the settings to your application if necessary.
8. In the display area "Data group selection", select the individual data groups in sequence from the function group "Optional parameters" (if available). Check all parameter settings and adjust the settings to your application if necessary.
9. Restart the NC and test the NC performance.
10. Correct the parameters if necessary.
11. Provided the NC is correctly parameterized, back up the configuration data in the user FEPROM. Select the menu item "Save configuration data permanently" from the context menu of the device "Motion" at the IndraWorks level.

3.5 Troubleshooting MTX advanced NC configurator reports error 31 after change in parameterization

Machine parameters are saved in the Macoda database in the NC kernel. If the number of channels is high (e.g. more than 30) and if multiple channel-specific parameters with a high number of individual entries were changed and if this parameter was changed in each channel, error 31 "NC kernel cannot write XML data" and additionally the error "Failed to open a MACODA block for write access" can be issued in the NC configurator. Changes in parameters cannot be made anymore.

Critical machine parameters are for example the following Macoda IDs:

- 600500020
(/GUI/PosDisp/Ord/Ch[i]/AddWin/AddAx/Dr[i]/ModeAddAx
and /GUI/PosDisp/Ord/Ch[i]/AddWin/AddAx/Dr[i]/PrioAddAx)
- 600500022
(/GUI/PosDisp/Ord/Ch[i]/ChWin/Dr[i]/ModeAxChWin
and /GUI/PosDisp/Ord/Ch[i]/ChWin/Dr[i]/PrioAxChWin)

An error results, as an internal limitation of the changed data sets was reached.

This limitation can be increased by a variable offset. There has to be the entry "<AdditionalMacodaParaDatasets>...</AdditionalMacodaParaDatasets>" in the file startup.xml in the root directory and in the directory /usrfep (template in the directory /feprom) and the entry has to be modified. Do not increase the offset value in very big steps, as less memory would be available for the remaining system. Start with a value of 10,000 and increase in steps of 5,000 if required.



To apply changes in the startup.xml, start with start-up mode 6 (bootstrapping).

4 Explanation of parameters and terms used

4.1 Explanations

Parameter names The following parameter description describes all parameters of the individual function and data groups in the same order as they are displayed in the NC configurator. Each parameter has an individual subchapter whose name is the parameter name.

The two following entries are especially highlighted in the table line at the beginning of each chapter, since they identify a parameter:

- **Parameter name:**
Name of the parameter in clear text (in table line, left).
- **Parameter ID:**
Short name for a parameter in the Configurator index tree (in table line to the right).

Parameter content and properties *Apart from the description of the parameter purpose and possible supplemented notes, restrictions or conditions the following information is listed for every parameters:*

- **Path:**
Every parameter is stored in a folder that is uniquely identified by specification of the search path in the Configurator index tree. The path always begins with the ID of the respective data group.
- **Value range:**
Specifies the input range for the parameter by specifying a lower and upper input value limit. Input values outside the range are not permitted.
In certain cases, it can happen that the specified value range is even more limited by internal controls.
- **Pre-assignment:**
Default parameter value within the value range (=factory setting).
If only **one** default value is specified, this **value** applies only for that parameter, or for all parameters equally if there are several instances (e.g. P[0..64]) of a parameter.
If there are several instances of a parameter **and** they have different default values, each default value will be assigned to a specific instance of the parameter.
Example:
0.000 -> P[0]
0.001 -> P[1]
0.005 -> P[2]
- **Unit:**
Unit refers to the unit system assigned to a parameter value. This unit system contains different basic units:
 - Length (metric or inches)
 - Angle (degrees)
 - Time (seconds)
 - Current, voltage (amps, volts)

- {Text}, {NC syntax} (alphanumeric characters such as letters, digits or special characters)

- **Meaning of parameter values:**

Here, more detailed explanations are given on the parameter values within the value range. Many parameter values consist of Yes/No or of short texts with specific meaning. For {Text} input, information on the possible content is defined here.

- **Access rights:**

The access rights specify whether a parameter may be read only or may be modified.

There are two types of access rights:

- RW: Read and write (= parameter value can be changed!)
- R: Read only (= parameter value cannot be changed!)

Five "user groups" are defined under the MTX user management.

As a rule, the access rights to a parameter are set for each of the user groups as follows:

- MTB (Machine Tool Builder): Read and write
- Setter: Read
- Operator: Read
- Service: Read and write (only allocated in exceptional cases!)
- Engineer: Read and write (only allocated in exceptional cases!)

- **Enabling:**

The enabling mode specifies how a parameter change is applied to become effective in the internal controls. In almost all cases, one of the following enabling modes is required:

CS: System cold start

SR: System reset

Other possible enabling modes are described in [chapter 3 "NC configurator" on page 11](#).

- **Dependency:**

Indicates dependencies on other parameters.

- **Macoda ID:**

Existing 9-digit parameter number still used internally, but which has been replaced by the new Parameter ID in NC Configurator.

- **Data type:**

Here, information is given on the data type of the parameter in NC Configurator and in Macoda (for compatibility reasons).

Data type in NC Configurator:

The data type is defined in an XML schema file and contains the following basic types for the parameters described here:

- int: Integer (whole or natural number)
- Float: 4 byte real (real numbers)
- double: 8 byte real (real numbers)
- string: (Characters)

- Boolean: (true, false)

The name of a "string" data type for a parameter could be as follows, for example:

Str8_t (isoLatin1String) → "8 characters"

Data type in MACODA:

The data type in MACODA is defined as follows:

Display and input format:

Integer: (whole or natural number)

Real: real number

String: Text

Supplementary information for data type:

signed: with plus or minus sign

unsigned: without plus or minus sign

Float: with floating decimal point

There exist different **element lengths** for each data type:

1 byte (integer, unsigned): 0...255

1 bytes (integer, signed): -127...+127

1 byte (string): 1 character

2 bytes (integer, unsigned): 0...65535

2 bytes (integer, signed): -32767...+32767

2 bytes (string): 2 characters

4 bytes (integer, unsigned): 0...4294967295

4 bytes (integer, signed): -2147483647...+2147483647

4 bytes (string): 4 characters

4 bytes (real, floating decimal point)

4 bytes (real, with exponent)

8 bytes (string): 8 characters

8 bytes (real, floating decimal point)

8 bytes (real, with exponent)

Example Description of the parameter "Physical axis name"

Physical drive name	SysDrName

Specifies a system-wide, unique, channel-independent **system name** for each drive Dr[i].

If the drive is defined as a "spindle with C-axis operation", the name of the **C-axis** of the spindle must be given.

Explanation of parameters and terms used

		 - The system names can be overridden by the names of the channel axes (see CHAN/Ch[k]/Coord/Acs/ChAx[1..8]/ChAx-Name (axis name)). - The drives are displayed in the NC configurator at least up to the last drive that was provided with a system drive name (not equal to empty string). Do not select a maximum number of NC-controlled drives (MaxNofDr) that is smaller than the last axis index with defined system drive name.
Path	MAIN/Dr[i]/SysDrName	
Value range	{Text}	
Pre-assignment	X	MAIN/Dr[1]/SysDrName
	Y	MAIN/Dr[2]/SysDrName
	Z	MAIN/Dr[3]/SysDrName
	No entry	MAIN/Dr[4..250]/SysDrName
Unit	-	
Description of parameter values	Characters	e.g.: X System name (name with letters only). Letters denoting an internal NC function (e.g. D, G, H, R, S, F, A, FA) are not allowed here. See "MTX 15VRS Programming Manual" (DOK-MTX***-NC**PRO*V15-RERS-EN-P).
	Characters + digits	e.g.: X2 System name (name with letters and one numerical sequence). To distinguish between the axis name and the following position data, write either "=" or a "blank" between them (X2=111.22 or X2 111.22).
		NOTICE Danger of damage to the workpiece or the machine!
		If a short axis name is part of a longer axis name (for example: Axes "X" and "X2") and if a decimal point is programmed after that, the longer axis name is always applied: X2.5 → Axis X2 travels to 0.5 If, on the other hand, you want axis X to travel to position 2.5, you must enter either "=" or a "blank" as delimitation: X=2.5 or X 2.5 → Axis X travels to 2.5
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00001	
Data type	Str8_NoUmlaut_t (isoLatin1String)	
	8-byte string, no umlauts for MACODA compatibility	

4.2 Terms

The following NC-specific terms are used for the following parameter description:

Sercos	Protocol to exchange data between the MTX (master) and the drives (slaves) connected via Ethernet.
Drive	Generic term for spindles and axes. Every drive is a device in the Sercos ring.
Spindle	Drive parameterized as a "spindle" or "spindle with C-axis operation". A spindle is usually operated in revolution speed control or serves to drive the tool (on lathes: to drive workpieces). A spindle may also be used to carry out simple positioning tasks (spindle orient) in position mode. A spindle with C-axis operation is a drive that can be operated both as a spindle and as an axis.
Axis	Drive that is not parameterized as a "spindle" or "spindle with C-axis operation". An axis is typically operated in position control mode. Axes are used to move and position tools, workpieces, etc. A distinction is made between the two task-specific types "machining axes" and "auxiliary axes".
Machining Axis	Axis (also called a "synchronous axis"), assigned to a channel. Only machining axes can form an interpolative relationship. The axes have to be assigned to the same channel (axis group). Axis groups consisting of a minimum of 2 machining axes are used to carry out movements on a plane, while axis groups consisting of a minimum of 3 machining axes are used to carry out movements in space. It depends on the design of the machine used whether the tool or the workpiece is moved by machining axes.
Auxiliary Axis	Axis (also called an "asynchronous axis"), that is not assigned to any channel. These axes can be driven from any channel and are usually used for handling or infeed tasks on the machine.
Linear axis	Linear axes can be moved within their respective travel limit range only and are programmed in "mm" or "inches".
Rotary axis	Rotary axes can be moved within their respective travel limit range only and are programmed in "degrees".
Endless Axis	Special type of rotary axis: Endless axes behave like rotary axes without a limit switch; that is, they can turn an arbitrary number of times in the same direction (modulo calculation).
Hirth axis	Special type of rotary or endless axis: Hirth axes can only travel to positions located on a fixed grid . All grid positions are equally distanced.
Coordinate	Coordinates describe a point (position of the tool) on the machine via spatial coordinates and/or pseudo coordinates (logical axis names = channel name) with active axis transformation. The axis transformation converts positions to physical axes (drives) internally.
Channel	(Machining channel) part program unit that can interpolatorily control an axis group assigned to it (see machining axis). Additionally, any spindle and any auxiliary axis in the system can be driven from each channel.



For a detailed description of axes, spindles and coordinates, please refer to the manual "Functional Description 15VRS - Basics" (DOK-MTX***-NC*F*BA*V15-RERS-EN-P).

5 Optional parameter setup (SUP) - Setup of configuration data –

5.1 General information

The data group "Optional parameters setup" allows you to individually show or hide data groups in the function group "Optional parameters".

The following functions can be activated for display and parameterization in the configurator "[Optional Parameters](#)" from [page 39](#) onwards:

- Intelligent Safety Technology (SAFE)
- Electronic axis coupling (COUP)
- Digitizing (DIG)
- Measuring (MEAS)
- Punching and nibbling (PUN)
- Online corrections (ONL)
- Move to positive stop (MPS)
- Splines (SPL)
- Tangential tool control (TTL)
- Thread cutting (THRD)
- Tool and workpiece management (TLWP)
- Transformation (TRA)
- Velocity-dependent output (VDO)

The specific settings of this data group of the "Optional parameters" function group are given in [chapter 19 "Safety Engineering \(SAFE\) - Optional parameters -"](#) on [page 341](#) to [chapter 31 "Velocity-dependent output \(VDO\) - Optional parameters"](#) on [page 413](#).

The displaying and parameterization of OEM-specific parameters (see [chapter 5.3 "OEM-specific parameters"](#) on [page 45](#)) can be activated with:

- OEM data group (OEM)



Individually showing or hiding data groups in the "Optional parameters" function group has absolutely **no** influence on the function in the NC kernel.

The parameters of the "Optional parameters setup" data group are only present within NC Configurator and not in the control kernel. When you close NC Configurator, these parameters are saved as an XML file in the file system of the control kernel. The file is named "sup.cfg" and is located under the directory root\cfg of the MTX.

5.2 Optional parameters

5.2.1 Intelligent Safety Technology (SAFE)

Intelligent Safety Technology (SAFE)	EnablSafe

Enables or disables the display of the data group "Intelligent Safety Technology (SAFE)" in the "Optional parameters" function group.

Path SUP/FuncGr[1]/EnablSafe

Optional parameter setup (SUP) - Setup of configuration data –

Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Intelligent safety technology (SAFE)" does not exist
	1 (Yes)	Data group "Intelligent safety technology (SAFE)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.2.2 Electronic axis coupling (COUP)

Electronic axis coupling (COUP)		EnablCoup
Enables or disables the display of the data group "Electronic axis coupling (COUP)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablCoup	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Electronic axis coupling (COUP)" does not exist
	1 (Yes)	Data group "Electronic axis coupling (COUP)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.2.3 Digitizing (DIG)

Digitizing (DIG)		EnablDig
Enables or disables the display of the data group "Digitizing (DIG)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablDig	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	

Description of parameter values	0 (No)	Data group "Digitizing (DIG)" does not exist
	1 (Yes)	Data group "Digitizing (DIG)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.2.4 Measuring (MEAS)

Measuring (MEAS)		EnablMeas
Enables or disables the display of the data group "Measuring (MEAS)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablMeas	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Measuring (MEAS)" does not exist
	1 (Yes)	Data group "Measuring (MEAS)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	1 byte integer, unsigned	

5.2.5 Punching and nibbling (PUN)

Punching and nibbling (PUN)		EnablPun
Enables or disables the display of the data group "Punching and Nibbling (PUN)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablPun	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Punching and nibbling (PUN)" does not exist
	1 (Yes)	Data group "Punching and nibbling (PUN)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Activation -
 Dependency -
 Macoda ID -
 Data type byte

5.2.6 Online corrections (ONL)

Online corrections (ONL)		EnablOnl
Enables or disables the display of the data group "Online Corrections (ONL)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablOnl	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Online corrections (ONL)" does not exist
	1 (Yes)	Data group "Online corrections (ONL)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.2.7 Move to positive stop (MPS)

Move to positive stop (MPS)		EnablMps
Enables or disables the display of the data group "Move to Positive Stop (MPS)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablMps	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Move to positive stop (MPS)" does not exist
	1 (Yes)	Data group "Move to positive stop (MPS)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.2.8 Splines (SPL)

Splines (SPL)		EnablSpl
Enables or disables the display of the data group "Splines (SPL)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablSpl	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Splines (SPL)" does not exist
	1 (Yes)	Data group "Splines (SPL)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.2.9 Tangential tool control (TTL)

Tangential tool control (TTL)		EnablTtl
Enables or disables the display of the data group "Tangential Tool Control (TTL)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablTtl	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Tangential tool control (TTL)" does not exist
	1 (Yes)	Data group "Tangential tool control (TTL)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.2.10 Thread cutting (THRD)

Thread cutting (THRD)		EnablThrd
Enables or disables the display of the data group "Thread cutting (THRD)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablThrd	

Optional parameter setup (SUP) - Setup of configuration data –

Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Threading (THRD)" does not exist
	1 (Yes)	Data group "Threading (THRD)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.2.11 Tool and workpiece management (TLWP)

Tool and workpiece management (TLWP)		EnablTlwp
Enables or disables the display of the data group "Tool and Workpiece Management (TLWP)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablTlwp	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Tool and Workpiece Management (TLWP)" does not exist
	1 (Yes)	Data group "Tool and Workpiece Management (TLWP)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.2.12 Transformation (TRA)

Transformation (TRA)		EnablTra
Enables or disables the display of the data group "Transformation (TRA)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablTra	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	

Description of parameter values	0 (No)	Data group "Transformation (TRA)" does not exist
	1 (Yes)	Data group "Transformation (TRA)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.2.13 Velocity-dependent output (VDO)

Velocity-dependent output (VDO)		EnablVdo
Enables or disables the display of the data group "Velocity-dependent Output (VDO)" in the "Optional parameters" function group.		
Path	SUP/FuncGr[1]/EnablVdo	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "Velocity-dependent output (VDO)" does not exist
	1 (Yes)	Data group "Velocity-dependent output (VDO)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

5.3 OEM-specific parameters

5.3.1 Prerequisites

Machine manufacturers equipped with client development facilities may use the NC Configurator to display and edit self-defined MACODA machine parameters. This requires adjustments in the file system of the NC kernel as well as in the installation directory of the IndraWorks GUI. This is only a summary of the steps required. More detailed information is given in a separate description if required.

The following additional data are required in the NC kernel:

- XML schema file "cfg_oem.xsd"

For describing the structure, the data types, the value ranges as well as other attributes of the OEM-specific configuration data. It is expected to be in the path "root/usrfep/schemas". A template can be found in "root/feprom/schemas".

- XML file "cfg_oem.def.gz"

Optional parameter setup (SUP) - Setup of configuration data –

This file is to contain the default values for the data described in the schema file `cfg_oem.xsd`. It is required by the NC Configurator, such as when an element is to be set to its default value. This file, compressed with Gnu-Zip, is expected to be in the path `"root/usrfep/cfg"`. A template can be found in `"root/feprom/cfg"`.

- Conversion file `"oemcvini.dat"`

For each OEM-specific MACODA parameter, this file contains the information concerning the conversion to the data described in the schema file `cfg_oem.xsd`. It is expected to be in the path `"root/usrfep"`. A template can be found in `"root/feprom"`.

In the installation directory of the IndraWorks GUI, only text files need to be extended:

- Language-dependent text files `"cfg_oem_de.txt"`, `cfg_oem_en.txt"`, `"cfg_oem_xx.txt"`:

For all data described in the schema file `cfg_oem.xsd` (nodes, sheet elements), the corresponding texts contained in these text files will be allocated to the text numbers defined there. The files already exist as templates in the path `"\Rexroth\IndraWorks\Config\Mtx\CfgData"` and can be extended by the OEM.

5.3.2 OEM data group (OEM)

OEM data group (OEM)		EnablOem
Enables or disables the display of the data group "OEM Data Group (OEM)" in the "OEM-specific parameters" function group.		
Path	SUP/FuncGr[2]/EnablOem	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	-
Unit	-	
Description of parameter values	0 (No)	Data group "OEM data group (OEM)" does not exist
	1 (Yes)	Data group "OEM data group (OEM)" exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	-	
Dependency	-	
Macoda ID	-	
Data type	byte	

6 Basic NC configuration (MAIN) - Configuration data setup -

6.1 General information

The basic NC configuration settings are made in the MAIN data group:

- [Maximum number of channels, page 47](#)
- [Number of channels, page 48](#)
- [Number of drives, page 48](#)
- [Number of drives, page 49](#)
- [Maximum number of spindles, page 52](#)
- [Selection of the drive function type \(axes, spindles with/without C-axis drive\) , page 53](#)
- [Drive parameterization, page 54](#)
- [NC-controlled drive, page 55](#)
- [Axis function of the NC-controlled drive, page 59](#)
- [Spindle function of the NC-controlled drive, page 61](#)
- [Drive system functions \(AxFunc\), page 59 and Spindle system functions \(SpFunc\), page 61](#)
- [Sercos system functions \(SCS\), page 62](#)

6.2 Maximum number of channels

6.2.1 General information

The MTX manages up to 60 machining channels (Ch[1..60]).

6.2.2 Maximum number of channels (startup.xml)

Maximum number of channels (startup.xml)	MaxNofCh

Specifies the maximum possible number of channels in the MTX that can be triggered simultaneously.

A changed value is automatically saved in the startup.xml file in the root directory or in the /userfep directory. Subsequently, restart the system to apply the changes. Only then, a larger actual number of channels than 12 can be configured using for example parameter MAIN/NofCh.

After each NC restart, the value of MaxNofCh active in the control is displayed. This value, however, can deviate from the previously entered value due to internal limits.

Path	MAIN/MaxNofCh	
Value range	1..60	
Pre-assignment	12	
Unit	-	
Description of parameter values	Maximum possible number of channels	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Activation	System restart (CS)
Dependency	Limits the MAIN/NofCh parameter
Macoda ID	None A changed value is saved in the startup.xml file.
Data type	Ch1_n_t (byte) 1 byte integer, unsigned

6.3 Number of channels

6.3.1 General information

The MTX manages up to 60 machining channels (Ch[1..60]).

6.3.2 Number of channels

Number of channels		NofCh
Specifies the number of channels in the MTX that can be triggered simultaneously. The minimum is 1 channel.		
According to the value specified here, the control generates the necessary number of channel interfaces in the PLC interface.		
Path	MAIN/NofCh	
Value range	1..60	
If more than 12 drives are to be actuated, adjust the <MaxNofCh> setting in the startup.xml file first. The file startup.xml is ex in the system search path (root directory → /usrfep → /feprom). A template is contained in the /feprom directory.		
Optionally, the MaxNofCh parameter can be changed in the NC Configurator.		
In any case, trigger a control startup to apply changes of the MaxNofCh setting.		
Pre-assignment	1	
Unit	-	
Description of parameter values	Applied number of machining channels.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	9040 00001	
(MACODA compatibility: Machine parameter 9040 00001 is greater than NofCh by 1 because it also counts one internal auxiliary channel. The MTX uses the auxiliary channel for internal control of machine functions.		
Data type	Ch1_n_t (byte) 1 byte integer, unsigned	

6.4 Maximum number of drives

6.4.1 General information

The MTX can actuate up to 250 drives. This number includes all physical axes, spindles and spindles with C-axis operation.

6.4.2 Maximum number of NC-controlled drives (startup.xml)

	Maximum number of NC-contr. drives (startup.xml)	MaxNofDr
	Specifies the maximum possible system drive index and thus the maximum number of NC-controlled drives used at the machine for this application.	
	A changed value is automatically saved in the startup.xml file in the root directory or in the /userfep directory. Subsequently, restart the system to apply the changes. Only then, a larger actual number of drives than 64 can be configured using for example parameter MAIN/NofDr.	
	After each NC restart, the value of MaxNofDr active in the control is displayed. This value, however, can deviate from the previously entered value due to internal limits.	
Path	MAIN/MaxNofDr	
Value range	0..250	
Pre-assignment	64	
Unit	-	
Description of parameter values	Maximum possible number of drives	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	Limits the MAIN/NofDr parameter	
Macoda ID	None A changed value is saved in the startup.xml file.	
Data type	Dr_t (byte) 1 byte integer, unsigned	

6.5 Number of drives

6.5.1 General information

The MTX can actuate up to 250 drives. This number includes all physical axes, spindles and spindles with C-axis operation.



The number of spindles and spindles with C-axis operation is not specified explicitly. Instead, they are defined by defining the drive function type of a drive Dr[i] if the property "Spindle" or "Spindle with C-axis drive" is set for one or more drives Dr[i] (see [chapter 6.7 "Selection of drive function types" on page 53](#)).

The system drive index "i" of the drives Dr[i] extends from 1 to 250. There is no index "0" – it is equal to "no drive" at the MTX!

The following special features apply to the display of the drives in the NC configurator:

- *The drive nodes are described (.../Dr[i])*
 - up to the last activated drive described in the parameter /MAIN/Dr[i]/EnablDr (equivalent with the drive index which is specified by means of the parameter /MAIN/NofDr)
 - and furthermore up to the last drive for which a system drive name is defined in the parameter /MAIN/Dr[i]/SysDrName...

Basic NC configuration (MAIN) - Configuration data setup -

- and at least up to the drive index which is specified by means of the parameter /MAIN/MinNofDrNodes.
- The adding of a drive with the function "New" from the context menu MAIN always generates a new drive node behind the last **described** drive node and assigns the drive functionality selected in the context menu (axis, spindle, spindle/C-axis) to the node.
- When deleting a drive with the function "Delete" from the context menu of the node MAIN always deletes the last **described** drive node. Besides the parameter /MAIN/NofDrNodes, the parameters Parameter /MAIN/NofDr and /MAIN/MinNofDrNodes are decremented. Furthermore, a system drive name defined in the parameter /MAIN/Dr[i]/SysDrName is automatically deleted.
- *The adding/deleting of drives through modification of the parameter /MAIN/NofDr has the following effect:*
 - An increase of /MAIN/NofDr activates all drives after the last activated drive up to the new index and assigns axis functionality to the drives.
 - A reduction of /MAIN/NofDr deactivates all drives, defined after the new index.

6.5.2 Number of NC-controlled drives

Number of NC-controlled drives	NofDr

Specifies the highest active system drive index and thus the maximum number of the NC-controlled drives used at the machine for this application.

When "NofDr" **increases**, all drives above the last active drive index are activated up to the index $i = \text{NofDr}$. In doing so, the drive function type "Axis" (rotary/linear axis) will be set.

It is possible to individually adapt the drive function type of individual drives in the context menu of MAIN/Dr[i] by selecting "Convert to...".

Besides the absolute number of drives, it is also possible to **increase** the number by "one drive" using the context menu in the MAIN/ folder:

1. Select the **MAIN** folder and right click to select the item with the function "New" from the context menu.

The following entries appear (with the new drive index), corresponding to the selectable drive function types:

- AxFunc[max.Index_{alt} + 1] and
- SpFunc[max.Index_{alt} + 1] and
- AxFunc[max.Index_{alt} + 1] / SpFunc[max.Index_{alt} + 1]

When the "NofDr" **decreases**, all drives from the last drive index and down to index $i = \text{NofDr}$ will be deactivated.



Existing data for the drives to be deleted can be lost during a Configurator session. Back up the data before making any changes!

Besides the absolute number of drives, it is also possible to **decrease** the number by "one drive" from the context menu in the MAIN/ folder:

1. Select the **MAIN** folder and right click to select the entry with the function "Delete" from the context menu:

Dr[max.Index_{alt}]

Path	MAIN/NofDr	
Value range	0..250 (Dr_{max})	
	If more than 64 drives are to be actuated, adjust the <MaxNofDr> setting in the startup.xml file first. The file startup.xml is ex in the system search path (root directory → /usrfep → /feprom). A template is contained in the /feprom directory. Optionally, the MaxNofDr parameter can be changed in the NC Configurator. In any case, trigger a control startup to apply changes of the MaxNofDr setting.	
Pre-assignment	4	
Unit	-	
Description of parameter values	0	No drive applied.
	1..250	Highest active system drive index for this application
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	Modification of NofDr might cause a modification of the NofDrNodes.	
Macoda ID	1001 00001	
Data type	Dr_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

6.5.3 Minimum number of configured drives

Minimum number of configured drives		MinNofDrNodes
Defines the minimum number of the configurable drives displayed in the NC configurator. Less drive nodes than given in the parameter are never displayed. More drive nodes are displayed in case a drive with a higher index is activated (parameter /MAIN/Dr[i]/EnablDr = yes) or provided with a system drive name (parameter /MAIN/Dr[i]/SysDrName <> empty string). The functions "New" and "Delete" from the context menu of the folder MAIN for inserting or deleting a drive always refers to the latest drive displayed.		
Path	MAIN/MinNofDrNodes	
Value range	0..250 (Dr_{max})	
Pre-assignment	0	
Unit	-	
Description of parameter values	Minimum number of the drive nodes displayed in the NC configurator	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	Immediately (IM)	
Dependency	A modification of MinNofDrNodes might cause a modification of the NofDrNodes.	
Macoda ID	6001 00001	

Data type Dr_t (byte)
1 byte integer, unsigned for MACODA compatibility

6.5.4 Number of configured drives

Number of configured drives	NofDrNodes

Provides the number of the configurable drives displayed in the NC configurator. The read-only-parameter was calculated automatically by the control and cannot be modified.

The number of the drive nodes displayed depends on several conditions.

The drive nodes are described (.../Dr[i])

- up to the last activated drive described in the parameter /MAIN/Dr[i]/EnablDr (equivalent to the drive index which is specified by the parameter /MAIN/NofDr) ...
- and up to the last drive for that a system drive name is defined by the parameter /MAIN/Dr[i]/SysDrName ...
- and at least up to the drive index which is specified by the parameter /MAIN/NofDr).

Path MAIN/NofDrNodes

Value range 0..250 (Dr_{max})

Pre-assignment 4

Unit -

Description of parameter values Number of the drive nodes displayed in the NC configurator

Access rights
MTB: Read
Setter: Read
Operator: Read

Activation None (IM)

Dependency The read-only-parameter was calculated automatically by the control. It depends on the following parameters (see explanations above as well):

- /MAIN/Dr[i]/EnablDr
- /MAIN/Dr[i]/SysDrName
- /MAIN/MinNofDrNodes

Macoda ID None

Data type Dr_t (byte)
1 byte integer, unsigned for MACODA compatibility

6.6 Maximum number of spindles

6.6.1 General information

The MTX can actuate up to 60 spindles and spindles with C-axis mode out of the total number of drives that can be actuated.

6.6.2 Maximum number of NC-controlled spindles (startup.xml)

Maximum number of NC-contr. spindles (startup.xml)	MaxNofSp

Defines the maximum number of NC-controlled spindles.

A changed value is automatically saved in the startup.xml file in the root directory or in the /userfep directory. Subsequently, restart the system to apply the changes. Only then, a larger actual number of spindles than 32 can be configured using the context menu of the main folder MAIN ("New") or of the drive folder MAIN/Dr[i] ("Change to").

After each NC restart, the value of MaxNofSp active in the control is displayed. This value, however, can deviate from the previously entered value due to internal limits.

Path	MAIN/MaxNofDr
Value range	0..60
Pre-assignment	32
Unit	-
Description of parameter values	Maximum possible number of spindles
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	None A changed value is saved in the startup.xml file.
Data type	Sp_t (byte) 1 byte integer, unsigned

6.7 Selection of drive function types

The **function type** of a drive Dr[i] is organized as follows:

- Linear/rotary axis
- Spindle
- Spindle with C-axis operation



A distinction is made between "linear axes" and "rotary axes" in the respective work folder under MAIN/Dr[i]/AxFunc/SysAxType (see [chapter 6.10 Drive System Functions \(AxFunc\)](#), page 59).

Spindles and spindles with C-axis operation and their importance as "system spindles" are described in greater detail in [9 Spindles \(SP\) - Basic Parameters](#), page 115.

The drive function type is selected from the context menu of the main folder MAIN and/or from the context menu MAIN/Dr[i]:

MAIN Context Menu

The context menu of folder **MAIN** allows you to create new drives Dr[i] in the parameter list and to select the drive function type straight away. With the function "**New**" in the context menu, a new drive folder Dr[i+1] can be inserted behind the last drive folder MAIN/Dr[i] in the MAIN/ folder, whereby the drive function type is also selected at the same time:

- AxFunc[i+1] → New drive moves a linear or rotary axis
- SpFunc[i+1] → New drive moves a spindle
- AxFunc[i+1]/SpFunc[i+1] → New drive moves a spindle which can be switched to C-axis mode

MAIN/Dr[i] context menu

The function "Delete" allows you to disable the last drive visible in the parameters list and removes the last drive Dr[i] in the drive folder MAIN/Dr[i].

The context menu of the drive folder **MAIN/Dr[i]** allows to change the drive function type using **Convert to...** (see above). Only the drive function types that are not the currently selected drive are ever shown (e.g.: drive Dr[i] is a linear or rotary axis (AxFunc[i+1]), so the alternatives "SpFunc[i+1]" and "AxFunc[i+1]/SpFunc[i+1]" are shown in the context menu).

Example:

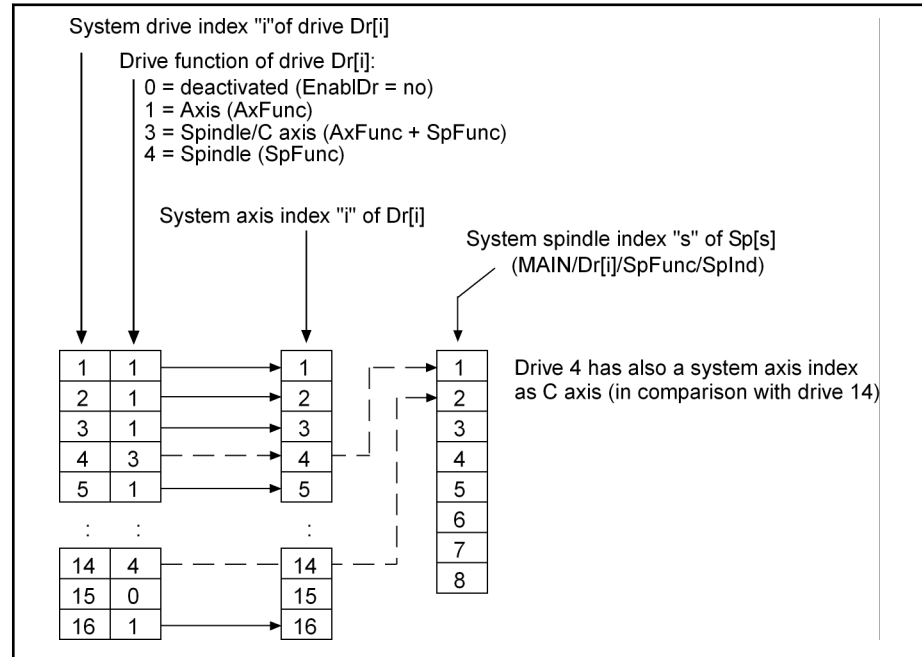
Drive function types and drive indices

Fig. 6-1: Definition of drive function types and drive indices

6.8 Drive parameterization

6.8.1 General information

Every drive Dr[i] is first described by the parameters from [chapter 6.9 "NC-controlled drive \(Dr\[i\]\)" on page 55](#) (NC-controlled drive (Dr[i])). Depending on the selection of the drive function type ([chapter 6.7 "Selection of drive function types" on page 53](#)), **different** parameter groups appear in addition to these parameters:

- If the drive is defined as a **linear or rotary axis**, **only** the parameter group AxFunc ([chapter 6.10 "Drive system functions \(AxFunc\)" on page 59](#)) and Scs (Sercos interface ([chapter 6.12 "Sercos system functions \(Scs\)" on page 62](#))) is displayed under the drive folder Dr[i].
- If the drive is defined as a **spindle with C-axis operation**, it is edited under the drive folder Dr[i] with the parameter group AxFunc ([chapter 6.10 "Drive system functions \(AxFunc\)" on page 59](#)) and SpFunc ([chapter 6.11 "Spindle system functions \(SpFunc\)" on page 61](#)) and the parameter group Scs Sercos interface ([chapter 6.12 "Sercos system functions \(Scs\)" on page 62](#)).
- If the drive is defined as a **spindle**, **only** the parameter group SpFunc ([chapter 6.11 "Spindle system functions \(SpFunc\)" on page 61](#)) and

Scs (Sercos interface ([chapter 6.12 "Sercos system functions \(Scs\)" on page 62](#))) is displayed under the drive folder Dr[i].

6.9 NC-controlled drive (Dr[i])

6.9.1 Physical drive name

Physical drive name	SysDrName

Specifies a system-wide, unique, channel-independent **system name** for each drive Dr[i].

If the drive is defined as a "spindle with C-axis operation", specify the name of the **c-axis**.



- The system names can be overridden by the names of the channel axes
[(see CHAN/Ch[k]/Coord/Acs/ChAx[1..8]/ChAxName (axis name)].
- The drives are displayed in the NC configurator at least up to the last drive that was provided with a system drive name (not equal to empty string).
Do not select a maximum number of NC-controlled drives (MaxNofDr) that is smaller than the last axis index with defined system drive name.

Path	MAIN/Dr[i]/SysDrName	
Value range	{Text}	
Pre-assignment	X	MAIN/Dr[1]/SysDrName
	Y	MAIN/Dr[2]/SysDrName
	Z	MAIN/Dr[3]/SysDrName
	No entry	MAIN/Dr[4..250]/SysDrName
Unit	-	
Description of parameter values	Characters	e.g.: X System name (name with letters only). Letters denoting an internal NC function (e.g. D, G, H, R, S, F, A, FA) are not allowed here. See "MTX 15VRS Programming Manual" (DOK-MTX***-NC**PRO*V15-RERS-EN-P).
	Characters + digits	e.g.: X2 System name (name with letters and one numerical sequence). To distinguish between the axis name and the following position data, write either "=" or a "blank" between them (X2=111.22 or X2 111.22).

NOTICE**Danger of damage to the workpiece or the machine!**

If a short axis name is part of a longer axis name (for example: Axes "X" and "X2") and if a decimal point is programmed after that, the longer axis name is always applied:

X2.5 → Axis X2 travels to 0.5

If, on the other hand, you want axis X to travel to position 2.5, you must enter either "=" or a "blank" as delimitation:

X=2.5 or X 2.5 → Axis X travels to 2.5

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00001	
Data type	Str8_NoUmlaut_t (isoLatin1String)	
	8-byte string, no umlauts for MACODA compatibility	

6.9.2 Enable drive

Enable drive	EnablDr

Enables or disables the drive Dr[i]

This parameter allows defined drives to be shown or hidden. For example, on machines with reduced drive, corresponding drives Dr[i] can be given the value "0" in EnablDr (in folder MAIN/Dr[i]/EnablDr). Any further parameters referring to such "hidden" drives are automatically reset or also hidden by the MTX.

You can also hide spindles using parameter SpType in folder SP/SpFunc/Sp[1..60]/Base.

Path	MAIN/Dr[i]/EnablDr	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	
Unit	-	
Description of parameter values	0	No
		Drive Dr[i] is disabled
	1	Yes
		Drive Dr[i] is enabled
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	Changing EnablDr during the last drive or after the last activated drive might result in changes for NofDr and NofDrNodes.	

Macoda ID 1001 00001
Data type Dr_t (byte)
 1 byte integer, unsigned for MACODA compatibility

6.9.3 Enable Sercos communication

Enable Sercos communication		EnablScsCom
Enables or disables Sercos communication to the drive Dr[i].		
Parameter ID		
Path	MAIN/Dr[i]/EnablScsCom	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No
	No Sercos communication to drive Dr[i] established	
		The setting "0" allows drives that do not exist as hardware in the system to be "simulated".
	1	Yes
	Sercos communication to drive Dr[i] established	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1050 00002	
Data type	byte	
	1 byte integer, unsigned for MACODA compatibility	

6.9.4 Enable virtual mode

Enable Virtual Mode		EnablVirtMode
Specifies whether a drive Dr[i] is a virtual drive or not.		
A virtual drive can, for example, be used as a master for an axis coupling (cam disk).		
Path	MAIN/Dr[i]/EnablVirtMode	
Value range	0	No
	1	Yes
Pre-assignment	0 (No)	
Unit	-	
Description of parameter values	0	No
	Normal operation	
	1	Yes
Disables the drive in the Sercos ring and sets the "axis disabled" or "spindle disabled" state.		

Basic NC configuration (MAIN) - Configuration data setup -

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1001 00010	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

6.9.5 Drive without command value input

Drive without command value input	SupprCmdValOut

Specifies for a drive Dr[i] (axis, spindle or spindle with C-axis operation),

- that no command values (S-0-0047, S-0-0036) shall be sent to the connected drive Dr[i], and
- that the Sercos main operation mode (S-0-0032) can be parameterized arbitrarily.

The following conditions apply:

- A drive Dr[i] without specified command value is always an axis or a spindle.
- Any movement preset from a part program or from the PLC is prohibited (runtime error).
- A drive without specified command value cannot be a member of an "axis coupling" or of a "gantry group".

Path	MAIN/Dr[i]/SupprCmdValOut	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Command values are specified for the drive Dr[i] (normal operation)
	1	Yes
		No command value is specified for the drive Dr[i].
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1001 00011	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

6.9.6 Delayed reaction in case of non-fatal drive errors

Delayed reaction in case of non-fatal drive errors	DelayedErrorReaction

Enables a delayed error reaction of the NC to non-fatal drive errors.

In case of some applications, it is required that the NC still controls the traversing profile of the axes and sets the axes to a coordinated standstill in order to avoid machine damages in case of error. An NC reaction in case of error is only possible for non-fatal errors (diagnostics F2xxx). Otherwise, the drive always reacts immediately with an error reaction. For more information on the use of the function, refer to the manual "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P), chapter "Delayed NC error reaction for non-fatal drive errors".

Path	MAIN/Dr[i]/DelayedErrorReaction	
Value range	0	Off
	1	Programmable
	2	ON
Pre-assignment	0	Off
Unit	-	
Description of parameter values	0	Off
		The delayed error reaction is switched off.
	1	Programmable
		The delayed error reaction can be switched on by the NC function "DERDelay".
	2	ON
		The delayed error reaction is switched on.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1001 00012	
Data type	Byte0_2_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

6.10 Drive system functions (AxFunc)

6.10.1 General information

For each drive Dr[i] being a linear or rotary axis (e.g. synchronous machining axes, C-axis function of a spindle), the following is defined:

- channel assignments and
- axis types

6.10.2 Axis type

Axis type		SysAxType
Specifies whether each drive Dr[i] is a "rotary axis" or a "linear axis".		
Path	MAIN/Dr[i]/AxFunc/SysAxType	
Value range	0	-
	1	Linear axis
	2	Rotary axis

Basic NC configuration (MAIN) - Configuration data setup -

	(1 .. 4)	For MACODA compatibility (values only apply for the internal representation of the MACODA parameter 1003 00004 "Axis motion type", which, compared to "SysAxType ", contains additional information on whether it is a modulo axis)
Pre-assignment	1	Linear axis
Unit	-	
Description of parameter values	0	-
	1	Linear axis Drive is a linear axis. Linear axes can be moved within their respective travel limit range only and are programmed in "mm" or "inches".
	2	Rotary axis Drive is a rotary axis. Rotary axes can be moved within their respective travel limit range only and are programmed in "degrees".
	(1 .. 4)	for MACODA compatibility: Axis motion type (refer to the notes under "Value range") 1 = linear axis without modulo calculation, 2 = rotary axis with modulo calculation, 3 = rotary axis without modulo calculation, 4 = linear axis with modulo calculation,
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	The following dependencies exist of the parameter EnablModCalc (path: AX/Dr[i]/PosLog):	
	- When switching the axis type from rotary to linear axis, the modulo operation is disabled automatically (EnablModCalc = 0). - When switching the axis type from rotary to linear axis, the modulo operation is enabled automatically (EnablModCalc = 1). - When switching the axis type to "-" (not defined), this change is not accepted if modulo operation is enabled (EnablModCalc = 1).	
Macoda ID	1003 00004	
Data type	Byte0_2_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

6.10.3 Channel assignment

Channel assignment	DefaultCh
<i>Specifies for each drive Dr[i],</i> - whether it is a machining axis (synchronous axis) or an auxiliary axis (asynchronous axis). - which channel (DefaultCh) a machining axis should be assigned to.	
 A maximum of 8 axes may be assigned to each channel.	



One existing machining channel is assigned to one active machining axis.

The channel assignment of a machining axis can be manipulated in the part program or in "manual data input" mode using the "GetAxis" function

GetAxis (GAX).

Path	MAIN/Dr[i]/AxFunc/DefaultCh	
Value range	0 (asynchronous axis) 1..60	
Pre-assignment	1	MAIN/Dr[1..3]/AxFunc/DefaultCh
	0	MAIN/Dr[4..250]/AxFunc/DefaultCh
Unit	-	
Description of parameter values	0	Asynchronous axis Drive Dr[i] is an auxiliary axis . Auxiliary axes can be activated from any channel. Auxiliary axes cannot be interpolated with other axes.
	1..60	Drive Dr[i] is a machining axis and is assigned to the specified channel "DefaultCh" = 1-60. As a rule, machining axes can be activated only by those part programs which are assigned to the channel specified here. The same applies to MDIs.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00002	
Data type	Ch_t (byte) 1 byte integer, unsigned for MACODA compatibility	

6.11 Spindle system functions (SpFunc)

6.11.1 General information

The system-wide spindle assignment for spindles with/without C-axis operation is set with respect to a Dr[i] drive.



The parameter group "Spindle" SpFunc is displayed if the drive function type for the folder MAIN/Dr[i] has been set to "SpFunc" (spindles without C-axis operation) or to "AxFunc/SpFunc" (spindles with C-axis operation).

6.11.2 Spindle index

Spindle index	Spind

Defines a system-wide system spindle index for each drive with spindle function (spindle or spindle/C-axis). The spindle-specific parameters of this drive are found under this index in the "Spindles (SP)" data group.

Basic NC configuration (MAIN) - Configuration data setup -

		This index is, for example, relevant for the assignment of machine parameters, interfaces, NC programming commands, etc.
Path	MAIN/Dr[i]/SpFunc/SpInd	
Value range	0..60	
Pre-assignment	1	
Unit	-	
Description of parameter values	0	No system spindle index could be assigned to the drive Dr[i].
	1..60	System spindle index (system-wide) for drive Dr[i]. Only 60 system spindles are allowed.
Access rights	Engineer:	Read
	Service:	Read
	MTB:	Read
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1001 00001	
Data type	Sp_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

6.12 Sercos system functions (Scs)

6.12.1 General information

Sercos-specific data (Sercos address and drive data) are defined for the drive Dr[i] (independent of drive function type!).

6.12.2 Sercos address

Sercos address		ScsAddr
Defines for drive Dr[i] which Sercos address is set on the drive Dr[i] designated for this.		
Each Sercos address (1...255) may be assigned only once for each ring. The value entered here must be identical with the value set in the drive module.		
Path	MAIN/Dr[i]/Scs/ScsAddr	
Value range	0..255	
Pre-assignment	1	MAIN/Dr[1]/Scs/ScsAddr
	250	MAIN/Dr[250]/Scs/ScsAddr
Unit	-	
Description of parameter values	0	No Sercos address defined
	1..255	Sercos address of drive Dr[i]
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	

Dependency	-
Macoda ID	1050 00004
Data type	unsignedByte
	1 byte integer, unsigned for MACODA compatibility

6.12.3 Drive type

Drive type		DrType
	Specifies the series and manufacturer of each drive Dr[i].	
Path	MAIN/Dr[i]/Scs/DrType	
Value range	0	Servodyn-D
	1	DIAX, ECODrive
	2	IndraDrive, HNC100-3X (HydraulicDrive)
	3	Sercos standard, functions defined with S parameters
	4	HNC100-2X (Rexroth hydraulic axis)
	5	REFU spindle
	6	Hitachi
Pre-assignment	2 (IndraDrive, HNC100-3X)	
Unit	-	
Description of parameter values	Drive type (series and manufacturer): See value range for meaning.	
	 The in-position logic of a drive depends on the drive type. Also observe NCO/Dr[i]/InPosRough (Inpos window, rough).	
	 The drive type HNC100-3X , representative of the current hydraulic drives of type HydraulicDrive , is treated as the drive type IndraDrive in the control. Please note that it a HydraulicDrive does not support all IndraDrive functions.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1050 00001	
Data type	Byte0_6_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

7 Channels (CHAN) - Basic parameters -

7.1 General information

In the channel folder CHAN of the "basic parameters", channel-specific settings for each channel Ch[k=1-60] defined in [Chapter 6 NC basic configuration \(MAIN\) setup of the configuration data, page 47](#) are considered:

- [Initialization of the channel state, page 65](#)
- [Path data in the channel, page 69](#)
- [Coordinate systems and their allocation to the channel axes, page 79](#)
- [Coupling spindle/channel axes, page 85](#)

7.2 Initialization of the channel status (Ini)

7.2.1 State after startup (switched-on state)

	Status after startup	NcResetState
	Specifies which NC functions should be effective after "switch on" or "reset" of the control.	
Path	CHAN/Ch[k]/Ini/NcResetState	
Value range	{NC syntax}	
Pre-assignment	30 blanks	
Unit	-	
Description of parameter values	Specification of NC functions (switch on string max. 240 characters)	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7060 00010	
Data type	Str240_NoUmlaut_t (isoLatin1String)	
	30 * 8 byte string for MACODA compatibility	

7.2.2 Status after channel reset

	State after channel reset	ChResetState
	Specifies which NC functions are applied after "M30", "channel reset" or "system reset for all channels" and allows the deactivation of init programs if required.	
	The following functions are enabled after	
	• M30:	all functions before the entry "#Reset:".
	• Channel reset:	all functions before the entry "#SysRes:"
	• System reset for all channels:	all functions

It is additionally possible to impede the init program execution. The keyword "#NoIniPrg" can be entered between the beginning of the string and the "#Reset:" entry.

Channels (CHAN) - Basic parameters -

If no "#NoIniPrg" exists, the control additionally calls the parameterized init program after the corresponding system event and the activation of the corresponding NC functions entered here. Further activations can be made in this ini program (see [chapter 7.2.3 "File name of the channel initialization program \(init program\)"](#) on page 66).

Path	CHAN/Ch[k]/Ini/ChResetState
Value range	{NC syntax}
Pre-assignment	Channel 1: G62 G48 G40 G140 G53 G153 BCR() D0 TRS() ATR() KVP() FFW() SHT() SCL() MIR() PLS() RND() ROT() G80 CRD()#Reset: CTA()#SysRes:DAX Channels 2..60: #Reset: CTA()#SysRes:
Unit	-
Description of parameter values	Specification of NC functions (string according to above-mentioned conditions, max. 240 characters)
	 When strings exclusively consisting of alphabetic characters are entered, they must be followed by a blank, e.g.: DIAOVE (= "DIA OVE").
	 For MTX13V04 and higher, the pre-assignment was significantly reduced for channel 1. Multiple NC-functions were applied to the preset init program for channel 1. Refer to the manual "MTX 15VRS Functional Description - Basics" (DOK-MTX***-NC*F*BA*V15-RERS-EN-P) in the chapter "Channel initialization program (init program)" for information which NC function have to be contained in parameter "ChReset-State" and which statements can be contained in the init program. After reading in the control archives created before the MTX13V04, settings for this parameter were not superimposed by init programs. In this case, only init programs not containing any NC functions were called.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7060 00020
Data type	Str240_NoUmlaut_t (isoLatin1String) 30 * 8 byte string for MACODA compatibility

7.2.3 File name of the channel initialization program (init program)

File name of the channel initialization program	ChIniProgFile

Specifies which init programs are to be called after "M30", "channel reset" or "system reset for all channels". Init programs supplement settings made in the parameter "ChResetState" (see [chapter 7.2.2 "Status after channel reset"](#) on page 65).

The file name has to be entered with its extension, but without directory path. The control searches in the specified file in the system search path (root directory → /usrfep → /feprom).

If no initialization program is to be executed in the respective channel, proceed as follows:

- Enter an empty string into this parameter or
- the keyword "#NoIniPrg" into the parameter "ChResetState".

Path	CHAN/Ch[k]/Ini/ChIniProgFile
Value range	{File name syntax without path}
Pre-assignment	Ch01Ini.prg .../Ch[1]/Ini/ChIniProgFile Ch02Ini.prg .../Ch[2]/Ini/ChIniProgFile Ch03Ini.prg .../Ch[3]/Ini/ChIniProgFile Ch04Ini.prg .../Ch[4]/Ini/ChIniProgFile Ch05Ini.prg .../Ch[5]/Ini/ChIniProgFile Ch06Ini.prg .../Ch[6]/Ini/ChIniProgFile Ch07Ini.prg .../Ch[7]/Ini/ChIniProgFile Ch08Ini.prg .../Ch[8]/Ini/ChIniProgFile Ch09Ini.prg .../Ch[9]/Ini/ChIniProgFile Ch10Ini.prg .../Ch[10]/Ini/ChIniProgFile Ch11Ini.prg .../Ch[11]/Ini/ChIniProgFile Ch12Ini.prg .../Ch[12]/Ini/ChIniProgFile <EmptyString> .../Ch[13..60]/Ini/ChIniProgFile
Unit	-
Description of parameter values	File name for init program (incl. file name extension, no path specification)  When creating a control archive with machine parameters, the parameterized init programs for the parameterized machining signals are automatically backed up as well if available in the system search path (without /feprom). It is also backed up if the keyword "#NoIniPrg" is entered into the "ChResetState" parameter of the corresponding channel. Analogously, when reading in a control archive with machine parameters, the init programs for the parameterized machining channels are automatically read in.  Refer to the manual "MTX 15VRS Functional Description - Basics" (DOK-MTX***-NC*F*BA*V15-RERS-EN-P) in the chapter "Channel initialization program (init program)" for information which NC function have to be contained in parameter "ChResetState" and which statements can be contained in the init program.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7060 00030

Channels (CHAN) - Basic parameters -

Data type Str28_NcProgName_t (isoLatin1String)
4 * 8 byte string for MACODA compatibility

7.2.4 Delete SetPos Offset with a control reset

	SetPos Offset	DelZpo
	Specifies whether a zero point set with "SetPos" should be deleted upon "re-set".	
Path	CHAN/Ch[k]/Ini/DelZpo	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No
	Do not delete SetPos.	
	<i>Set zero points are deleted only if SetPos</i>	
	- has been programmed in the part program / by manual data input, or - has been entered in CHAN/Ch[k]/Ini/ChResetState (state after channel reset) and is triggered by a reset.	
	1	Yes
	Delete SetPos	
	It is not necessary to enter SetPos into the parameter "Channel reset state" ChResetState (in the folder CHAN/Ch[k]/Ini/).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00510	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

7.2.5 Automatic program reselection

	Automatic program reselection	AutoProgSel
	Specifies whether the function "Automatic program reselection" should be enabled.	
Path	CHAN/Ch[k]/Ini/AutoProgSel	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
	No automatic program reselection	
	1	Yes

Reselection of the last program selected after

- enabling an automatic mode, or
- channel reset while automatic mode is enabled.

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7060 00410	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

7.3 Path data (Path)

7.3.1 Velocity data (Vel)

Maximum path velocity

Maximum path velocity		MaxChVel
Defines the maximum permissible feed value on a channel.		
This parameter is intended as protection against any F-value that has been programmed too high by mistake.		
Path	CHAN/Ch[k]/Path/Vel/MaxChVel	
Value range	0.000 .. 10000000000.000	
	0.000 .. 9999999999.999 for MACODA compatibility	
Pre-assignment	1000000.000	
Unit	mm/min	
Description of parameter values	Maximum permissible feed value	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7030 00120	
Data type	Vel_t (Double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

Reduced rapid traverse

Reduced rapid traverse		RedChVel
Specifies the velocity to which all rapid traverse movements of synchronous axes (G00) should be reduced if the channel interface signal qCh_RedRap "Reduced rapid traverse" is enabled.		
This has no impact on the spindle speed and the path feed on the respective channel Ch[k].		
Path	CHAN/Ch[k]/Path/Vel/RedChVel	
Value range	0.000 .. 10000000000.000	

Channels (CHAN) - Basic parameters -

	0.000 .. 99999.999 for MACODA compatibility	
Pre-assignment	0.000	
Unit	mm/min	
Description of parameter values	0.000	Reduced rapid traverse velocity is equal to the feed programmed last on the channel Ch[k].
	>0.000	Reduced rapid traverse
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7030 00110	
Data type	Vel_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

7.3.2 Feed override (Ovrd)

Maximum feed override

Maximum feed override		MaxChOvrd
	Specifies the maximum permissible feed override (as a %) on the channel Ch[k].	
	The feed override internally effective in the NC is always limited to this parameter value.	
Path	CHAN/Ch[k]/Path/Ovrd/MaxChOvrd	
Value range	0.00 .. 999.00	
Pre-assignment	120.00	
Unit	%	
Description of parameter values	Maximum permissible feed override	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7030 00012	
Data type	Ovr_t (float, 2 decimal places)	
	4-byte float, 2 decimal places for MACODA compatibility	

Feed override for step[1..32]

Feed override for step[1..32]	ChOvrdStep[1..32]
Specifies the values of the individual steps of the feed override on the channel Ch[k].	
Irrespective of the currently selected step, the maximum permissible override value is always adhered to within the NC according to CHAN/Ch[k]/Path/Ovrd/MaxChOvrd.	
Path	CHAN/Ch[k]/Path/Ovrd/ChOvrdStep[1..32]

Value range	0.00 .. 150.00	
Pre-assignment	0.00 .../ChOvrStep[1] 1.00 .../ChOvrStep[2] 2.00 .../ChOvrStep[3] 4.00 .../ChOvrStep[4] 6.00 .../ChOvrStep[5] 8.00 .../ChOvrStep[6] 10.00 .../ChOvrStep[7] 20.00 .../ChOvrStep[8] 30.00 .../ChOvrStep[9] 40.00 .../ChOvrStep[10] 50.00 .../ChOvrStep[11] 60.00 .../ChOvrStep[12] 70.00 .../ChOvrStep[13] 80.00 .../ChOvrStep[14] 100.00 .../ChOvrStep[15] 120.00 .../ChOvrStep[16] 0.00 .../ChOvrStep[17..32]	
	MACODA compatibility:	
	120.00 7030 00010 [1] (channel k), no relevance	
	ChOvrStep[n] 7030 00010 [n + 1] (channel k)	
Unit	%	
Description of parameter values	ChOvrStep[1..32]	Percentage values for a maximum of 32 override steps. Values can be freely selected and programmed up to 150%. Since the individual values do not have to be in ascending order, a Gray code-based override is possible as well. The override step to be active is selected via the channel-related interface signals "qCh_Override_00"- "qCh_Override_04", unless "0.00" is entered for all individual parameters [1] to [32]. In such a case, the control interprets the interface signals "qCh_Override_00"... "qCh_Override_15" directly as an override value scaled 0.01 %. This allows to set the override within the range of 0% to 655.35% in steps of 0.01%.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7030 00010	

(MACODA compatibility: In machine parameter 7030 00010, the indices are shifted by 1, i.e. ChOvrStep[1] has index 2.)

Data type Ovr_t (float, 2 decimal places)
4-byte float, 2 decimal places for MACODA compatibility

Example:

Enabling the various steps via channel-related interface (IF) signals:

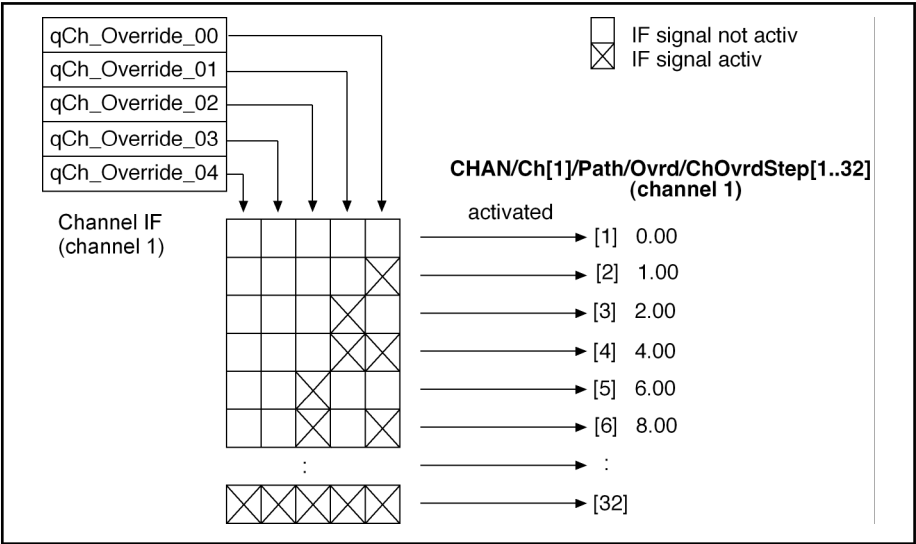


Fig. 7-1: Feed override (channel) and interface signals

7.3.3 Rapid traverse override (RapTravOvr)

Activating rapid traverse override

Enable rapid traverse override	EnablRapTravOvr

Allows the usage of a second override input as rapid traverse override.

The PLC interface was extended with a second override interface. A second override potentiometer (signals qCh_RapOverride_00..qCh_RapOverride_15) can be created in the channel interface MT_qCh. Additionally, the rapid traverse override can be set on 100% via the qCh_Rap100 signal.

The signals iCh_RapOverride0 and iCh_RapOverride100 signalizes in the channel output interface MT_iCh whether the active rapid traverse override is either 0% or 100%.

Path	CHAN/Ch[k]/Path/RapTravOvr/EnablRapTravOvr	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		The second override input (rapid traverse override) is switched off. The override input as usual is effective.
	1	Yes
		The second override input (rapid traverse override) is enabled. The NC switches automatically to the rapid traverse override during the interpolation in the rapid traverse. The

		signaling of the channel waiting state "Channel feed 0" is also effective for the rapid traverse override.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7030 00014	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

7.3.4 Channel SHAPE (SHAPE)

SHAPE order for jerk-limited path mode (G8)

SHAPE order for jerk-limited path mode (G8)		ChShapeG8
Specifies, for active jerk-limited velocity profile generation, into how many interpolation cycles a path acceleration jump is split up if G08 {SHAPE} has been programmed without SHAPE order.		
If the jerk limitation of the channel was enabled using the NC function "Jerk-Control(1)", this parameter is not important.		
Path	CHAN/Ch[k]/Path/Shape/ChShapeG8	
Value range	0 .. 100	
Pre-assignment	10	
Unit	-	
Description of parameter values	Number of interpolation cycles	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00320	
Data type	Byte0_100_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

SHAPE order for jerk-limited Positioning mode (G9)

SHAPE order for jerk-limited Positioning mode (G9)		ChShapeG9
Defines the path jerk for the path movement in positioning mode.		
The parameter is enabled by G9 (position mode with velocity 0 at block end) without further parameters.		
If the jerk limitation of the channel was enabled using the NC function "Jerk-Control(1)", this parameter is not important.		
Path	CHAN/Ch[k]/Path/Shape/ChShapeG9	
Value range	0 .. 100	
Pre-assignment	0	
Unit	-	

Channels (CHAN) - Basic parameters -

Description of parameter values 0, 1 The values 0 and 1 of the SHAPE order deactivate jerk limitation if the "minimum SHAPE order for jerk limitation" MinChShape (in folder CHAN/Ch[k]/Path/Shape/) is also deactivated.

The SHAPE order is calculated from the permitted path jerk with the following formula:

$$SHAPE\ order = \frac{Path\ acceleration[\frac{m}{s^2}]}{Path\ jerk[\frac{m}{s^3}] * Interpolation\ cycle\ time[\mu s] * 10^{-6}}$$

Fig. 7-2: Calculation of SHAPE order

>1 SHAPE order in positioning mode

Access rights **MTB:** Write
Setter: Read
Operator: Read

Activation System reset (SR)

Dependency -

Macoda ID 7050 00340

Data type Byte0_100_t (byte)
1 byte integer, unsigned for MACODA compatibility

Minimum SHAPE order for path jerk limitation (G8, G9)

Minimum SHAPE order for path jerk limitation (G8, G9)	MinChShape

Defines the maximum path jerk for the path movement.

It affects:

- G8: activation of the G8 default SHAPE (ChShapeG8)
- G8(SHAPE<Shape order>): Path SHAPE with velocity opt. Block transition
- G9: activation of the G9 default SHAPE (ChShapeG9)
- G9(SHAPE<Shape order>): Path SHAPE with positioning block
- G9(SIN<Shape order>): SIN² path SHAPE with positioning block

This parameter does not have any effect on the resulting jerk of the axis-specific jerk limitations, such as G9 (ASHAPE). Both the default path SHAPE orders from parameters ChShapeG8/ChShapeG9 and the programmed path SHAPE orders cannot fall below the minimum path SHAPE order.

If the jerk limitation of the channel was enabled using the NC function "Jerk-Control(1)", this parameter is not important.

Path CHAN/Ch[k]/Path/Shape/MinChShape

Value range 0 .. 100

Pre-assignment 0

Unit -

Description of parameter values 0, 1 Path SHAPE order 0 or 1 defines that no limitation is in effect.

The minimum path SHAPE order is calculated from the maximum **permissible** path jerk with the following formula:

$$SHAPE\ order = \frac{Path\ acceleration[\frac{m}{s^2}]}{Path\ jerk[\frac{m}{s^3}] * Interpolation\ cycle\ time[\mu s] * 10^{-6}}$$

Fig. 7-3: Calculation of SHAPE order

>1 SHAPE order for path jerk limitation (G8, G9)

Access rights **MTB:** Write
Setter: Read
Operator: Read

Activation System reset (SR)

Dependency -

Macoda ID 7050 00330

Data type Byte0_100_t (byte)

1 byte integer, unsigned for MACODA compatibility

Example:

SHAPE order for path jerk limitation (G8, G9)

With an interpolation cycle time of 4000 ms and a maximum path acceleration of 10 000 mm/s², a SHAPE order of 25 limits the path jerk to 100 m/s³.

7.3.5 Acceleration/brake data (Acc)

Maximum path acceleration

Maximum path acceleration		MaxChAcc
Defines the maximum positive acceleration permitted on the programmed path (contour).		
Path	CHAN/Ch[k]/Path/Acc/MaxChAcc	
Value range	0.000 .. 1000.000 0.001 .. 999.999 for MACODA compatibility	
Pre-assignment	10.000	
Unit	m/s ²	
Description of parameter values	Maximum permissible positive path acceleration	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7030 00210	
Data type	Acc_t (double, 3 decimal places)	
	4 byte float, 3 decimal places for MACODA compatibility	

Maximum path deceleration

Maximum path deceleration		MaxChDec
	Defines the maximum negative acceleration permitted on the programmed path (contour).	
Path	CHAN/Ch[k]/Path/Acc/MaxChDec	
Value range	0.000 .. 1000.000 0.001 .. 999.999 for MACODA compatibility	
Pre-assignment	10.000	
Unit	m/s ²	
Description of parameter values	Maximum permissible negative path acceleration	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7030 00220	
Data type	Acc_t (double, 3 decimal places) 4 byte float, 3 decimal places for MACODA compatibility	

7.3.6 Control of the axis jump (Jump)

Minimum angle for considering the axis jump

Minimum angle to consider the axis jump capability		MinAngVelJump
	Defines the minimum angle of a knee in contour which, when exceeded, causes the path velocity at a transition from one segment to the next to be reduced when G8 is active. The axis acceleration reserve AxAccReserve (in folder AX/Dr[i]/Acc) is to be considered. This parameter is only applicable if no jerk limitation is enabled in the respective channel. The control compares the transition angle at an identified knee in the contour with the value entered here. The tool path feed is not reduced for smaller transition angles provided that this is not prevented by the deceleration distance monitoring performed as part of the look-ahead function.	
	 The angle effective in the respective channel Ch[k] can be modified in the part program by programming "DefTangTrans" (DTT).	
Path	CHAN/Ch[k]/Path/Jump/MinAngVelJump	
Value range	0.000 .. 50.000	
Pre-assignment	0.000	
Unit	° (degrees)	
Description of parameter values	Transition angle of a knee in the contour for keeping within the set axis jump	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	

Dependency	-
Macoda ID	7030 00310
Data type	Double0_50_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

7.3.7 Kinematic limitation (KinLim)

Kinematic limitation level (presetting)

Kinematic limitation level (presetting)	KinLimLevel

In case of an active jerk limitation, it specifies the limitation of the maximum path velocity and path acceleration in order not to overload the participating axes.

Kinematic limitation level 0 ensures that axis velocity, axis acceleration and axis jerk remain for each axis during all traversing motions below the maximum values specified in AX/Dr[i]/Vel/MaxVel, AX/Dr[i]/Acc/FeedAxAcc or RapidAxAcc und AX/Dr[i]/Jerk/FeedAxJerk or RapidAxJerk . This limitation always expects the worst case. That is a maximum path velocity and a maximum path acceleration at the same time. Since this is rarely the case, the limitation is often too high and the travel times can be longer than expected.

An increasing kinematic limitation level results in higher values for the maximum path velocity and path acceleration and thus reduces the travel times. Due to this behavior, the maximum axis values allowed might be exceeded more often.

Generally, kinematic limitation level 3 can be recommended. If an application requires a smaller value, use programming JKC(KLL<value>).

Path	CHAN/Ch[k]/Path/KinLim/KinLimLevel
Value range	0 .. 3
Pre-assignment	0
Unit	-
Description of parameter values	Kinematic limitation level
Access rights	MTB: Write Setter: Write Operator: Read
Activation	System reset (SR)
Dependency	The parameter is only applied if the jerk limitation is active in the respective channel (activated by the NC functions JKC or JKC(1)).
Macoda ID	7030 00410
Data type	Byte0_3_t (byte) 1 byte integer, unsigned for MACODA compatibility

Kinematic limitation scaling (presetting)

Kinematic limitation scaling (presetting)	KinLimWeight

For each channel, this parameter specifies the importance of the maximum path velocity possible compared to the maximum path acceleration for the machining. The higher the parameter value, the more important is the maximum path velocity possible.

If the participating axes cannot apply the programmed values for the path velocity and the path acceleration, the axis limit values are not exceeded in

Channels (CHAN) - Basic parameters -

	many cases due to the reduction of the maximum path velocity as well as by the reduction of the maximum path acceleration.	
Path	CHAN/Ch[k]/Path/KinLim/KinLimWeight	
Value range	0.001 .. 1000.000	
Pre-assignment	1	
Unit	-	
Description of parameter values	< 1	The maximum possible path velocity is less important than the maximum path acceleration possible.
	1	The maximum possible path velocity is as important as the maximum possible path acceleration.
	>1	The maximum path velocity possible is more important than the maximum path acceleration possible.
Access rights	MTB:	Write
	Setter:	Write
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7030 00420	
Data type	Double0.001_1000_t	
	8 byte float, 3 decimal places for MACODA compatibility	

Kinematic limitation filter(presetting)

Kinematic limitation filter (presetting)		KinLimFilter
	The parameter specifies the SHAPE filter runtime for each channel in case of an active jerk limitation.	
Path	CHAN/Ch[k]/Path/KinLim/KinLimFilter	
Value range	0.0 .. 200.0	
Pre-assignment	0.0	
Unit	ms	
Description of parameter values	0.0	The runtime of the SHAPE filter is automatically generated from the ratio of the acceleration value to the jerk values of the channels axes when enabling the jerk limitation by the NC function JKC or JKC(1).
	>0.0	The parameter KinLimFilter specifies the runtime of the SHAPE filter when enabling the jerk limitation by the NC function JKC or JKC(1).
Access rights	MTB:	Write
	Setter:	Write
	Operator:	Read
Activation	System reset (SR)	
Dependency	- The parameter KinLimFilter is only active if the jerk limitation is active in the respective channel (activated by the NC functions JKC or JKC(1)). - In case of an interpolation cycle time under 1000 µs (SysCycTime), the maximum value of KinLimFilter is reduced to 200 cycles.	
	Example: For SysCycTime = 500 µs, the upper limit is 200 * 500 µs = 100000 µs = 100 ms	

Macoda ID 7030 00430
 Data type Float0_200_t
 4 byte float, 1 decimal place for MACODA compatibility

7.4 Coordinates (Coord)

7.4.1 Axis coordinate system (ACS)

General information

In every channel Ch[k], there can be up to 8 channel axes ChAx[1..8] defined that span the **axis coordinate system (ACS)**.

Axis name

Axis name	ChAxName

Defines the axis name of the channel Ch[k] in the active axis coordinate system for a channel axis ChAx[1..8].

The original physical drive names

(MAIN/Dr[i]/SysDrName) of the system axes are overridden by the "axis names" ChAxName of the channel axis. That way, different system axes can be addressed with the same axis name from different channels.

Path	CHAN/Ch[k]/Coord/Acs/ChAx[1..8]/ChAxName
Value range	{Text}
Pre-assignment	-
Unit	-
Description of parameter values	Characters Designation of the axis name: Identification using characters, e.g. X. Characters denoting an internal NC function (e.g. D, G, H, R, S, F, A, FA) are not allowed here. See "MTX 15VRS Programming Manual" (DOK-MTX***-NC**PRO*V15-RERS-EN-P), table of the Default programming formats. Characters + digits Designation of the axis name: Identification using characters or a sequence of number X2. To distinguish between the axis name and the following position data, write either "=" or a "blank".

NOTICE

Danger of damage to the workpiece or the machine!

If a short axis name is part of a longer axis name (for example: Axes "X" and "X2") and if a decimal point is programmed after that, the longer axis name is always effective:

X2.5 → Axis X2 travels to 0.5

If, on the other hand, you want axis X to travel to position 2.5, you must enter either "=" or a "blank" as delimitation:

X=2.5 or X 2.5 → Axis X travels to 2.5

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Channels (CHAN) - Basic parameters -

Activation	System restart (CS)
Dependency	-
Macoda ID	7010 00010
Data type	Str8_NoUmlaut_t (isoLatin1String)
	8 byte string for MACODA compatibility

Example:

Name of channel axes

Although they are programmed under identical names, the machining axes on channels 1, 2 and 3 address different system axes. Axis "B" can be programmed on channels 2 and 3 only (see figure below).

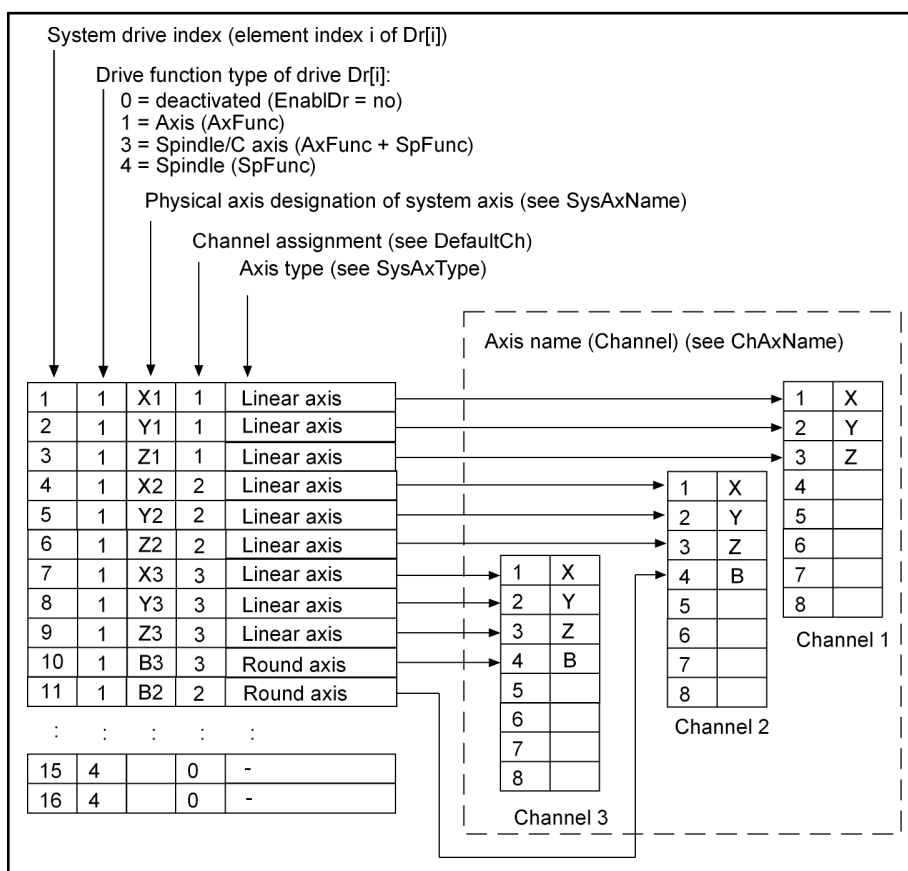


Fig. 7-4: Channel axes - system axes

Axis description

Axis meaning	ChAxClass

Defines the meaning of channel axis ChAx[1..8] with regard to machining function. The axis classification determines the axis interpolation parameter required for center point programming with circular or helical interpolation.

The axes X, Y and Z define a Cartesian axis coordinate system (=workpiece coordinate system WCS if no axis transformation is enabled) for a typical Cartesian standard machine tool. With regard to the machining function, each axis has a unique meaning. Since the MTX allows to distribute all axes in the system to different channels as desired and to "lend" these axes to other single channels under program control, this parameter defines which axes span the X, Y and z-coordinates of the axis coordinate system (= WCS if no axis

transformation is active), irrespective of the axis names configured (= coordinate meaning).

Path CHAN/Ch[k]/Coord/Acs/ChAx[1..8]/ChAxClass

Please note the following general rules:

- In every channel Ch[k], each channel axis ChAx[1..8] with the "axis name" ChAxName must have a coordinate meaning assigned to it
- In every channel Ch[k], each coordinate meaning may be assigned once only (except for "no technical relevance" and "Neutral")
- The settings of this parameter can be changed in the part program by means of G17(...), G18 or G19(...)
- G513 resets the settings of this parameter.

Value range	0	no technical relevance
	1	main axis X
	2	main axis Y
	3	main axis Z
	100	main axis A
	200	main axis B
	300	main axis C
	999	Neutral

Pre-assignment	1	main axis X .../ChAx[1]/ChAxClass
	2	main axis Y .../ChAx[2]/ChAxClass
	3	main axis Z .../ChAx[3]/ChAxClass
	0	no technical relevance .../ChAx[4..8]/ChAxClass

Unit -

Description of parameter values	0	no technical relevance The logical axis ChAx[1..8]/ChAxName is not a main axis in the axis coordinate system ACS (= WCS if no axis transformation is enabled)
	1..300	main axis X, Y, Z, A, B, C The axis named ChAxName (in folder CHAN/Ch[k]/Coord/Acs/ChAx[1..8]/) is main axis X, Y, Z, A, B or C in the axis coordinate system ACS (= WCS if no axis transformation is enabled)
	999	Neutral
		as for (0) no technical relevance

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Activation System restart (CS)

Dependency -

Macoda ID 7010 00030

Channels (CHAN) - Basic parameters -

Data type int
4 byte integer, unsigned for MACODA compatibility

Optional axis name[1..8] (OptAxName)

Optional axis name[1..8] (OptAxName)	OptChAx-Name[1..8]

Defines up to 8 optional axis names per channel Ch[k].

Based on these optional axis names, the "axis transfer" function can be used temporarily

- to transfer axes from other channels to the channel currently selected and
- to rename axes on a channel.

Path CHAN/Ch[k]/Coord/Acs/OptAxName/OptChAxName[1..8]

Value range {Text}

Pre-assignment -

Unit -

Description of parameter values**Characters**

Opt. axis designation:

Identification only with characters, e.g. X.

Letters denoting an internal NC function (e.g. D, G, H, R, S, F, A, FA) are not allowed here.

See "MTX 15VRS Programming Manual" (DOK-MTX***-NC**PRO*V15-RERS-EN-P), table of the "Default programming formats".

Characters + digits

Opt. axis designation:

Identification with character and a sequence of number, e.g. X2.

To distinguish between the axis name and the following position data, write either "=" or a "blank".

NOTICE**Danger of damage to the workpiece or the machine!**

If a short axis name is part of a longer axis name (for example: Axes "X" and "X2") and if a decimal point is programmed after that, the longer axis name is always effective:

X2.5 → Axis X2 travels to 0.5

If, on the other hand, you want axis X to travel to position 2.5, you must enter either "=" or a "blank" as delimitation:

X=2.5 or X 2.5 → Axis X travels to 2.5

Access rights

MTB:	Write
Setter:	Read
Operator:	Read

Activation System restart (CS)

Dependency -

Macoda ID 7010 00020

Data type Str8_NoUmlaut_t (isoLatin1String)

8 byte string for MACODA compatibility

Example:

Axis Name (opt.)

Channel 1 can

- temporarily transfer a total of five axes from other channels under the axis names A1, A2, U, V and W, and
- rename the names of your own axes X, Y and Z with the optional axis names A1, A2, U, V and W.

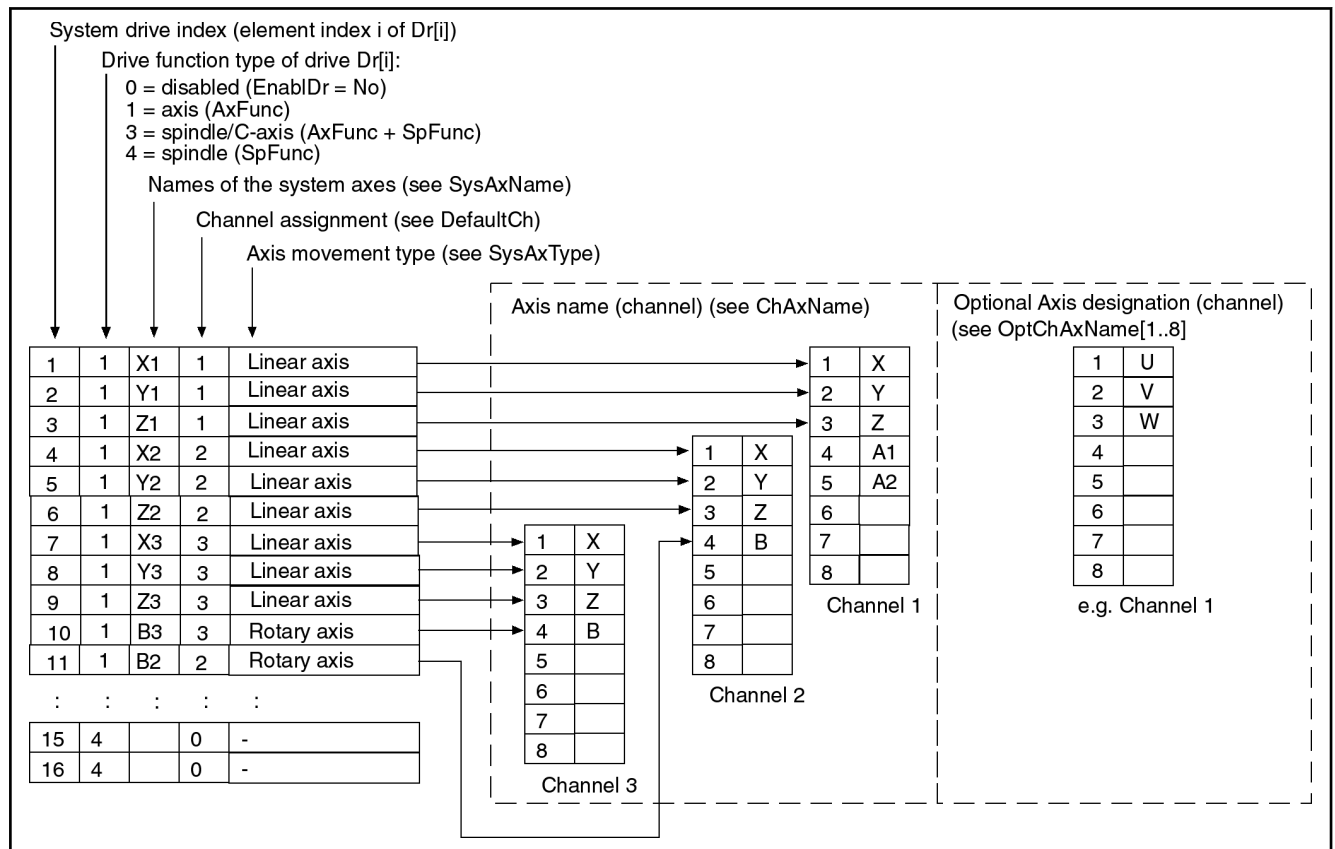


Fig. 7-5: Optional axis designation

7.4.2 Machine coordinates (MCS)

General information

Up to six coordinate identifiers are defined here, which span the machine coordinates when axis transformation 1 is enabled.

Coordinate name[1..6]

Coordinate name[1..6]	CartCoord[1..6]

Defines the names to be used to designate the six coordinates of axis transformation 1.

Path CHAN/Ch[k]/Coord/Mcs/CartCoord[1..6]

Value range {Text}

Pre-assignment X .../CartCoord[1]: 1. coordinate of the MCS
 Y .../CartCoord[2]: 2. coordinate of the MCS

Channels (CHAN) - Basic parameters -

	Z	.../CartCoord[3]: 3. coordinate of the MCS
	PHI	.../CartCoord[4]: 4. coordinate of the MCS
	THETA	.../CartCoord[5]: 5. coordinate of the MCS
	PSI	.../CartCoord[6]: 6. coordinate of the MCS
Unit	-	
Description of parameter values	For example x, y, z: 1. -3. Cartesian coordinates of axis transformation 1 in the MCS For example, PHI, THETA, PSI: 4. - 6. rotation or orientation coordinates of axis transformation 1 in the MCS	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7080 00020	
Data type	Str8_NoUmlaut_t (isoLatin1String)	
	String 8 byte for MACODA compatibility	

7.4.3 Workpiece coordinates (WCS)

General information

The coordinates in channel Ch[k] that correspond to the workpiece coordinates when axis transformation 2 is enabled are defined here.

Name of the linear coordinate[1..3]

Name of the linear coordinates[1-3]		LinCoord[1..3]
Defines the name of the linear coordinates LinCoord[1..3] of a channel Ch[k] in the WCS when axis transformation 2 is enabled. The axis transformation enabled in the NC program can specify what subset of the 6 possible coordinates LinCoord[1..3] and OriCoord[1..3] can be programmed.		
Path	CHAN/Ch[k]/Coord/Wcs/LinCoord[1..3]	
Value range	{Text}	
		The coordinates of the WCS have to be defined with lower-case letters.
Pre-assignment	x	LinCoord[1]
	y	LinCoord[2]
	z	LinCoord[3]
Unit	-	
Description of parameter values	For example, x, y, z: Name of the 1st-3rd linear coordinates in the WCS of axis transformation 2	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	

Dependency	-
Macoda ID	7080 00010
Data type	Str8_NoUmlaut_t (isoLatin1String) String 8 byte for MACODA compatibility

Name of the rotary coordinate[1..3]

		Name of the rotary coordinates[1..3]	OriCoord[k]
		Defines the names of the rotary coordinates OriCoord[1..3] of a channel Ch[k] in the WCS when axis transformation 2 is enabled.	
		The axis transformation enabled in the NC program can specify what subset of the 6 possible coordinates LinCoord[1..3] and OriCoord[1..3] can be programmed.	
Path		CHAN/Ch[k]/Coord/Wcs/OriCoord[1..3]	
Value range		{Text}	
		 The coordinates of the WCS have to be defined with lower-case letters.	
Pre-assignment	phi	OriCoord[1]	
	theta	OriCoord[2]	
	psi	OriCoord[3]	
Unit	-		
Description of parameter values	For example, phi, theta, psi: Name of the 1st-3rd rotary coordinates in the WCS of axis transformation 2		
Access rights	MTB:	Write	
	Setter:	Read	
	Operator:	Read	
Activation	System restart (CS)		
Dependency	-		
Macoda ID	7080 00010		
Data type	Str8_NoUmlaut_t (isoLatin1String)		
	String 8 byte for MACODA compatibility		

7.5 Coupling (Couple)

7.5.1 General information

This is where channel-specific settings are made in the case of coupling (master spindle/axis, G96).

7.5.2 Main spindle

Main spindle	MainSp
<i>Defines for each channel Ch[k] the</i> • Spindle whose speed determines the feed on the channel Ch[k] when G95 is enabled • Master spindle for thread cutting G33 • Master spindle for dwell time in revolutions, G4(S...)	

Channels (CHAN) - Basic parameters -

	The selection is always made in the channel spindles. The main spindle can be modified in the part program using "MAINSP()".	
Path	CHAN/Ch[k]/Couple/MainSp	
Value range	1 .. 8	
Pre-assignment	1	
Unit	-	
Description of parameter values	Index of the channel spindle ChSp[1..8], used as main spindle MainSp. The assignment system spindle to channel spindle is carried out in the parameter SpIndChSp (path: SP/SpAdmin/ChSp/ChSpGrAlloc/Ch[k]/ChSp[1..8]).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7020 00010	
Data type	ChSp1_n_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

7.5.3 Channel axis for G96

Channel axis for G96		AxG96
	Defines the channel axis ChAx[1..8], the position of which the control internally calculates the necessary revolution speed when G96 is enabled.	
Path	CHAN/Ch[k]/Couple/AxG96	
Value range	1..8	
Pre-assignment	1	
Unit	-	
Description of parameter values	Specification of the index of the channel axis ChAx[1..8] when G96 is enabled	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7010 00110	
Data type	ChAx1_n_t (unsignedByte)	
	1 byte integer, unsigned for MACODA compatibility	

7.5.4 Coordinate system for G96 reference axis

Coordinate system for G96 reference axis	CoordG96
Defines for the channel Ch[k] the reference coordinate system for G96 from which the position is taken to calculate the spindle revolution speed.	
<i>Set the reference coordinate system depending on the machine design:</i>	
- MCS if the center of the spindle is always at position 0 of the reference axis; - LCS if the spindle is positioned in relation to various stationary tools;	

		PCS if the tools are not arranged in parallel to the axis.
Path		CHAN/Ch[k]/Couple/CoordG96
Value range	1	PCS
	2	MCS(1) (LCS)
	3	MCS(0)
Pre-assignment	3	MCS(0)
Unit	-	
Description of parameter values	1	PCS Reference system: Program coordinate system (PCS)
	2	MCS(1) (LCS) Reference system: Local machine coordinate system (LCS) (machine coordinate system with zero point offsets)
	3	MCS(0) Reference system: Machine coordinate system (MCS)
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation		System reset (SR)
Dependency	-	
Macoda ID		7010 00120
Data type		Byte1_3_t (byte) 1 byte integer, unsigned for MACODA compatibility

8 Axes (AX) - Basic parameters -

8.1 General information

For every drive $Dr[i]$ defined in the MAIN data group, the following axis-specific parameters must be set:

- Limit switch monitoring, page 89
- Axis velocity data, page 92
- Override for asynchronous axes, page 94
- Axis SHAPE, page 94
- Feed forward, page 96
- Axis acceleration data, page 97
- Axis jerk data, page 99
- Jogging, page 100
- Positioning logic/modulo calculation, page 105
- Location programming, page 109
- Torque reduction, page 111
- Override evaluation for asynchronous axes, page 112

The parameter group "Override for asynchronous axes" (chapter 8.13 "Override for asynchronous axes (AsynchrAxOvrldf)" on page 112) only concerns drives that have been defined as "asynchronous axes" (see chapter 6 NC Basic Configuration (MAIN) - Configuration Data Setup, page 47, parameter SupprCmdValOut in the folder "MAIN/Dr[i]/").

8.2 Limit switch monitoring (Pos)

8.2.1 General information

There are two software limit switches, each of which sets a positive or negative traversing range limit within the physically possible traversing range of an axis. Between these two positive and negative traversing range limits is the "permissible traversing range" (see the figure below).

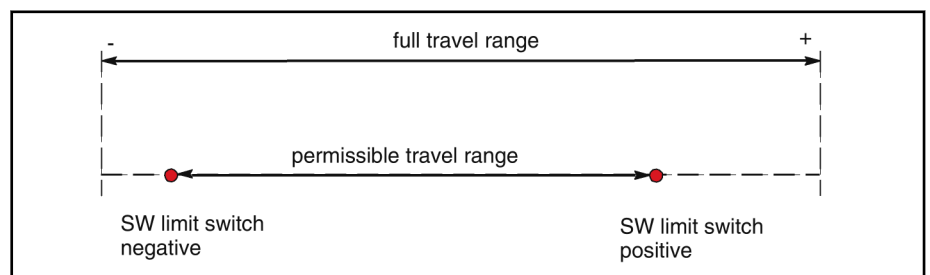


Fig. 8-1: The traversing range of a software limit switch within the full traversing range

The included traversing ranges of both software limit switch may overlap, swap over or exist simultaneously (see the figure below).

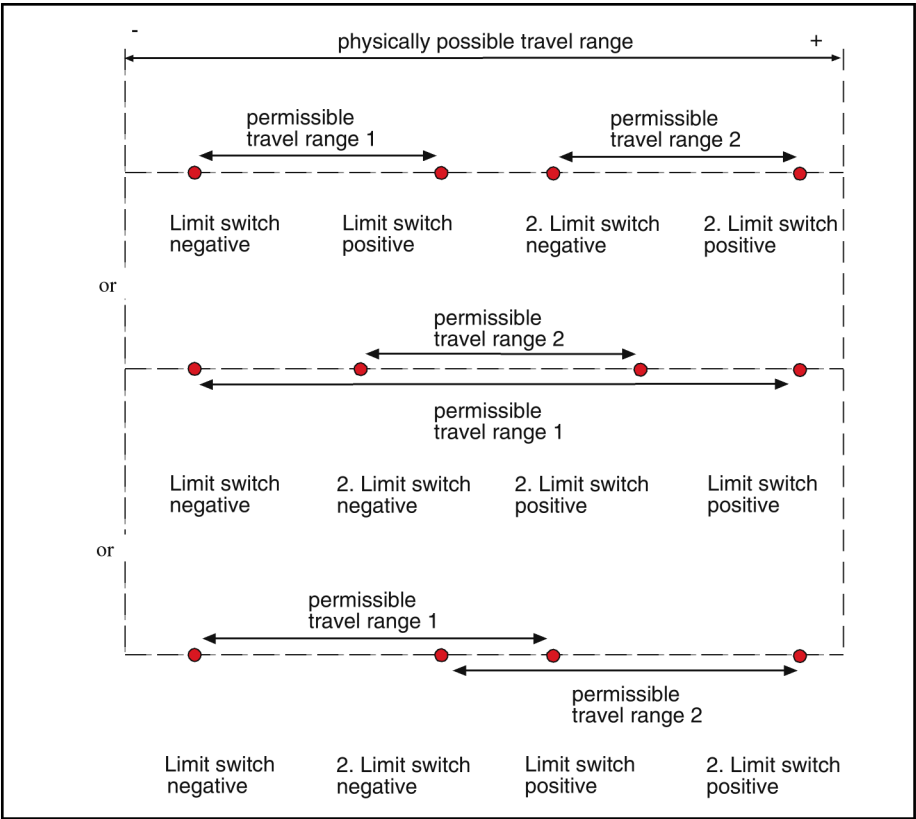


Fig. 8-2: Overlapping, swapped and independent traversing ranges of two software limit switch



The software limit switches TrvLim[1..2] of axes Dr[i] can be hidden statically by EnablTrvLim or dynamically, i.e. temporarily, using the corresponding axis interface. This makes it possible to swap from one permitted traversing range to another.

8.2.2 Software limit switch (TrvLim[1..2])

Enabling the software limit switch

Enable software limit switch	EnablTrvLim

Enables/disables software limit switch 1 or 2 (TrvLim[1..2]).

NOTICE

Mechanical collision possible because software travel limit 2 disabled!

If software travel limit 1 has been disabled, software travel limit 2 will be automatically "disabled", even though it was not explicitly disabled. Both limit switches are always enabled/disabled simultaneously in the NC Configurator.

Path	AX/Dr[i]/Pos/TrvLim[1..2]/EnablTrvLim	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No

		Software limit switch (TrvLim[1..2]) disabled
	1	Yes
		Software limit switch (TrvLim[1..2]) enabled
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1020 00001	
	Macoda values -99999.998 .. 99999.998 indicate "Both software limit switches are enabled".	
	Macoda value 99999.999 indicates "Both software limit switches are disabled".	
Data type	Byte0_1_t (byte)	
	8 byte float, 3 decimal places for MACODA compatibility	

Positive travel limit

Positive travel range limit		PoTrvLim
Limits the maximum permissible position input in the positive travel direction. If a position beyond this maximum allowable position is entered, a runtime error will occur unless the limit switches are disabled. If the "positive travel limit" is less than the "negative travel limit", this leads to a runtime error as soon as a position is entered.		
Path	AX/Dr[i]/Pos/TrvLim[1..2]/PoTrvLim	
Value range	-1000000.000 .. 1000000.000 -99999.998 .. 99999.998 for MACODA compatibility 99999.999 for MACODA compatibility (this value only applies to Macoda ID 1020 00001 and indicates "Software limit switch 1 and 2 (TrvLim[1..2]) disabled".	
Pre-assignment	1000.000	
Unit	mm or ° (degrees)	
Description of parameter values	Maximum permissible position input in positive travel direction.	
 The values set here must not exceed the limits stored under Sercos S-0-0049 (position limit, positive) in the respective drive!		
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1020 00001, 1020 00003	

Axes (AX) - Basic parameters -



Contrary to the displays in the NC Configurator, different Macoda ID assignments apply to both software limit switches (positive).

Software limit switch 1: AX/Dr[i]/Pos/TrvLim[1]/PoTrvLim corresponds to Macoda ID 1020 00001.

Software limit switch 2: AX/Dr[i]/Pos/TrvLim[2]/PoTrvLim corresponds to Macoda-ID 1020 00003.

Data type Pos_t (double, 3 decimal places)
8 byte float, 3 decimal places for MACODA compatibility

Negative travel limit

Negative travel range limit	NeTrvLim

Limits the maximum permissible position input in the negative travel direction.

If a position below this minimum allowable position is entered, a runtime error will occur unless the limit switches are disabled.

If the "positive travel limit" is **smaller** than the "negative travel limit", it causes a runtime error as soon as a position is entered.

Path AX/Dr[i]/Pos/TrvLim[1..2]/NeTrvLim

Value range -1000000.000 .. 1000000.000
-99999.998 .. 99999.998 for MACODA compatibility

Pre-assignment -1000.000

Unit mm or ° (degrees)

Description of parameter values Minimum permissible position input in negative travel direction.



The values set here must not fall below the limits stored under Sercos S-0-0050 (position limit, negative) in the respective drive!

Access rights **MTB:** Write
Setter: Read
Operator: Read

Activation System reset (SR)

Dependency -

Macoda ID 1020 00002, 1020 00004



Contrary to the displays in the NC Configurator, different Macoda ID assignments apply to both software limit switches (negative).

Software limit switch 1: AX/Dr[i]/Pos/TrvLim[1]/NeTrvLim corresponds to Macoda ID 1020 00002.

Software limit switch 2: AX/Dr[i]/Pos/TrvLim[2]/NeTrvLim corresponds to Macoda-ID 1020 00004.

Data type Pos_t (double, 3 decimal places)
8 byte float, 3 decimal places for MACODA compatibility

8.3 Axis velocity data (Vel)

8.3.1 Maximum axis velocity

Maximum axis velocity	MaxVel

	<i>Specifies for each axis Dr[i]</i>	
	- its maximum permissible axis velocity, and - its rapid traverse velocity.	
Path	AX/Dr[i]/Vel/MaxVel	
Value range	0.000 .. 1000000000.000	
	0.000 .. 36000000.000 for MACODA compatibility	
Pre-assignment	10000.000	
Unit	mm/min or °/min	
Description of parameter values	Maximum permissible rapid traverse or axis velocity	
	 The drive monitors the axis velocity by means of the Sercos parameter S-0-0091 (drive limit). "MaxVel" has to be less than this drive limit.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1005 00002	
Data type	Vel_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

8.3.2 Feed calculation

Feed computation		FeedCalc
	Specifies for each machining axis whether it shall take feed calculation into account in the channel with FeedAd (FAD) or be moved along with the channel axes contributing to the feed.	
Path	AX/Dr[i]/Vel/FeedCalc	
Value range	0	Take axes into account with "FeedAd" (FAD)
	1	Move axes along simultaneously (interpolation)
Pre-assignment	0	Take axes into account with "FeedAd" (FAD)
Unit	-	
Description of parameter values	0	The machining axis is taken into account for feed calculation while FAD is enabled.
	1	The machining axis is moved along while FAD is enabled. The dynamic limit values of these axes are taken into account and the interpolative relationship of these axes is retained (axes running synchronously).
	 This parameter has no relevance to auxiliary axes. If no axes have been programmed in an NC block to contribute to the feedrate calculation, there is the option to program the feedrate of the other axes via the 2nd feedrate value (via address "Omega").	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Axes (AX) - Basic parameters -

Activation	System reset (SR)
Dependency	-
Macoda ID	1003 00020
Data type	Byte0_1_t (byte)
	1 byte integer, unsigned for MACODA compatibility

8.4 Override for asynchronous axes (Ovrd)

8.4.1 Activating axis-specific override

Enable axis-specific override		EnablAxAsynchrOvrd
Specifies for each axis Dr[i] defined as an auxiliary axis (in "DefaultCh" in folder MAIN/Dr[i]/AxFunc/) whether the override function for auxiliary axes is to be enabled.		
 This parameter has no relevance to machining axes.		
Path	AX/Dr[i]/Ovrd/EnablAxAsynchrOvrd	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No
	Override function for auxiliary axes is not enabled (= override is fixed to 100%).	
	1	Yes
	Override function for auxiliary axes is enabled. The current override value is entered directly over the axis interface of the axis concerned (qAx_Override_00 to qAx_Override_15).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00100	
	(MACODA compatibility: Machine parameter 1003 00100 is less than EnablAxAsynchrOvrd by 1, i.e. the value "-1" corresponds in this case to the setting "0" (No).)	
Data type	Byte0_1_t (byte)	
	1 byte integer for MACODA compatibility	

8.5 Axis SHAPE (SHAPE)

8.5.1 SHAPE order for axis

SHAPE order for axis	AxShape
<i>Specifies for each axis Dr[i]:</i>	
the default setting for G9(ASHAPE) (= axis-wise programmed SHAPE),	

		- the default setting for unprogrammed axis SHAPE orders with G9(<Axis><SHAPE order>,...) - the default setting for asynchronous axis movements and - the SHAPE order for axis jogging
Path	AX/Dr[i]/Shape/AxShape	
Value range	1 .. 99	
Pre-assignment	1	
Unit	-	
Description of parameter values	0, 1	SHAPE order 0 or 1 defines that no limitation is in effect.
	>1	SHAPE order for axis
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1003 00008	
Data type	Byte1_99_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	
	<i>Example:</i>	

Setting Ashape parameter

Xs=30 Ys=20 Zs=10 Vs=40	;Enabled axis SHAPE orders:
G9(ASHAPE)	;Xs=30,Ys=20,Zs=10
G9(X50) X0 Y0 Z0	;Xs=50,Ys=20,Zs=10
V312 FA1000	;Vs=40
V312 FA1000 ASO5	;Vs=5
or	
Jog X+	;Xs=30

8.5.2 Minimum SHAPE order for axis

Minimum SHAPE order for axis	MinAxShape

Specifies for each axis Dr[i] the minimum SHAPE order (jerk limit).

The minimum axis-related order of SHAPE is effective with:

- G9(ASHAPE)
- G9(<Coord><shape>)
- Asynchronous axis movements (e.g. U100 FA10000)
- Axis jogging

Neither the "SHAPE order for axis" AxShape (in folder AX/Dr[i]/Shape/), nor an axis-specific SHAPE order programmed with G9 or ASO may fall below the minimum axis-related SHAPE order.

The minimum SHAPE order is calculated from the allowed axis jerk with the following formula:

Axes (AX) - Basic parameters -

$$\text{SHAPE order} = \frac{\text{Axis acceleration}[\text{m/s}^2]}{\text{Jerk}[\text{m/s}^3] * \text{Interpolation cycle time}[\mu\text{s}] * 10^{-6}}$$

Fig. 8-3: Calculation of SHAPE order

Path	AX/Dr[i]/Shape/MinAxShape	
Value range	1 .. 99	
Pre-assignment	1	
Unit	-	
Description of parameter values	0, 1	SHAPE order 0 or 1 defines that no limitation is in effect.
	>1	Minimum axis-related SHAPE order
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1003 00007	
Data type	Byte1_99_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Example:

Minimum axis-related SHAPE order

With an interpolation cycle time of 4000 ms and an axis acceleration of 10,000 mm/smm/s², a SHAPE order of 25 limits the axis jerk to 100 m/s³.

8.6 Feed forward (FeedFwd)

8.6.1 Feed forward can be activated (via FFW/FFWA)

Feed forward can be activated (via FFW/FFWA)	AxFeedFwd

Specifies for each axis whether to switch to secondary mode -1 of the drive in the case of "FeedForward" (FFW/FFWA) programming.

Under secondary mode -1 (S-0-0033), a parameter must be entered for an operation mode which provides the function "Zero following error movement".

The "Zero following error movement" functionality is configured in Sercos parameters which differ depending on the drive series:

- *IndraDrive:*
 - P-0-0040 (pilot control calculation for pilot velocity control)
 - S-0-0348 (proportional amplification pilot acceleration control)
- *ServoDyn D:*
 - P-0-0500 (calculation pilot velocity control)
 - P-0-0501 (calculation pilot acceleration control)
- *DiAx, EcoDrive:* P-0-005

Path	AX/Dr[i]/FeedFwd/AxFeedFwd	
Value range	0	No
	1	Yes

Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No
		No switching of drive to secondary mode -1 after programming of FFW/FFWA
	1	Yes
		Switching of drive to secondary mode -1 after programming of FFW/FFWA
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1003 00009	
Data type	Byte0_1_t (byte)	
	4 byte integer, unsigned for MACODA compatibility	

8.7 Axis acceleration data (Acc)

8.7.1 Maximum axis acceleration in feed mode

Max. acceleration in feed mode		FeedAxAcc
Specifies for each axis Dr[i] the maximum acceleration in feed mode (G1, G2, ...).		
 The values set here must be smaller than the limits stored under S-0-0138 for the respective drive!		
At block transitions within an interpolation cycle, the parameter "Axis acceleration jump" AxAccReserve (in folder AX/Dr[i]/Acc/) applies.		
Path	AX/Dr[i]/Acc/FeedAxAcc	
Value range	0.000 .. 1000.000	
	0.001 .. 1000.000 for MACODA compatibility	
Pre-assignment	1.000	
Unit	m/s ² or 1000°/s ²	
Description of parameter values	Maximum axis acceleration in Feed mode	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	S-0-0138	
	AxAccReserve (axis acceleration reserve)	
Macoda ID	1010 00001	
Data type	Acc_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

8.7.2 Maximum axis acceleration in rapid traverse

	Max. acceleration in rapid traverse	RapidAxAcc
	Specifies for each axis Dr[i] the maximum acceleration in rapid traverse (G0, ...).	
	 The values set here must be smaller than the limits stored under S-0-0138 for the respective drive!	
	At block transitions within an interpolation cycle, the parameter "Axis acceleration jump" AxAccReserve (in folder AX/Dr[i]/Acc/) applies.	
Path	AX/Dr[i]/Acc/RapidAxAcc	
Value range	0.000 .. 1000.000	
Pre-assignment	0.000	
Unit	m/s ² or 1000°/s ²	
Description of parameter values	0.000 For rapid traverse, the following value set under FeedAxAcc becomes effective >0.000 Maximum axis acceleration in rapid traverse	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	S-0-0138 AxAccReserve (axis acceleration reserve) FeedAxAcc (max. acceleration in feed mode)	
Macoda ID	1010 00003	
Data type	Acc_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

8.7.3 Axis acceleration reserve

	Axis acceleration reserve	AxAccReserve
	Defines for every machining axis Dr[i] the maximum acceleration that can be generated (velocity jump) at the block transitions.	
	The axis jump at the knee in the contour is defined by the axis acceleration reserve AxAccReserve and additionally by the axis jerk reserve AxJerkReserve (in folder AX/Dr[i]/Jerk) if the jerk limitation is enabled.	
Path	AX/Dr[i]/Acc/AxAccReserve	
Value range	0.000 .. 1000.000	
Pre-assignment	8.000	
Unit	m/s ² or 1000 °/s ²	
Description of parameter values	Permissible axis acceleration jump at block transition	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	

Macoda ID 1010 00012
 Data type JumpAcc_t (double, 3 decimal places)
 8 byte float, 3 decimal places for MACODA compatibility

8.8 Axis jerk data (Jerk)

8.8.1 Maximum axis jerk in feed mode

	Max. jerk in feed mode	FeedAxJerk
	Specifies for each axis Dr[i] the maximum jerk in feed mode (G1, G2, ...).	
Path	AX/Dr[i]/Jerk/FeedAxJerk	
Value range	0.000 .. 1000000.000	
Pre-assignment	0.000	
Unit	m/s ³ or 1000°/s ³	
Description of parameter values	Synchronous axes	
	0.000	If the jerk limitation of the channel was enabled using the NC function "JerkControl(1)", this value causes a runtime warning.
	>0.000	If the jerk limitation of the channel was enabled using the NC function "JerkControl(1)", the value specified is used for the jerk limitation.
	Asynchronous axes	
	0.000	The jerk limitation of the asynchronous axes is disabled.
	>0.000	The jerk limitation limits the jerk of the asynchronous axis to the given value.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1010 00020	
Data type	Jerk_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

8.8.2 Maximum axis jerk in rapid traverse

	Max. jerk in rapid traverse	RapidAxJerk
	Specifies for each axis Dr[i] the maximum jerk in rapid traverse (G0,...).	
Path	AX/Dr[i]/Jerk/RapidAxJerk	
Value range	0.000 .. 1000000.000	
Pre-assignment	0.000	
Unit	m/s ³ or 1000°/s ³	
Description of parameter values	Synchronous axes	
	0.000	If the NC function "JerkControl(1)" is enabled, the following value for rapid traverse movements set under FeedAxJerk becomes effective.

Axes (AX) - Basic parameters -

	>0.000	If the NC function "JerkControl(1)" is enabled, the following value for rapid traverse movements set here becomes effective.
	Asynchronous axes	
	0.000	The value set under FeedAxJerk is used.
	>0.000	The jerk limitation limits the jerk of the asynchronous axis to the given value.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	FeedAxJerk (max. axis jerk in feed mode)	
Macoda ID	1010 00022	
Data type	Jerk_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

8.8.3 Axis jerk reserve

Axis jerk reserve		AxJerkReserve
Defines for every machining axis Dr[i] the maximum jerk that can be generated (acceleration jump) at the block transitions.		
This parameter is only applicable when the jerk limitation was enabled in the respective channel by the NC function "JerkControl(1)".		
The axis jump at the knee in the contour is defined by the axis acceleration reserve AxAccReserve (in folder AX/Dr[i]/Acc) and additionally by the axis jerk reserve AxJerkReserve if the jerk limitation is enabled.		
Path	AX/Dr[i]/Jerk/AxJerkReserve	
Value range	0.000 .. 1000000.000	
Pre-assignment	5.000	
Unit	m/s ³ or 1000°/s ³	
Description of parameter values	Jerk reserve allowed at the block transitions	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1010 00013	
Data type	Jerk_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

8.9 Jog mode (Jog)

8.9.1 General information

The parameters of the jog mode include distances (scaling of an increment), acceleration, jerk and jog velocity.

8.9.2 Jog mode (increments and acceleration)

Scaling of a jog increment

Scaling of a jog increment		IncrAxJog
	Specifies for each axis how an increment will be scaled in jog or handwheel mode.	
	If the input signal qAx_JogInch (inch increments) is set on the axis interface of a linear axis, the control multiplies the value set here by the factor "2.54". thus, inch increments can be traversed in jog mode.	
Path	AX/Dr[i]/Jog/IncrAxJog	
Value range	0.000 .. 100.000 0.001 .. 100.000 for MACODA compatibility	
Pre-assignment	1.000	
Unit	µm or 0.001°	
Description of parameter values	Increment scaling in jog or handwheel mode	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1015 00001	
Data type	Float0_100_t (float, 3 decimal places) 4 byte float, 3 decimal places for MACODA compatibility	

Variable jog distance

Variable jog distance		VarDistAxJog
	Specifies for each axis Dr[i] how many increments shall be travelled in the travel step "X increments" after pressing the jog button (see also Jog velocity variable steps, page 104 VarDistAxVel in folder AX/Dr[i]/Jog/Vel/).	
Path	AX/Dr[i]/Jog/VarDistAxJog	
Value range	1.000 .. 1000000.000 1 .. 1000000 for MACODA compatibility	
Pre-assignment	100000.000	
Unit	Incr	
Description of parameter values	Number of increments in the travel step "X increments"	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1015 00002	
Data type	JogDist_t (double, 3 decimal places) 4 byte integer, unsigned for MACODA compatibility	

Jog acceleration

Jog acceleration		AccAxJog
	Specifies for each axis Dr[i] the acceleration and delay in setup mode when jogging.	
Path	AX/Dr[i]/Jog/AccAxJog	
Value range	0.000 .. 1000.000	
Pre-assignment	0.000	
Unit	m/s ² or 1000 °/s ²	
Description of parameter values	Axis acceleration or delay (in setup mode when jogging)	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1010 00002	
Data type	Acc_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

Jog jerk

Jog jerk		JerkAxJog
	Specifies for each axis Dr[i] the jerk value in setup mode when jogging.	
Path	AX/Dr[i]/Jog/JerkAxJog	
Value range	0.000 .. 1000000.000	
Pre-assignment	0.000	
Unit	m/s ³ or 1000°/s ³	
Description of parameter values	0.000	The value set under AX/Dr[i]/Jerk/FeedAxJerk is used. If FeedAxJerk contains the value 0.000 as well, the jerk limitation is switched off.
	>0.000	The jerk is limited to the value set.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	AX/Dr[i]/Jerk/FeedAxJerk (max. axis jerk in feed mode)	
Macoda ID	1010 00021	
Data type	Jerk_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

8.9.3 Jog mode (jog velocity)

Jog velocity slow

Jog velocity slow	SlowAxVel
Specifies for each axis Dr[i] the velocity when jogging at velocity level "manual feed slow".	

Path	AX/Dr[i]/Jog/Vel/SlowAxVel
Value range	0.000 .. 1000000000.000 0.000 .. 36000000.000 for MACODA compatibility
Pre-assignment	250.000
Unit	mm/min or °/min
Description of parameter values	Jog velocity "manual feed slow"
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1005 00003
Data type	Vel_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

Jog velocity medium

Jog velocity medium		MediumAxVel
Specifies for each axis Dr[i] the velocity when jogging at velocity level "manual feed medium".		
Path	AX/Dr[i]/Jog/Vel/MediumAxVel	
Value range	0.000 .. 1000000000.000 0.000 .. 36000000.000 for MACODA compatibility	
Pre-assignment	2500.000	
Unit	mm/min or °/min	
Description of parameter values	Jog velocity "manual feed medium"	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1005 00004	
Data type	Vel_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

Jog velocity fast

Jog velocity fast		FastAxVel
Specifies for each axis Dr[i] the velocity when jogging at velocity level "manual feed fast".		
Path	AX/Dr[i]/Jog/Vel/FastAxVel	
Value range	0.000 .. 1000000000.000 0.000 .. 36000000.000 for MACODA compatibility	
Pre-assignment	5000.000	
Unit	mm/min or °/min	
Description of parameter values	Jog velocity "manual feed fast"	

Axes (AX) - Basic parameters -

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1005 00005	
Data type	Vel_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

Jog velocity rapid traverse

Jog velocity rapid traverse		RapidAxVel
	Specifies for each axis Dr[i] the velocity when jogging at velocity level "manual feed rapid traverse".	
Path	AX/Dr[i]/Jog/Vel/RapidAxVel	
Value range	0.000 .. 1000000000.000 0.000 .. 36000000.000 for MACODA compatibility	
Pre-assignment	10000.000	
Unit	mm/min or °/min	
Description of parameter values	Jog velocity "manual feed rapid traverse"	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1005 00006	
Data type	Vel_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

Jog velocity defined steps

Jog velocity defined steps		DefStepAxVel
Specifies for each axis Dr[i] the velocity when jogging at velocity level "manual feed for defined steps".		
Path	AX/Dr[i]/Jog/Vel/DefStepAxVel	
Value range	0.000 .. 1000000000.000 0.000 .. 36000000.000 for MACODA compatibility	
Pre-assignment	250.000	
Unit	mm/min or °/min	
Description of parameter values	Jog velocity "manual feed for defined steps"	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1005 00007	

Data type Vel_t (double, 3 decimal places)
8 byte float, 3 decimal places for MACODA compatibility

Jog velocity variable steps

Jog velocity variable steps		VarDistAxVel
	Specifies for each axis Dr[i] the velocity when jogging at velocity level "manual feed for variable steps".	
Path	AX/Dr[i]/Jog/Vel/VarDistAxVel	
Value range	0.000 .. 1000000000.000 0.000 .. 36000000.000 for MACODA compatibility	
Pre-assignment	250.000	
Unit	mm/min or °/min	
Description of parameter values	Jog velocity "manual feed for variable steps"	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1005 00008	
Data type	Vel_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

8.10 Positioning logic/Enable modulo calculation (PosLog)

8.10.1 General information

If an axis Dr[i] is operated with modulo calculation, it is a special type of linear or rotary axis.

In this case, the control automatically reads a "modulo value" (sometimes also called a modulo factor) from the drive (parameter S-0-0103) during Sercos startup.

With the "modulo value", the current position of the axis after reaching the target position is transformed back into a suitable position range between "0" and the "modulo value" (**endless axis**). If the absolute position value programmed is greater than the modulo value, the control responds with a runtime error.

The programming zero point can be defined using the "reset modulo axis" function LinModZp (LMZ).

A suitable positioning logic is selected for the axis (see "Positioning logic" parameter ModePosLog in folder AX/Dr[i]/PosLog/).

8.10.2 Enabling the modulo calculation

Enable the modulo calculation		EnablModCalc
Enables or disables modulo calculation for the axis Dr[i].		
Path	AX/Dr[i]/PosLog/EnablModCalc	
Value range	0	No
	1	Yes

Axes (AX) - Basic parameters -

	(1 .. 4)	For MACODA compatibility (values only apply for the internal representation of the MACODA parameter 1003 00004 "Axis motion type", which, compared to "EnablModCalc", contains additional information on whether it is a linear or rotary axis)
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Modulo calculation disabled
	1	Yes
		Modulo calculation enabled
	(1 .. 4)	for MACODA compatibility: Axis motion type (refer to the notes under "Value range")
		1 = linear axis without modulo calculation,
		2 = rotary axis with modulo calculation,
		3 = rotary axis without modulo calculation,
		4 = linear axis with modulo calculation,
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	The following dependency exists to the parameter SysAxType (path: MAIN/Dr[i]/AxFunc):	
	When enabling the modulo calculation (EnablModCalc = 1) an automatic switch to rotary axis takes place if neither linear nor rotary axis is defined as axis type.	
Macoda ID	1003 00004	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

8.10.3 Positioning logic

	Positioning logic	ModePosLog
	Specifies, for each axis with enabled modulo calculation (see parameter "Enable modulo calculation", Element EnablModCalc in the same folder chapter 8.10.2 "Enabling the modulo calculation" on page 105), in which direction the axis has to rotate to move to a programmed target position.	
Path	AX/Dr[i]/PosLog/ModePosLog	
Value range	0	No positioning logic
	1	shortest path (DC)
	2	ACP/ACN sign logic
	4	limited path
Pre-assignment	1	shortest path (DC)
Unit	-	
Description of parameter values	0	position specification

The direction of rotation when moving into a target position is determined exclusively by the balance of the new and the previous position:

- positive result: clockwise rotation,
- negative result: counterclockwise rotation

1 shortest path (DC)

The endless axis always moves on the shortest possible path to its physical target position. Programmed signs, if any, are ignored. Therefore, the maximum movement is 180 degrees or half of the current modulo value.

2 ACP/ACN sign logic

The direction of rotation for moving into the target position is determined exclusively by the signs programmed:

- + equals clockwise rotation
"+" does not have to be programmed)
- - equals counterclockwise rotation

The maximum movement per block is always less than 360 degrees or is less than the current modulo value. If G91 is active (relative dimension programming), it remains effective.

4 limited path

The enabled software travel limits (TrvLim[1] and TrvLim[2]) determine the permissible traversing range and the possible direction. The permissible traversing range extends from the negative to the positive limit switch. The limit switch ranges turn active once the axis has been referenced. Coordinate specifications are valid both in the negative and the positive modulo ranges (e.g. 20° corresponds to -340°). The coordinate signs and the local coordinate attribute "DC" are ignored. The attributes "ACP", "ACN" and "IC", on the other hand, are observed. In case of violation of the working range, they cause an error message.

Access rights **MTB:** Write
 Setter: Read
 Operator: Read

Activation System reset (SR)

Dependency -

Macoda ID 1003 00005

Data type Byte0_4_t (byte)

1 byte integer, unsigned for MACODA compatibility

Example:

Positioning logic with options 1 (shortest path (DC)) and 2 (sign logic (ACP / ACN))

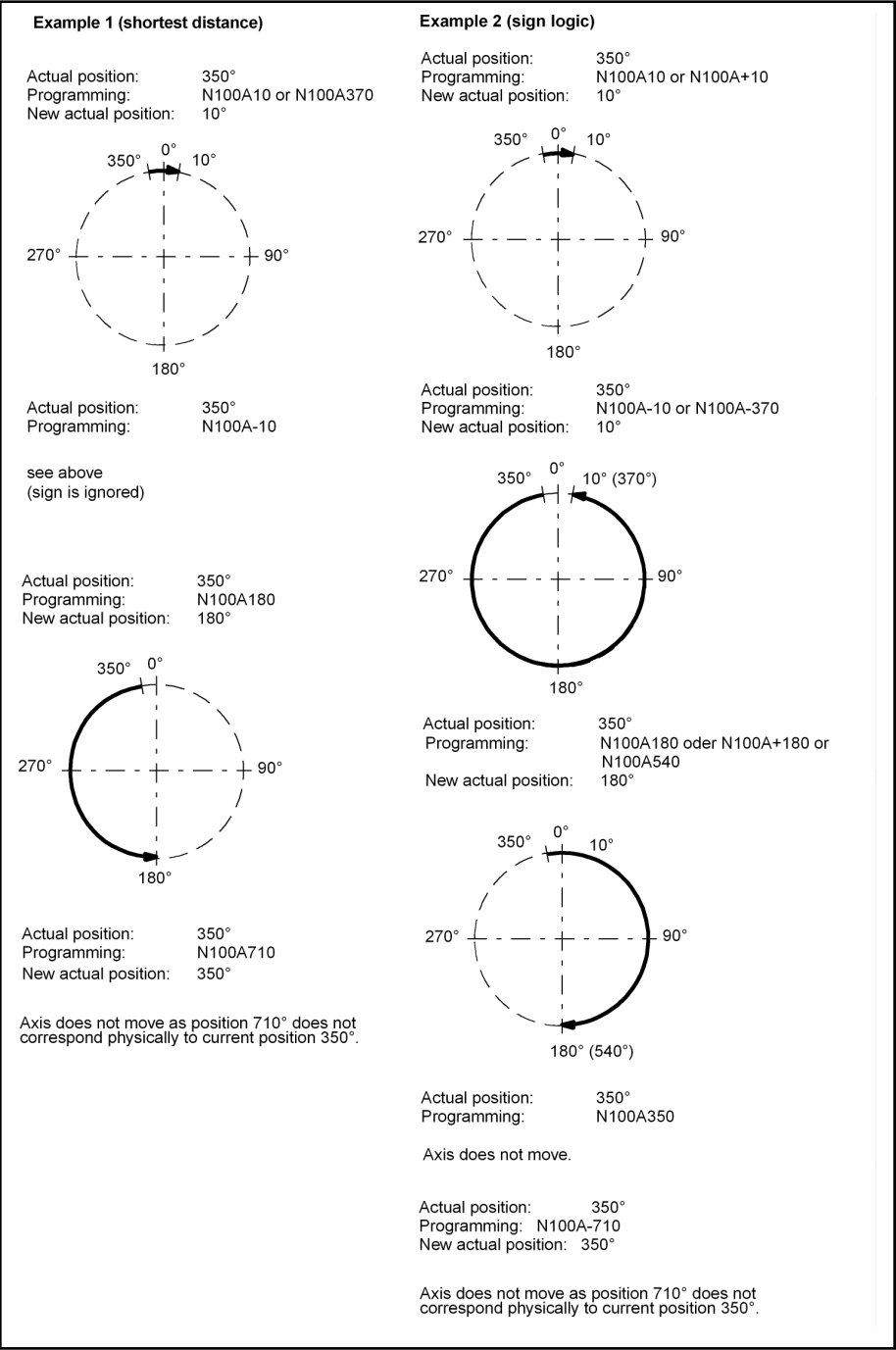


Fig. 8-4: Positioning logic

8.10.4 Positioning logic can be switched via NC program

Positioning logic can be switched per NC program	EnablChgPosLog

Specifies, for each axis with enabled modulo calculation (see parameter "Enable modulo calculation", Element EnablModCalc in the same folder [chapter 8.10.2 "Enabling the modulo calculation" on page 105](#)), whether its

	positioning logic can be switched with the NC programming command "Pos-Mode (PMD)".	
Path	AX/Dr[i]/PosLog/EnablChgPosLog	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Positioning Logic Cannot Be Switched per NC Program.
	1	Yes
		Positioning logic can be switched per NC program.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1003 00050	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

8.11 Location programming (LocProg)

8.11.1 General information

Location programming concerns axes that shall be employed as "Hirth axes". Upon selecting the type of location programming in the parameter ModeLocProg, an axis Dr[i] automatically becomes a Hirth axis.

8.11.2 Type of location programming

Type of location programming		ModeLocProg
	Determines for Hirth axes whether their position values shall be programmed and displayed by "position input" or by "location programming".	
Path	AX/Dr[i]/LocProg/ModeLocProg	
Value range	0	position specification
	1	Location programming
Pre-assignment	1	Location programming
Unit	-	
Description of parameter values	0	position specification
		Programming position values of a Hirth axis in ° (degrees)
	1	Location programming
		Programming the position of a Hirth axis by location number
		Irrespective of the above, all shift values (e.g. resulting from a zero shift) are interpreted in ° (degrees).
Access rights	MTB:	Write
	Setter:	Read

	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00056	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

8.11.3 Splitting

Number of locations

Number of positions	NofLoc

Specifies for rotary or endless axes the number of corresponding grid positions.

 This parameter has no relevance to axes that are neither rotary nor endless axes.

Hirth axes can only travel to positions located on a fixed grid. All grid positions are equally distanced.

Path	AX/Dr[i]/LocProg/Part/NofLoc	
Value range	0 .. 1000000	
	0 .. 360000 for MACODA compatibility	
Pre-assignment	0	
Unit	-	
Description of parameter values	for rotary axes: grid positions per revolution for endless axes: grid positions per modulo range.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00055	
Data type	Int0_1M_t (int)	
	4 byte integer, unsigned for MACODA compatibility	

8.11.4 Automatic location correction

Enabling automatic location correction

Enable automatic location correction	EnablAutoCorr

Specifies whether or not the control shall automatically correct programmed position values not coinciding with the Hirth grid.

Path	AX/Dr[i]/LocProg/AutoCorr/EnablAutoCorr	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	

Description of parameter values	0	No
		No automatic correction of programmed positions. Positions not coinciding with the Hirth grid will trigger a run-time error.
	1	Yes
		Automatic correction of programmed positions. Every position programmed will be corrected with the effect that the Hirth axis moves to the nearest grid position.
	 Corrections and zero point shifts must end on the Hirth grid (see also "number of locations" NofLoc parameter in folder AX/Dr[i]/LocProg/Part/).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00057	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

8.12 Torque reduction (TorqRed)

8.12.1 Reduced maximum torque when torque reduction enabled

Reduced torque at active torque reduction		RedMaxTorq
Specifies for each axis Dr[i] what value the Sercos parameter S-0-0092 temporarily overwrites in the drive as soon as the axis interface signal "torque reduction" (qAx_TrqLim) becomes logically TRUE.		
	 The drive has to be parameterized in a way that the value in S-0-0092 is interpreted as % of the nominal torque!	
Path	AX/Dr[i]/TorqRed/RedMaxTorq	
Value range	0.000 .. 1000.000	
	0.00 .. 500.00 for MACODA compatibility	
Pre-assignment	1.000	
Unit	%	
Description of parameter values	Reduced maximum torque (default)	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00010	
Data type	Torq_t (float, 3 decimal places)	

4-byte float, 2 decimal places for MACODA compatibility

8.12.2 Maximum standstill torque

Maximum standstill torque		MaxStandstTorq
Defines for each axis Dr[i] its standstill torque limit .		
Path	AX/Dr[i]/TorqRed/MaxStandstTorq	
Value range	0.00 0.01 .. 1000.00	
Pre-assignment	0.00	
Unit	%	
Description of parameter values	0.00	No standstill torque monitoring enabled
	>0.00	Value of maximum torque that may be produced when axis is at a standstill.
 The corresponding drive has to be parameterized in a way that the value in S-0-0092 is interpreted as % of the nominal torque!		
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00070	
Data type	Torq_t (float, 3 decimal places)	
	8 byte float, 2 decimal places for MACODA compatibility	

8.13 Override for asynchronous axes (AsynchrAxOvrdf)

8.13.1 General information

Settings for the "asynchronous axes" (auxiliary axes) are made here (see also parameter "Channel assignment" DefaultCh in folder MAIN/Dr[i]/AxFunc/DefaultCh).

8.13.2 Max. override

Maximum override		MaxAxOvrdf
Specifies the maximum permissible feed override (as a %) for asynchronous axes.		
Path	AX/AsynchrAxOvrdf/MaxAxOvrdf	
Value range	0.00 .. 999.00	
Pre-assignment	120.00	
Unit	%	
Description of parameter values	Maximum permissible feed override	
Access rights	MTB: Write Setter: Read Operator: Read	

Activation	System restart (CS)
Dependency	-
Macoda ID	7030 00012
Data type	Ovr_t (float, 2 decimal places) 4-byte float, 2 decimal places for MACODA compatibility

8.13.3 Override value for step [1..32]

Override value for step[1..32]	AxOvrStep[1-32]

Specifies the values of the individual steps of the feed override for an asynchronous axis.

In order for the override logic for auxiliary axes (from AX/Dr[i]/Ovr/EnablAxAsynchrOvr) to become active, it must be enabled for the respective auxiliary axis in the parameter Override asynchronous axes (EnablAxAsynchrOvr in the folder AX/Dr[i]/Ovr/) (see [chapter 8.4.1 "Activating axis-specific override" on page 94](#)).

Unless the override function is enabled, the effective feedrate override for the auxiliary axis concerned is always 100%.

Irrespective of the currently selected step, the maximum permissible override value will always be adhered to within the NC according to AX/AsynchrAxOvrDlf/MaxAxOvr.

Path	AX/AsynchrAxOvrDlf/AxOvrStep[1..32]
Value range	0.00 .. 150.00

Pre-assignment	0.00	.../AxOvrStep[1]
	1.00	.../AxOvrStep[2]
	2.00	.../AxOvrStep[3]
	4.00	.../AxOvrStep[4]
	6.00	.../AxOvrStep[5]
	8.00	.../AxOvrStep[6]
	10.00	.../AxOvrStep[7]
	20.00	.../AxOvrStep[8]
	30.00	.../AxOvrStep[9]
	40.00	.../AxOvrStep[10]
	50.00	.../AxOvrStep[11]
	60.00	.../AxOvrStep[12]
	70.00	.../AxOvrStep[13]
	80.00	.../AxOvrStep[14]
	100.00	.../AxOvrStep[15]
	120.00	.../AxOvrStep[16]
	0.00	.../AxOvrStep[17-32]

MACODA compatibility:

120.00 7030 00010 [1] ("Channel 0"), no relevance

AxOvrStep[n] 7030 00010 [n + 1] ("Channel 0")

Unit %

Description of parameter values Percentage values for a maximum of 32 override steps. Values can be freely selected and programmed up to 150%. Since the individual values do not have to be in ascending order, a Gray code-based override is possible as well.

 The override step to be active is selected via the axis-related interface signals "qAx_Override_00"-"qAx_Override_04", unless "0.00" is entered for all individual parameters [1] to [32]. In such a case, the control interprets the interface signals

 "qAx_Override_00"..."qAx_Override_15" directly as an override value scaled 0.01 %. This allows to set the override within the range of 0% to 655.35% in steps of 0.01%.

Access rights **MTB:** Write

Setter: Read

Operator: Read

Activation System restart (CS)

Dependency -

Macoda ID 7030 00010 ("Channel 0")

 (MACODA compatibility: In machine parameter 7030 00010, the indices are shifted by 1, i.e. AxOvrStep[1] has index 2.)

Data type Ovr_t (float, 2 decimal places)

 4-byte float, 2 decimal places for MACODA compatibility

Example:

Enabling the individual steps via axis-related interface (IF) signals:

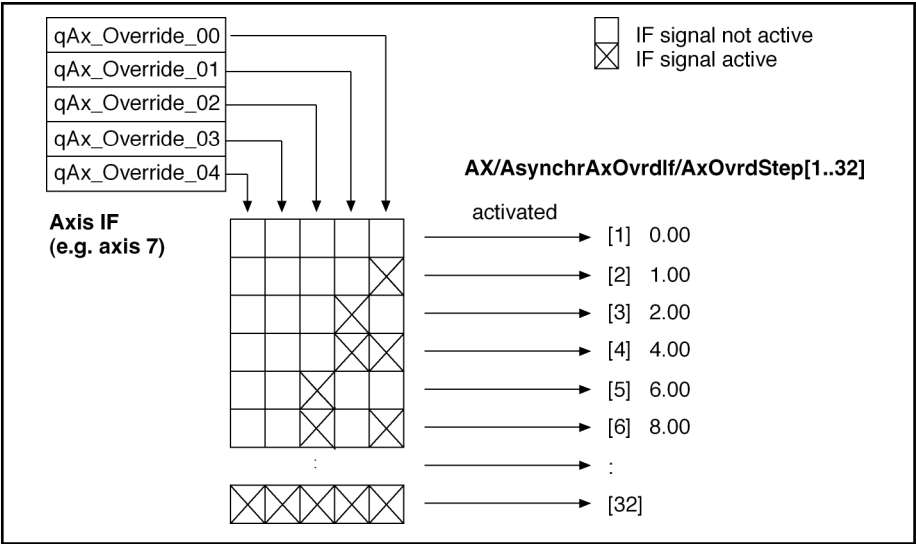


Fig. 8-5: Feed override (auxiliary axes) and interface signals

9 Spindles (SP) - Basic parameters

9.1 General information

The "spindle functions" of the system spindles Sp[1..60] (spindles or spindles with C-axis operation) are parameterized in the Spindles (SP) data group of the basic parameters.



Additional C-axis operation of a system spindle is configured in the data groups "basic NC configuration (MAIN)" and "axes (AX)" under the paths MAIN/Dr[i]/AxFunc and AX/Dr[i]/... of the respective spindle drive Dr[i].

Select the following settings for maximum 60 spindles:

- [Spindle functions of system spindles, page 115](#)
- [Spindle group functions, page 144](#)
- [Spindle management, page 152](#)

9.2 Spindle functions (SpFunc)

9.2.1 General information

Spindles are defined in 2 steps:

1. by assigning the system spindle index to the system-wide drive index Dr[i]
2. by assigning the system spindles Sp[1..60] to channel spindles

9.2.2 Definition of system spindle

MTX manages up to 60 system spindles.

The type and number of system spindles is first defined in the data group "MAIN" by the selection of the drive function type ([Chapter 6 NC basic configuration \(MAIN\) setup of the configuration data, page 47](#)) (applies to spindles with or without C-axis operation). In doing so, the system spindle index will be **automatically** assigned to the system-wide drive index Dr[i].

"System spindle 1 (Sp[1])" (system spindle index = 1) is assigned to the drive Dr[i] that has been configured as first spindle or as spindle/C-axis when selecting the drive function type ([Chapter 6 NC Basic Configuration \(MAIN\) Setup of the Configuration Data, page 47](#)); "system spindle 2 (Sp[2])" (system spindle index = 2) is assigned to the next correspondingly assigned drive [i + x], and so on.

Additionally, in the context menu of the spindle folder "SP/SpFunc" of the "SP" data group, the number of spindles (not including spindles with C-axis operation!) can be incremented by one spindle Sp[n+1] or an existing spindle deleted (spindle with highest system spindle index (Sp[max]) first!).



For each spindle added, an additional parameter group "Drive" Dr[i]/AxFunc is automatically created in the "MAIN" and "AX" data groups (only for a spindle with C-axis operation). The parameters contained therein and all associated settings must also be accounted for when making further parameter settings.

The order of assignment is independent of the "spindle type" configured under SpType in the folder SP/SpFunc/Sp[1..60]/Base/. The spindle type can be

a Sercos spindle, auxiliary spindle and analog spindle. The total number of spindles, of whatever spindle type, is limited to 60.

9.2.3 Assignment of system spindle to channel spindle/spindle group

A system spindle can be assigned to a "channel spindle" or a "spindle group" (=compatibility mode).

- **Channel spindles:**

Up to eight system spindles can be assigned to one channel at a time. These spindles are then named "channel spindles".

- **Spindle groups:**

The first eight system spindles can be distributed over as many as four spindle groups, where all system spindles are also assigned to every defined channel Ch[k] (see figure "Spindle groups (compatibility mode)").

Example: System spindle 1-8 exists in channel 1-3.

All remaining spindles or spindles with C-axis operation defined under MAIN/Dr[i] are ignored in compatibility mode.

The system spindle to be selected as channel spindle/spindle group is located in the Spindle management folder (SP/SpAdmin); see Spindle Management (see [Chapter 9.4 Spindle Management \(SpAdmin\)](#), page 152)

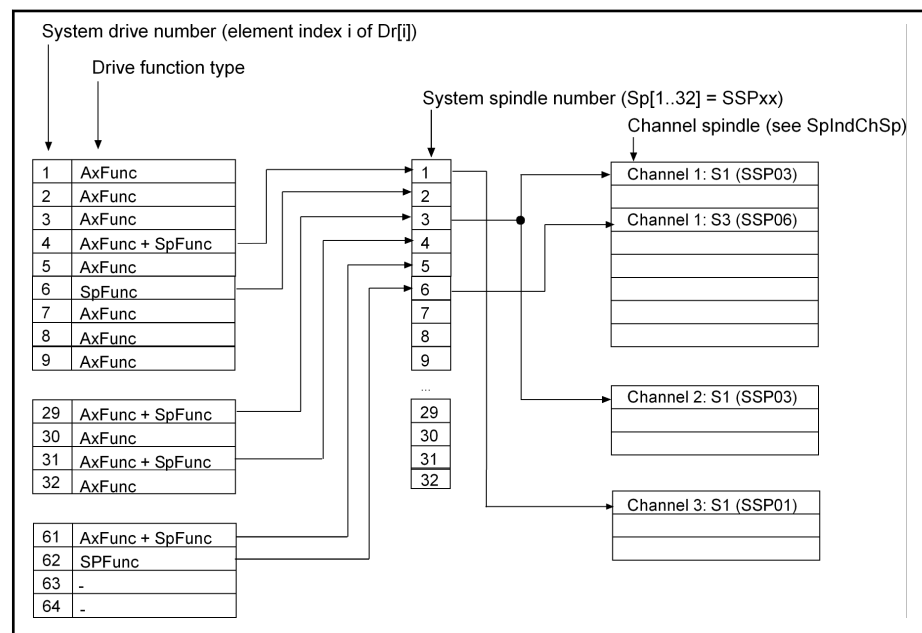


Fig. 9-1: System spindle assignment of channel spindles

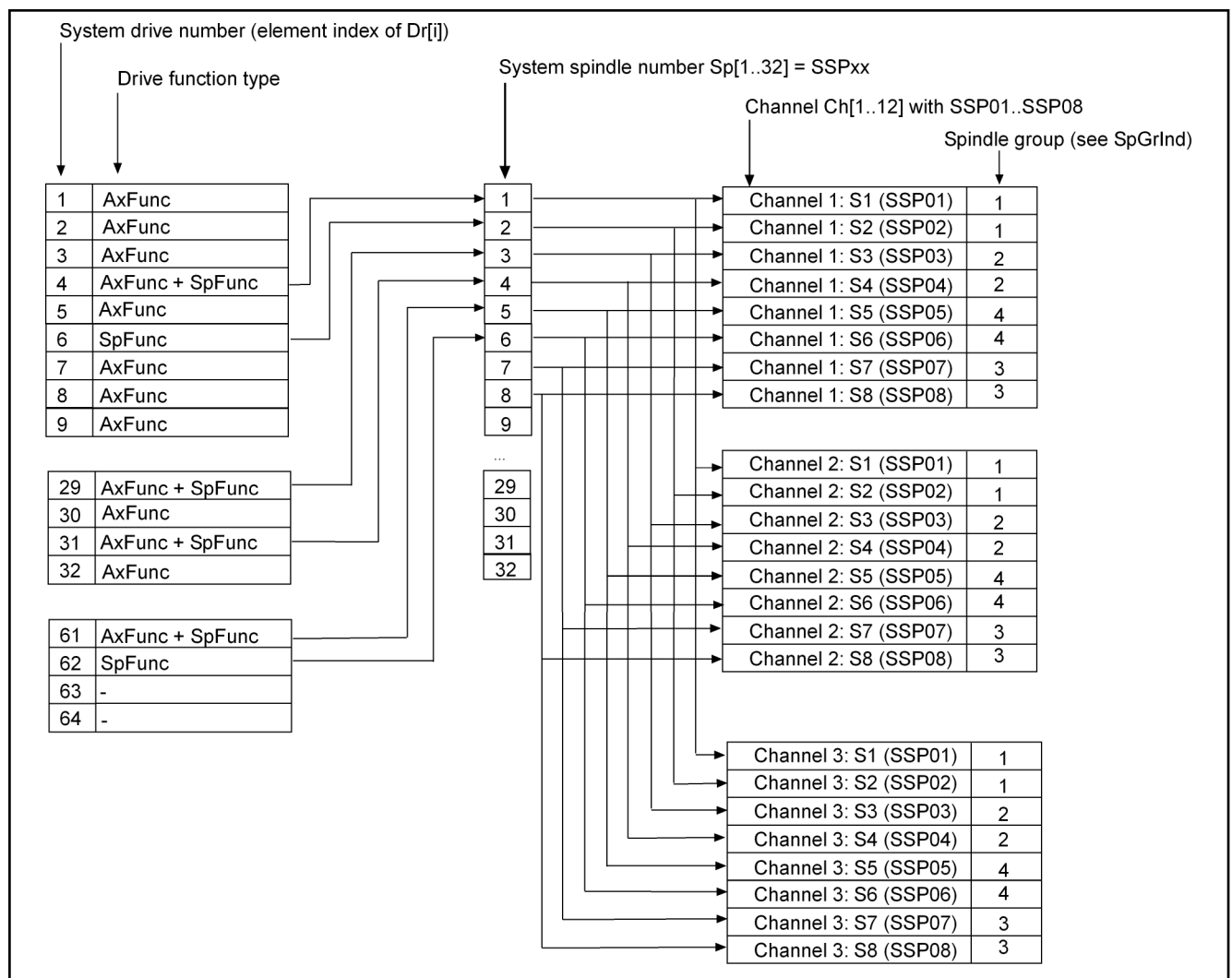


Fig. 9-2: System spindle assignment of spindle groups (compatibility mode)

9.2.4 Functional overview

Every system spindle Sp[1..60] is parameterized with the following data:

- Basic data, page 117
- Gears, page 120
- Spindle override, page 128
- Speed window, page 130
- Jog spindle, page 133
- Spindle turret axis, page 134
- Spindle triggering, page 135
- Spindle syntax for system spindles / channel spindles, page 135
- Spindle syntax[1..8], page 136

9.2.5 Basic data

Physical spindle designation

Physical spindle designation	SysSpName

Permits the definition of a system-comprehensive, unique spindle name for each system spindle Sp[1..60].



Depending on whether a name is specified in SysSpName or not, there are different impacts.

Case 1: A name is specified in SysSpName:

- For diagnostic messages, the SysSpName content is used.
- For the system spindle programming (SSPM(), SSPO(), ...), the content of SysSpName as well as the identifier SSP01 .. SSP60 and SSp01 .. SSp60 can be used.

Case 2: No name is specified in SysSpName:

- For diagnostic messages, the internally generated identifiers SSP01 .. SSP60 used.
- For the system spindle programming (SSPM(), SSPO(), ...), only the internally generated identifiers SSP01 .. SSP60 and SSp01 .. SSp60 can be used.

Path	SP/SpFunc/Sp[1..60]/Base/SysSpName	
Value range	{Text}	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Characters	e.g. "SP"
		The identification "S" is invalid!
	Characters + digits	e.g. "SP5"
		The identifications "S0" to " S9", "S00" to "S99", "SSP00" to "SSP99" and "SSp00" to "SSp99"!
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00401	
Data type	Str8_NoUmlaut_t (isoLatin1String)	
	8-byte string, no umlauts for MACODA compatibility	

Spindle type

Spindle type	SpType

Defines for each system spindle Sp[1..60]:

- the spindle type and
- whether it shall be hidden

The spindle types differ as follows:

	Sercos spindle	External spindle	Analog spindle	External spindle with encoder	Analog spindle with encoder
The spindle functionality is implemented in the NC	Yes	No	Yes	No	Yes
Syntax elements of the various spindle functions are defined via machine parameter	Yes	Yes	Yes	Yes	Yes
There is a spindle-specific I/O interface available for each spindle	Yes	Yes, without function	Yes	Yes	Yes
Spindles can be assigned to spindle groups (parallel spindles) irrespective of the spindle type	Yes	Yes	Yes	Yes	Yes
The spindle is actuated by:	NC	PLC	NC	PLC	NC
Max. number of spindles which can be configured per type	60	60	4*)	60	4*)

*) The total of the number of "analog spindles" and "analog spindles with encoder" may not exceed 4.

Tab. 9-1: Spindle types



Specify the assignment between analog spindles and the desired analog outputs with "assignments of analog outputs [1..4]" AnalOAllocOut [1..4] (in the folder PLC/AnaIO/FuncAllocOut/).

Path	SP/SpFunc/Sp[1..60]/Base/SpType	
Value range	0	spindle is hidden
	1	Sercos spindle
	2	External spindle (auxiliary function spindle)
	3	Analog spindle
	4	External spindle with encoder
	5	Analog spindle with encoder
Pre-assignment	1	Sercos spindle
Unit	-	
Description of parameter values	0	spindle is hidden You may use this function if, e.g., in the case of a whole series of machines, the number of spindles used can be configured. First, you should configure all spindles in view of the final, maximum expansion options and then you can simply disable those spindles that are not required on a machine.
	1	Sercos spindle System spindle Sp[1..60] is a Sercos spindle.
	2	External spindle (auxiliary function spindle) System spindle Sp[1..60] is an externally realized spindle.

Spindles (SP) - Basic parameters

	3	Analog spindle System spindle Sp[1..60] is an analog spindle (max. 4).
	4	External spindle with encoder System spindle Sp[1..60] is an externally spindle with encoder.
	5	Analog spindle with encoder System spindle Sp[1..60] is an analog spindle with encoder (max. 4).
	 The total number of all spindles defined with SpType may not exceed 60. The total of the number of "analog spindles" and "analog spindles with encoder" may not exceed 4.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00001	
Data type	Byte0_5_t (byte) 1 byte integer, unsigned for MACODA compatibility	

9.2.6 Gears

Number of gear steps

Number of gear ranges		NofGears
	Specifies for every spindle Sp[1..60] in the system the number of gear.	
Path	SP/SpFunc/Sp[1..60]/Gear/NofGears	
Value range	1..4	
Pre-assignment	1	
Unit	-	
Description of parameter values	Maximum number of gears	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00010	
Data type	Byte0_4_t (byte) 1 byte integer, unsigned for MACODA compatibility	

Gear change method

Gear change method	GearChgMethod
Defines for every spindle Sp[1..60] in the system the type of gear range changing.	

Normally, a mechanical gear is changed when programming a function to select gear ranges (e.g. M42). The NC can trigger a winding change by switching this parameter.

Path	SP/SpFunc/Sp[1..60]/Gear/GearChgMethod	
Value range	0	Gears
	1	Winding
Pre-assignment	0	Gears
Unit	-	
Description of parameter values	0	Gears
		Mechanical switching of a gear with "spindle spinning".
	1	Winding
		Switching contactor to operate a motor with different winding or different winding wiring (star / delta connection).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00008	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Gear step[1..4]

The following data can be defined for up to 4 gear ranges GearStep[1..4]:

- Maximum acceleration
- Minimum speed
- Maximum acceleration
- Acceleration in position control mode[1]
- Acceleration in position control mode[2]
- Switchover speed to 2nd acceleration
- Max. spindle jerk in position control operation

Maximum speed of GearStep[1..4]

Maximum Speed of GearStep[1..4]	MaxSpSpeed

Defines spindle-specifically Sp[1..60] the following for each gear GearStep[1..4] (number parameterized in NofGears in the folder SP/SpFunc/Sp[1..60]/Gear/):

- the maximum spindle speed and
- the maximum speed for tapping without compensating chuck (G63)

If a command value higher than the maximum spindle speed specified in this parameter is entered for the gear GearStep[1..4] currently set, either the speed will be limited or the next higher gear GearStep[(1..3)+1] will be selected automatically.

Path	SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]/MaxSpSpeed
Value range	0.000 .. 200000.000
Pre-assignment	6000.000

Spindles (SP) - Basic parameters

Unit	1/min
Description of parameter values	Maximum spindle speed in one gear range GearStep[1..4]
Access rights	MTB: Write
	Setter: Read
	Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00012
Data type	SpSpeed_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

Minimum speed of GearStep[1..4]

Minimum Speed of GearSet[1..4]	MinSpSpeed

Defines the minimum spindle speed Sp[1..60] for each gear GearStep[1..4] (number in NofGears in the folder SP/SpFunc/Sp[1..60]/Gear/ parameterized) spindle-specifically.

If a lower command value than the min. spindle speed specified in this parameter is entered for the gear range currently set, either a manual change of gear will be suggested or the next lower gear will be **automatically** selected.

**Automatic gear select:**

Syntax for

- "individual spindles"
(see SpGearAutoSel in the folder SP/SpFunc/SpSyn/Syn[1..8]/),
- "Spindle groups"
(see SpGrGearAutoSel in the folder SP/SpGrFunc/SpGrSyn[1..4]/).

Path	SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]/MinSpSpeed
Value range	0.000 .. 200000.000
Pre-assignment	0.000
Unit	1/min
Description of parameter values	Minimum spindle revolution speed in one gear range GearStep[1..4]
Access rights	MTB: Write
	Setter: Read
	Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00011
Data type	SpSpeed_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

Maximum acceleration of GearStep[1..4]

Maximum Acceleration of GearStep[1..4]	MaxSpAcc

Defines the spindle acceleration on "clockwise, counterclockwise" and "spindle stop" specifically for each spindle Sp[1..60] for each gear range GearStep[1..4] (number in NofGears in the folder SP/SpFunc/Sp[1..60]/Gear/ parameterized).



Acceleration values of drive controlled functions, e.g. "spindle positioning" or "spindle oscillation" are stored in the drive and therefore not affected by this parameter.

Path	SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]/MaxSpAcc
Value range	0.000 .. 100000.000
Pre-assignment	100.000
Unit	rad/s ²
Description of parameter values	Maximum acceleration in a gear step GearStep[1..4]
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00031
Data type	SpAcc_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

Acceleration in position controller range[1] in GearStep[1..4]

Acceleration in Position Controller Range[1] in GearStep[1..4]	MaxSpAccPosCtrl[1]

Defines for each spindle Sp[1..60] in gear step GearStep[1..4]:

- when acceleration when positioning interface is enabled (e.g. position-synchronous mode) **below** the "switching speed" defined in SwiSpeed (in the folder SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]/)
- the acceleration for tapping without compensating chuck (G63) across the entire speed range

Path	SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]/MaxSpAccPosCtrl[1]
Value range	0.000 .. 100000.000
Pre-assignment	0.000
Unit	rad/s ²
Description of parameter values	0.000 At this value, • The maximum spindle acceleration within the active gear step (MaxSpAcc) in the folder SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]/ • The parameters MaxSpAccPosCtrl[2] and SwiSpeed in the folder SP/SpFunc/Sp[1..60]/Gear/GearStep[g]/ are not active • Tapping without compensating chuck (G63) uses the max. axis acceleration FeedAxAcc (in the folder AX/Dr[j]/Acc/) >0.000 1. acceleration for active position interface
Access rights	MTB: Write

Spindles (SP) - Basic parameters

	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00051	
Data type	SpAcc_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

Acceleration in position controller range[2] in GearStep[1..4]

Acceleration in position controller range[2] in GearStep[1..4]		MaxSpAccPosCtrl[2]
Defines for every spindle Sp[1..60] in gear step GearStep[1..4] the acceleration when positioning interface is enabled (e.g. position-synchronous mode) above the "switch speed" defined in SwiSpeed (in the folder SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]).		
Path	SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]/MaxSpAccPosCtrl[2]	
Value range	0.000 .. 100000.000 0.001 .. 100000.000 for MACODA compatibility	
Pre-assignment	0.010	
Unit	rad/s ²	
Description of parameter values	Position interface: 2. Acceleration	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00052	
Data type	SpAcc_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

Switch speed to 2nd acceleration of GearStep[1..4]

Switch speed to 2nd acceleration of GearStep[1..4]		SwiSpeed
<i>Defines for each spindle (in folder SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]), when it is switched between the</i> "1. Acceleration" MaxSpAccPosCtrl[1] "2. acceleration" MaxSpAccPosCtrl[2] in case of an active position interface Above the given revolution speed, MaxSpAccPosCtrl[2] takes effect; below, MaxSpAccPosCtrl[1].		
Path	SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]/SwiSpeed	
Value range	0.000 .. 200000.000	
Pre-assignment	0.000	
Unit	1/min	
Description of parameter values	0.000	Via the full speed range, only the 1st acceleration (MaxSpAccPosCtrl[1]) takes effect.

	>0.000	Switch speed between 1st and 2nd acceleration
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00053	
Data type	SpSpeed_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

Maximum spindle jerk in position control operation of GearStep[1..4]

Maximum spindle jerk in position control operation of Gear-Step[1..4]		MaxSpJerkPosCtrl
Defines the maximum jerk in the position control operation (tapping without compensating chuck, G63) for each spindle Sp[1..60] for each gear step GearStep[1..4] (number in NofGears in the folder SP/SpFunc/Sp[1..60]/Gear/parameterized).		
Path	SP/SpFunc/Sp[1..60]/Gear/GearStep[1..4]/MaxSpJerkPosCtrl	
Value range	0.000 .. 100000.000	
Pre-assignment	0.000	
Unit	rad/s ³	
Description of parameter values	0.000	The jerk limitation of the spindle is disabled.
	>0.000	The jerk limitation limits the spindle jerk to the given value.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00055	
Data type	Double0_100T_t (double, 3 decimal places)	

Drive-side parameter sets (ScsParSet)

The optimum controller parameter settings of the spindle drive for speed mode and position mode may (but do not have to) differ. For optimum adjustment, up to three Sercos parameter sets can be assigned to each gear step. The first NC parameter set is always assigned to the spindle mode with speed interface.

Example:

Drive Commissioning with IndraDrive

The following results for 2 NC parameter sets per gear step:

Number of gear ranges	Drive parameter set (S-x-) for	
	NC parameter block 1	NC parameter block 2
1	0	1
2	0, 1	2, 3

Spindles (SP) - Basic parameters

	Drive parameter set (S-x-) for	
Number of gear ranges	NC parameter block 1	NC parameter block 2
3	0, 1, 2	3, 4, 5
4	0, 1, 2, 3	4, 5, 6, 7

Tab. 9-2: Drive parameter sets (example 1)

For 3 parameter sets per gear step, you get:

	Drive parameter set (S-x-) for		
Number of gear ranges	NC parameter block 1	NC parameter block 2	NC parameter block 3
1	0	1	2
2	0, 1	2, 3	4, 5

Tab. 9-3: Drive parameter sets (example 2)

The spindle of a milling machine possesses 2 gears.

The drive is to be customized optimally for tapping without compensating chuck (G63) and normal spindle mode.

The following settings exist (see below for parameters):

NofScsParSets = 2 → 2 drive parameter block per gear step

PosCtrlSpSet = 2 → 2nd parameter set is used for G63

PosCtrlCAxSet = 1 → no relevance, since no C-axis operation used.

In the parameter "Number of gears" NofGears (in the folder SP/SpFunc/Sp[1..60]/Gear/), there is the setting:

NofGears = 2 → 2 gear steps

For the drive, the following results:

- Gear stage 1 in spindle mode uses Sercos parameters S-0-xxxx and P-0-xxxx.
- Gear 2 in spindle mode uses Sercos parameters S-1-xxxx and P-1-xxxx.
- Gear 1 in G63 mode uses Sercos parameters S-2-xxxx and P-2-xxxx.
- Gear 2 in G63 mode uses Sercos parameters S-3-xxxx und P-3-xxxx.

Drive parameter sets per gear step (IndraDrive only)

Drive parameter sets per gear step (IndraDrive only)	NofScsParSets

Specification of multiple parameter sets per gear step to optimize the controller parameters for operation using speed interface or positioning interface.

The first NC parameter set is **always** assigned to spindle mode with speed interface. NC parameter sets 1, 2 or 3 can be assigned to position-controlled spindle mode (tapping/positioned-controlled spindle and C-axis), depending on the number configured.

Path	SP/SpFunc/Sp[1..60]/Gear/ScsParSet/NofScsParSets
Value range	1 .. 3
Pre-assignment	1
Unit	-
Description of parameter values	Number of NC parameter sets used per gear step
Access rights	MTB: Write

	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00009 [1]	
Data type	Byte0_3_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Parameter set for G63 and position-controlled spindle mode (IndraDrive only)

Parameter set for G63 and position-controlled Spindle mode (only IndraDrive)		PosCtrlSpSet
Assignment of the controller parameter set for G63 and position-controlled spindle mode (SPM, SPCC) when multiple parameter sets exist for each gear step (NofScsParSets > 1).		
Path	SP/SpFunc/Sp[1..60]/Gear/ScsParSet/PosCtrlSpSet	
Value range	1 .. 3	
Pre-assignment	1	
Unit	-	
Description of parameter values	NC parameter set No. for G63 and position-controlled spindle mode (SPM, SPCC)	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00009 [2]	
Data type	Byte0_3_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Parameter set for C-axis operation (IndraDrive only)

Parameter set for C-axis operation (IndraDrive only)		PosCtrlCAxSet
Assignment of the controller parameter set for C-axis operation (STA) when multiple parameter sets exist for each gear step (NofScsParSets >1).		
Path	SP/SpFunc/Sp[1..60]/Gear/ScsParSet/PosCtrlCAxSet	
Value range	1 .. 3	
Pre-assignment	1	
Unit	-	
Description of parameter values	NC parameter set no. for C-axis operation	
 This parameter is not effective for a combined spindle-turret axis in turret mode! In turret mode, the CST axis with IndraDrive automatically uses the first drive parameter set following the parameter sets assigned to the spindle.		
Access rights	MTB:	Write
	Setter:	Read

Spindles (SP) - Basic parameters

	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00009 [3]	
Data type	Byte0_3_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

9.2.7 Spindle override

Maximum spindle override

Maximum spindle override		MaxSpOvrd
Specifies the maximum permissible spindle override.		
The spindle override internally effective in the NC is always limited to this parameter value.		
Path	SP/SpFunc/Sp[1..60]/Ovrd/MaxSpOvrd	
Value range	0.00 .. 150.00	
Pre-assignment	120.00	
Unit	%	
Description of parameter values	Maximum spindle override	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00042	
Data type	Ovr_t (float, 2 decimal places)	
	4-byte float, 2 decimal places for MACODA compatibility	

Override value for step[1..32]

Override value for step[1..32]		SpOvrdStep[1..32]
Specifies the values of the individual steps of the spindle override for the spindle Sp[1..60].		
Irrespective of the currently selected step, the maximum permissible override value will always be complied with in the NC according to SP/SpFunc/Sp[1..60]/Ovrd/MaxSpOvrd.		
Path	SP/SpFunc/Sp[1..60]/Ovrd/SpOvrdStep[1..32]	
Value range	0.00 .. 150.00	
Pre-assignment	45.00 .../SpOvrdStep[1] 50.00 .../SpOvrdStep[2] 55.00 .../SpOvrdStep[3] 60.00 .../SpOvrdStep[4] 65.00 .../SpOvrdStep[5] 70.00 .../SpOvrdStep[6] 75.00 .../SpOvrdStep[7] 80.00 .../SpOvrdStep[8]	

	85.00	.../SpOvrStep[9]
	90.00	.../SpOvrStep[10]
	95.00	.../SpOvrStep[11]
	100.00	.../SpOvrStep[12]
	105.00	.../SpOvrStep[13]
	110.00	.../SpOvrStep[14]
	115.00	.../SpOvrStep[15]
	120.00	.../SpOvrStep[16]
	0.00	.../SpOvrStep[17..32]
	MACODA compatibility:	
	120.00	1040 00041 [1], no relevance
	SpOvrStep[n]	1040 00041 [n + 1]
Unit	%	
Description of parameter values	Percentage values for a maximum of 32 override steps. Values can be freely selected and programmed up to 150%. Since the individual values do not have to be in ascending order, a Gray code-based override is possible as well. The override step to be active is selected via the channel-related interface signals "qSp_Override_00"... "qSp_Override_04" unless not all individual parameters SpOvrStep [1] up to [32] are assigned with a value of "0.00". In such a case, the control interprets the interface signals "qSp_Override_00"... "qSp_Override_15" directly as override value with the scaling 0.01 %. This allows to set the override within the range of 0% to 655.35% in steps of 0.01 %.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00041	
	(MACODA compatibility: in machine parameter 1040 00041, the indices are shifted by 1, i.e. SpOvrStep[1] has index 2.)	
Data type	Ovr_t (float, 2 decimal places)	
	4-byte float, 2 decimal places for MACODA compatibility	

Example:

Enabling the various steps via spindle-related interface (IF) signals:

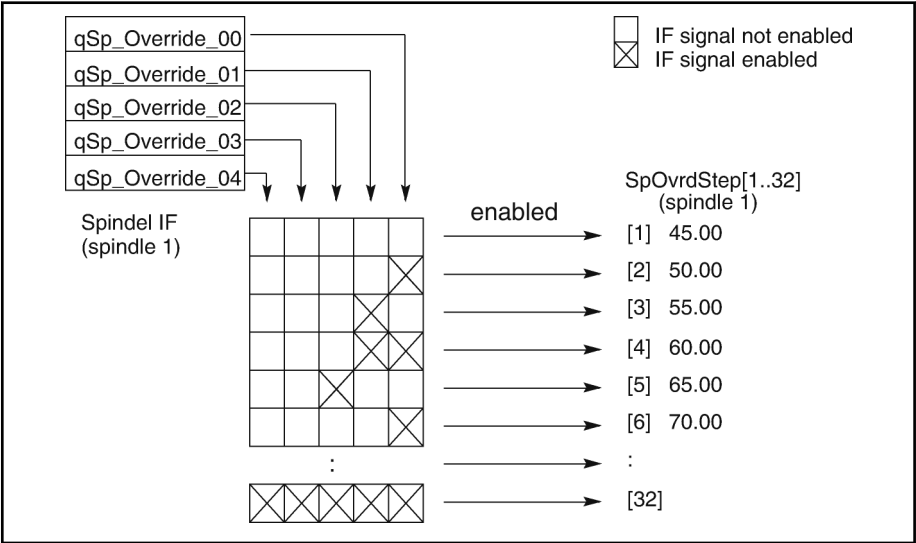


Fig. 9-3: Spindle override values and interface signals

9.2.8 Speed window

Speed window in RPM

Speed window absolute		AbsWin
Defines for each spindle Sp[1..60] the speed window for output of the spindle interface signal "Speed reached" (iSp_ProgSpReach).		
If the difference between the programmed speed (accounted for by the spindle override) and the actual speed is within the window parameterized, the control activates the spindle interface signal "Spindle speed reached".		
The values of the parameters AbsWin and PercWin are simultaneously effective; i.e. if either one of the two conditions is met, the signal "Spindle speed reached" will be set.		
Path	SP/SpFunc/Sp[1..60]/SpeedWin/AbsWin	
Value range	0.000 .. 1000.000	
Pre-assignment	2.000	
Unit	1/min	
Description of parameter values	Absolute speed difference	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00020	
Data type	Double0_1T_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

Speed window in percent

Speed window in percent	PercWin

Defines for each spindle Sp[1..60] the speed window for output of the spindle interface signal "Speed reached" (iSp_ProgSpReach).

If the balance of the actual spindle speed and the programmed spindle speed (scaled by the spindle override) is less than or equal to the % rate specified here, the control activates the spindle-specific interface signal "Spindle speed reached".

The values of the parameters AbsWin and PercWin are simultaneously effective; i.e. if either one of the two conditions is met, the signal "Spindle speed reached" will be set.

Path	SP/SpFunc/Sp[1..60]/SpeedWin/PercWin
Value range	0.00 .. 100.00
Pre-assignment	0.00
Unit	%
Description of parameter values	Speed difference in percent
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00021
Data type	Float0_100_t (float, 2 decimal places) 4-byte float, 2 decimal places for MACODA compatibility

Standstill window

Standstill window	StandstillWin

This parameter only applies to the spindle types (SP/SpFunc/Sp[1..60]/Base/SpType) "analog spindle with encoder" and "external spindle with encoder".

To detect spindle standstill, this parameter defines a threshold value, the standstill window, for each spindle Sp[1..60].

If the amount of the actual speed falls below this threshold, the control activates the spindle interface signal "Spindle stop" (iSp_Stop).

Path	SP/SpFunc/Sp[1..60]/SpeedWin/StandstillWin
Value range	0.000 .. 1000.000
Pre-assignment	10.000
Unit	1/min
Description of parameter values	Speed threshold
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00022

Spindles (SP) - Basic parameters

Data type Double0_1T_t (double, 3 decimal places)
8 byte float, 3 decimal places for MACODA compatibility

9.2.9 External spindle encoder

Drive number of the external spindle encoder

Drive number of the external spindle encoder		DrIndSpEnco
This parameter only applies to the spindle types (SP/SpFunc/Sp[1..60]/Base/SpType) "analog spindle with encoder" and "external spindle with encoder".		
Defines for each spindle Sp[1..60] the number of the Sercos drive transporting the encoder data of an "external" spindle to the NC.		
The Sercos parameter is determined by ScsParSpEnco.		
Path	SP/SpFunc/Sp[1..60]/Encoder/DrIndSpEnco	
Value range	0.. 250	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	No drive selected
	>0	Drive Number
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00081	
Data type	Dr_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Sercos Parameter of the External Spindle Encoder

Sercos parameter of the external spindle encoder		ScsParSpEnco
This parameter only applies to the spindle types (SP/SpFunc/Sp[1..60]/Base/SpType) "analog spindle with encoder" and "external spindle with encoder".		
Defines for each spindle Sp[1..60] the Sercos parameter of the encoder data of an "external" spindle.		
The drive number is determined by DrIndSpEnco.		
Path	SP/SpFunc/Sp[1..60]/Encoder/ScsParSpEnco	
Value range	<EmptyString>	
	S-0-xxxx or P-0-xxxx	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Sercos parameters	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	

Macoda ID 1040 00082
Data type Str16_t (isoLatin1String)
 2 x 8-byte string for MACODA compatibility

9.2.10 Jogging spindles

Jog speed[1..8]

Jogging spindles	JogSpeed[1..8]

Provides different speeds for the functions M3 and M4 for specific spindles Sp[1..60], from which a specific speed can be selected with the spindle interface signals "Jog speed Bit 0" (qSp_ManSpeed_00) to "Jog speed Bit 2" (qSp_ManSpeed_02).



Since spindle speeds are stored without any relation to gear steps, the spindle speed activated may be limited by the respective gear step.

Path	SP/SpFunc/Sp[1..60]/Jog/Speed/JogSpeed[1..8]	
Value range	0.000 .. 100000.000	
Pre-assignment	0.000	.../JogSpeed[1]
	1000.000	.../JogSpeed[2]
	2000.000	.../JogSpeed[3]
	3000.000	.../JogSpeed[4]
	4000.000	.../JogSpeed[5]
	5000.000	.../JogSpeed[6]
	6000.000	.../JogSpeed[7]
	7000.000	.../JogSpeed[8]
Unit	1/min	
Description of parameter values	Jog speeds for M3 or M4	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00015	
Data type	SpJogSpeed_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

Example:

Enabling the Respective Spindle Jog Speed via Spindle Interface (IF) Signals:

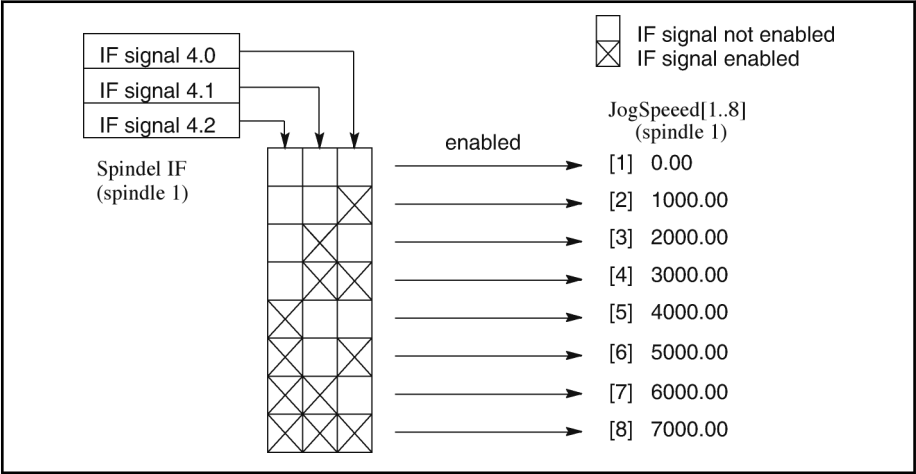


Fig. 9-4: jog speed via spindle interface (IF) signal

9.2.11 Spindle/Turret Axis

Turret mode and measuring system

Turret mode and measuring system		MeasSysSpTurrAx
Serves to adapt spindle/C-axis switching when using a combination of (cutter) spindle and tool turret on a lathe.		
Path	SP/SpFunc/Sp[1..60]/SpTurrAx/MeasSysSpTurrAx	
Value range	0	no tool turret (C-axis)
	1	tool turret with separate measuring system
	2	tool turret without separate measuring system
Pre-assignment	0	no tool turret (C-axis)
Unit	-	
Description of parameter values	0	no tool turret exists (normal mode)
	1	The functionality can currently not be used!
	2	The functionality is currently available in connection with IndraDrives. Minimum requirement: IndraDrive firmware MPx-04y03. The tool turret has its own gear step for driving the turret. The measuring system available in the motor is used jointly for both the spindle and the turret.
IndraDrive:		
In the drive, spindle mode is configured with the parameter sets 0 to 7 (1st to 8th gear). For turret mode, the first parameter set following the highest spindle parameter set used is relevant.		
An internally defined, permanent CPL variable (@_C_AXISxy_POS!) is used to secure the turret position when changing over to spindle mode. Upon changeover to turret mode, the axis position with the defining elements of this variable is loaded.		
Access rights	MTB:	Write

	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00060	
Data type	Byte0_2_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

9.2.12 Spindle actuation for channel [k=1..n]

Spindle can be actuated from the channel

Spindle can be actuated by the channel		EnablChCtrl
Actuating a spindle Sp[1..60] from a different channel Ch[k]. Any spindle receiving a movement instruction from a channel is reserved for that channel. A second channel can re-control this spindle only if this is permitted by the first channel. In exceptional cases, it can be necessary for more than one channel to have permission to randomly reverse a spindle. This spindle can be preset with this parameter. For more detailed information on spindle management, please refer to the manual "MTX 15VRS Functional Description - Basics" (DOK-MTX***-NC*F*BA*V15-RERS-EN-P). Also refer to the description of the NC functions "SpAdmin", "SSpAdm" and "SSpAdmOff" in the "MTX 15VRS Programming Manual" (DOK-MTX***-NC**PRO*V15-RERS-EN-P).		
Path	SP/SpFunc/Sp[1..60]/SpCtrl/Ch[k]/EnablChCtrl	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Cannot drive the active spindle Sp[1..60] from a different channel.
	1	Yes
		Any channel Ch[k] marked with 1 can drive the particular spindle Sp[1..60] randomly (skipping the spindle management) with M3, M4, M5, M19, M4x and Sx.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00070	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

9.2.13 Spindle syntax for system/channel Spindles

Every system spindle running as a channel spindle is reserved a spindle syntax. Since up to 8 spindles can run on a channel, the spindle syntax covers 8 different setting options.

9.2.14 Spindle syntax[1..8]

Spindle clockwise rotation

Spindle Clockwise Rotation		SpCW
Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Clockwise rotation".		
<i>Please note the following for setting the individual parameters:</i>		
- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group. - In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60])), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])). - Every syntax has to be unique.		
Path	SP/SpFunc/SpSyn/Syn[1..8]/SpCW	
Value range	{Text}	
Pre-assignment	M103	.../Syn[1]/SpCW
	M203	.../Syn[2]/SpCW
	-	.../Syn[3..8]/SpCW
	Unit	-
Description of parameter values	Spindle syntax for "Clockwise rotation" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00101	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Spindle clockwise rotation - Coolant on

Spindle clockwise rotation - Coolant on	SpCWCool
Defines spindle syntax Syn[1..8] for every channel spindle (1 to 8) the function "Clockwise rotation - Coolant ON".	
<i>Please note the following for setting the individual parameters:</i>	
- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group. - In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60])), the function can be enabled over the spindle group as long as the syntax has been agreed	

there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])).

- Every syntax has to be unique.

Path	SP/SpFunc/SpSyn/Syn[1..8]/SpCWCool	
Value range	{Text}	
Pre-assignment	M113	.../Syn[1]/SpCWCool
	M213	.../Syn[2]/SpCWCool
	-	.../Syn[3..8]/SpCWCool
Unit	-	
Description of parameter values	Spindle syntax for "Clockwise rotation - Coolant ON" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00102	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Spindle counterclockwise rotation

Spindle Counterclockwise Rotation	SpCCW

Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Counterclockwise rotation".

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see "Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60])), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])).
- Every syntax has to be unique.

Path	SP/SpFunc/SpSyn/Syn[1..8]/SpCCW	
Value range	{Text}	
Pre-assignment	M104	.../Syn[1]/SpCCW
	M204	.../Syn[2]/SpCCW
	-	.../Syn[3..8]/SpCCW
Unit	-	
Description of parameter values	Spindle syntax for "Counterclockwise rotation" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Spindles (SP) - Basic parameters

Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00103
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility

Spindle counterclockwise rotation - Coolant on

Spindle counterclockwise rotation - Coolant on		SpCCWCool
Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Counterclockwise rotation Coolant ON".		
<i>Please note the following for setting the individual parameters:</i>		
- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group. - In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60])), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])). - Every syntax has to be unique.		
Path	SP/SpFunc/SpSyn/Syn[1..8]/SpCCWCool	
Value range	{Text}	
Pre-assignment	M114	.../Syn[1]/SpCCWCool
	M214	.../Syn[2]/SpCCWCool
	-	.../Syn[3..8]/SpCCWCool
Unit	-	
Description of parameter values	Spindle syntax for "Counterclockwise rotation - Coolant ON" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00104	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Spindle stop

Spindle stop	SpStop
Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Spindle stop".	
<i>Please note the following for setting the individual parameters:</i>	
If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.	

- In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/)), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])).
- Every syntax has to be unique.

Path	SP/SpFunc/SpSyn/Syn[1..8]/SpStop		
Value range	{Text}		
Pre-assignment	M105	.../Syn[1]/SpStop	
	M205	.../Syn[2]/SpStop	
	-	.../Syn[3..8]/SpStop	
Unit	-		
Description of parameter values	Spindle syntax for "Spindle stop" function		
Access rights	MTB:	Write	
	Setter:	Read	
	Operator:	Read	
Activation	System restart (CS)		
Dependency	-		
Macoda ID	1040 00105		
Data type	Str8_t (isoLatin1String)		
	8 byte string for MACODA compatibility		

Spindle positioning

Spindle Positioning	SpPos

Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Spindle positioning" (orient spindle).

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/)), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])).
- Every syntax has to be unique.

Path	SP/SpFunc/SpSyn/Syn[1..8]/SpPos		
Value range	{Text}		
Pre-assignment	M119	...	/Syn[1]/SpPos
	M219	...	/Syn[2]/SpPos
	-	...	/Syn[3..8]/SpPos
Unit	-		
Description of parameter values	Spindle syntax for "Spindle positioning" function		

Spindles (SP) - Basic parameters

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00106	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Automatic gear selection

Automatic gear selection		SpGearAutoSel
Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Automatic gear select".		
<i>Please note the following for setting the individual parameters:</i>		
- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group. - In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/)), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])). - Every syntax has to be unique.		
Path	SP/SpFunc/SpSyn/Syn[1..8]/SpGearAutoSel	
Value range	{Text}	
Pre-assignment	M140	.../Syn[1]/SpGearAutoSel
	M240	.../Syn[2]/SpGearAutoSel
	-	.../Syn[3..8]/SpGearAutoSel
Unit	-	
Description of parameter values	Spindle syntax for "Automatic gear select" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00107	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Manual selection gear step 1

Manual selection gear step 1	SpGear1Sel
Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Manual gear select GTS 1".	

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/)), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])).
- Every syntax has to be unique.

Path	SP/SpFunc/SpSyn/Syn[1..8]/SpGear1Sel		
Value range	{Text}		
Pre-assignment	M141	.../Syn[1]/SpGear1Sel	
	M241	.../Syn[2]/SpGear1Sel	
	-	.../Syn[3..8]/SpGear1Sel	
Unit	-		
Description of parameter values	Spindle syntax for "Manual gear select GTS 1" function		
Access rights	MTB:	Write	
	Setter:	Read	
	Operator:	Read	
Activation	System restart (CS)		
Dependency	-		
Macoda ID	1040 00108		
Data type	Str8_t (isoLatin1String)		
	8 byte string for MACODA compatibility		

Manual selection gear step 2

Manual selection gear step 2	SpGear2Sel

Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Manual gear select GTS 2".

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/)), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])).
- Every syntax has to be unique.

Path	SP/SpFunc/SpSyn/Syn[1..8]/SpGear2Sel		
Value range	{Text}		
Pre-assignment	M142	.../Syn[1]/SpGear2Sel	

Spindles (SP) - Basic parameters

	M242	.../Syn[2]/SpGear2Sel
	-	.../Syn[3..8]/SpGear2Sel
Unit	-	
Description of parameter values	Spindle syntax for "Manual gear select GTS 2" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00109	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Manual selection gear step 3

Manual selection gear step 3		SpGear3Sel
Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Manual gear select GTS 3".		
<i>Please note the following for setting the individual parameters:</i>		
- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group. - In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/)), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])). - Every syntax has to be unique.		
Path	SP/SpFunc/SpSyn/Syn[1..8]/SpGear3Sel	
Value range	{Text}	
Pre-assignment	M143	.../Syn[1]/SpGear3Sel
	M243	.../Syn[2]/SpGear3Sel
	-	.../Syn[3..8]/SpGear3Sel
Unit	-	
Description of parameter values	Spindle syntax for "Manual gear select GTS 3" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00110	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Manual selection gear step 4

Manual selection gear 4	SpGear4Sel

Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Manual gear select GTS 4".

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60])), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])).
- Every syntax has to be unique.

Path	SP/SpFunc/SpSyn/Syn[1..8]/SpGear4Sel	
Value range	{Text}	
Pre-assignment	M144	.../Syn[1]/SpGear4Sel
	M244	.../Syn[2]/SpGear4Sel
	-	.../Syn[3..8]/SpGear4Sel
Unit	-	
Description of parameter values	Spindle syntax for "Manual gear select GTS 4" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00111	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Gear idle selection (disengaged)

Disengage gear (idle gear)	SpGearIdleSel

Defines for every channel spindle (1 to 8) in the channel the spindle syntax Syn[1..8] for the function "Disengage - select idle gear".

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- In the case of spindles without assigned syntax, individual programming of this function is disabled. If the respective spindle is assigned to a spindle group, however, (see Assigned spindle group" SpGrInd" (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60])), the function can be enabled over the spindle group as long as the syntax has been agreed there (see "Spindle clockwise rotation" SpGrCW (in the folder SP/SpGrFunc/SpGrSyn[1..4])).

Spindles (SP) - Basic parameters

		Every syntax has to be unique.
Path	SP/SpFunc/SpSyn/Syn[1..8]/SpGearIdleSel	
Value range	{Text}	
Pre-assignment	M148	.../Syn[1]/SpGearIdleSel
	M248	.../Syn[2]/SpGearIdleSel
	-	.../Syn[3..8]/SpGearIdleSel
Unit	-	
Description of parameter values	Spindle syntax for "Disengage - select idle gear" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00112	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

9.3 Spindle group functions (SpGrFunc)

9.3.1 General information

All spindles grouped into one of the spindle groups (1..4) follow the corresponding spindle group syntax[1..4].

9.3.2 Spindle group syntax[1..4]

Spindle clockwise rotation

Spindle Clockwise Rotation		SpGrCW
Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Clockwise rotation". Which spindles are to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/). The assignment can be changed for specific channels in the part program. <i>Please note the following for setting the individual parameters:</i> - If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group. - Spindle groups without assigned syntax cannot be programmed with the corresponding function. - Every syntax has to be unique.		
Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrCW	
Value range	{Text}	
Pre-assignment	M3	.../SpGrSyn[1]/SpGrCW
	-	.../SpGrSyn[2..4]/SpGrCW
Unit	-	
Description of parameter values	Spindle group syntax for "Clockwise rotation" function	

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00201	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Spindle clockwise rotation - Coolant on

Spindle clockwise rotation - Coolant on	SpGrCWCool

Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Clockwise rotation - Coolant ON".

The spindles to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/).

The assignment can be changed for specific channels in the part program.

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- Spindle groups without assigned syntax cannot be programmed with the corresponding function.
- Every syntax has to be unique.

Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrCWCool	
Value range	{Text}	
Pre-assignment	M13	.../SpGrSyn[1]/SpGrCWCool
	-	.../SpGrSyn[2..4]/SpGrCWCool
Unit	-	
Description of parameter values	Spindle group syntax for "Clockwise rotation - Coolant ON" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00202	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Spindle counterclockwise rotation

Spindle Counterclockwise Rotation	SpGrCCW

Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Counterclockwise rotation".

Which spindles are to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/).

Spindles (SP) - Basic parameters

The assignment can be changed for specific channels in the part program.

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- Spindle groups without assigned syntax cannot be programmed with the corresponding function.
- Every syntax has to be unique.

Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrCCW	
Value range	{Text}	
Pre-assignment	M4	.../SpGrSyn[1]/SpGrCCW
	-	.../SpGrSyn[2..4]/SpGrCCW
Unit	-	
Description of parameter values	Spindle group syntax for "Counterclockwise rotation" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00203	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Spindle counterclockwise rotation - Coolant on

Spindle counterclockwise rotation - Coolant on	SpGrCCWCool

Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Counterclockwise rotation - Coolant ON".

The spindles to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/).

The assignment can be changed for specific channels in the part program.

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- Spindle groups without assigned syntax cannot be programmed with the corresponding function.
- Every syntax has to be unique.

Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrCCWCool	
Value range	{Text}	
Pre-assignment	M14	.../SpGrSyn[1]/SpGrCCWCool
	-	.../SpGrSyn[2..4]/SpGrCCWCool
Unit	-	
Description of parameter values	Spindle group syntax for "Counterclockwise rotation - Coolant ON" function	
Access rights	MTB:	Write

	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00204	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Spindle stop

Spindle stop	SpGrStop

Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Spindle stop".

The spindles to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/).

The assignment can be changed for specific channels in the part program.

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- Spindle groups without assigned syntax cannot be programmed with the corresponding function.
- Every syntax has to be unique.

Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrStop	
Value range	{Text}	
Pre-assignment	M5	.../SpGrSyn[1]/SpGrStop
	-	.../SpGrSyn[2..4]/SpGrStop
Unit	-	
Description of parameter values	Spindle group syntax for "Spindle stop" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00205	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Spindle positioning

Spindle Positioning	SpGrPos

Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Spindle positioning" (orient spindle).

The spindles to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/).

The assignment can be changed for specific channels in the part program.

Spindles (SP) - Basic parameters

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- Spindle groups without assigned syntax cannot be programmed with the corresponding function.
- Every syntax has to be unique.

Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrPos
Value range	{Text}
Pre-assignment	M19 .../SpGrSyn[1]/SpGrPos - .../SpGrSyn[2..4]/SpGrPos
Unit	-
Description of parameter values	Spindle group syntax for "Spindle positioning" function
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00206
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility

Automatic gear selection

Automatic gear selection		SpGrGearAutoSel
Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Automatic gear selection".		
The spindles to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/).		
The assignment can be changed for specific channels in the part program.		
<i>Please note the following for setting the individual parameters:</i>		
• If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group. • Spindle groups without assigned syntax cannot be programmed with the corresponding function. • Every syntax has to be unique.		
Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrGearAutoSel	
Value range	{Text}	
Pre-assignment	M40 .../SpGrSyn[1]/SpGearAutoSel - .../SpGrSyn[2..4]/SpGearAutoSel	
Unit	-	
Description of parameter values	Spindle group syntax for "Automatic gear selection" function	
Access rights	MTB: Write Setter: Read	

Operator:	Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00207
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility

Manual selection gear step 1

Manual selection gear step 1		SpGrGear1Sel
Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Manual selection gear step 1". The spindles to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/). The assignment can be changed for specific channels in the part program. <i>Please note the following for setting the individual parameters:</i> • If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group. • Spindle groups without assigned syntax cannot be programmed with the corresponding function. • Every syntax has to be unique.		
Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrGear1Sel	
Value range	{Text}	
Pre-assignment	M41 .../SpGrSyn[1]/SpGrGear1Sel - .../SpGrSyn[2..4]/SpGrGear1Sel	
Unit	-	
Description of parameter values	Spindle group syntax for "Manual gear selection GTS 1" function	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00208	
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility	

Manual selection gear step 2

Manual selection gear step 2	SpGrGear2Sel
Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Manual selection gear step 2". The spindles to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/). The assignment can be changed for specific channels in the part program.	

Spindles (SP) - Basic parameters

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- Spindle groups without assigned syntax cannot be programmed with the corresponding function.
- Every syntax has to be unique.

Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrGear2Sel		
Value range	{Text}		
Pre-assignment	M42	.../SpGrSyn[1]/SpGrGear2Sel	
	-	.../SpGrSyn[2..4]/SpGrGear2Sel	
Unit	-		
Description of parameter values	Spindle group syntax for "Manual gear select GTS 2" function		
Access rights	MTB:	Write	
	Setter:	Read	
	Operator:	Read	
Activation	System restart (CS)		
Dependency	-		
Macoda ID	1040 00209		
Data type	Str8_t (isoLatin1String)		
	8 byte string for MACODA compatibility		

Manual selection gear step 3

Manual selection gear step 3		SpGrGear3Sel
Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Manual selection gear step 3".		
The spindles to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/).		
The assignment can be changed for specific channels in the part program.		
<i>Please note the following for setting the individual parameters:</i>		
• If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group. • Spindle groups without assigned syntax cannot be programmed with the corresponding function. • Every syntax has to be unique.		
Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrGear3Sel	
Value range	{Text}	
Pre-assignment	M43	.../SpGrSyn[1]/SpGrGear3Sel
	-	.../SpGrSyn[2..4]/SpGrGear3Sel
Unit	-	
Description of parameter values	Spindle group syntax for "Manual gear select GTS 3" function	
Access rights	MTB:	Write
	Setter:	Read

Operator:	Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00210
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility

Manual selection gear step 4

Manual selection gear 4		SpGrGear4Sel
Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Manual selection gear step 4". The spindles to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/). The assignment can be changed for specific channels in the part program. <i>Please note the following for setting the individual parameters:</i> • If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group. • Spindle groups without assigned syntax cannot be programmed with the corresponding function. • Every syntax has to be unique.		
Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrGear4Sel	
Value range	{Text}	
Pre-assignment	M144	.../SpGrSyn[1]/SpGrGear4Sel
	-	.../SpGrSyn[2..4]/SpGrGear4Sel
Unit	-	
Description of parameter values	Spindle group syntax for "Manual gear select GTS 4" function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00211	
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility	

Gear idle selection (disengaged)

Disengage gear (idle gear)	SpGrGearIdleSel
Defines for each spindle group (1 to 4) the spindle syntax SpGrSyn[1..4] for the function "Disengage - select idle gear". The spindles to be assigned to the individual spindle groups is specified in the parameter "Assigned spindle group" SpGrInd (in the folder SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/). The assignment can be changed for specific channels in the part program.	

Spindles (SP) - Basic parameters

Please note the following for setting the individual parameters:

- If the function is also to be output as an auxiliary function on the interface, the respective syntax must be entered as an auxiliary function in the "Auxiliary functions" (AUXF) data group.
- Spindle groups without assigned syntax cannot be programmed with the corresponding function.
- Every syntax has to be unique.

Path	SP/SpGrFunc/SpGrSyn[1..4]/SpGrGearIdleSel		
Value range	{Text}		
Pre-assignment	M48	...	/SpGrSyn[1]/SpGrGearIdleSel
	-	...	/SpGrSyn[2..4]/SpGrGearIdleSel
Unit	-		
Description of parameter values	Spindle group syntax for "Disengage - Idle gear" function		
Access rights	MTB:	Write	
	Setter:	Read	
	Operator:	Read	
Activation	System restart (CS)		
Dependency	-		
Macoda ID	1040 00212		
Data type	Str8_t (isoLatin1String)		
	8 byte string for MACODA compatibility		

9.4 Spindle management (SpAdmin)

9.4.1 General information

The system spindles are assigned in the spindle management. The context menu of the spindle management folder ("SpAdmin" under SP/) provides the following selection with **"Change in"**:

- **ChSp:** Assignment of the system spindle to the channel spindle on the corresponding channel and
- **Sp:** Assignment of the system spindle to the spindle group (compatibility mode)

Macoda ID 1040 00999 (0 = System spindle, 1 = Channel spindle)

From the context menu of "ChSp", you get the path SP/SpAdmin/ChSp/ChSpGrAlloc/Ch[k]/, under which the system spindles Sp[1..60] are assigned channel-specifically (Ch[k]) to the channel spindles ChSp[1..8]/.

From the context menu of "Sp", you get the path SP/SpAdmin/Sp/SpGrAlloc/, under which the system spindles Sp[1..60] are assigned to the spindle groups.

The spindle management provides the parameterization of the **block switching control** additionally to the spindle assignment. The block switching control is always provided irrespective of the context menu selection "ChSp" or "Sp". It allows that the continuous block switching in the machining channels becomes dependent on certain spindle states.

9.4.2 Spindles in the "ChSp" selection (channel spindle mode)

System spindles[1..60] - Spindle group[1..8]

For each channel Ch[1...60] in the channel spindle mode, up to 8 channel spindles ChSp[1..8] can be assigned from the existing system spindles Sp[1..60]. The assignment is carried out in the parameter **SpIndChSp** (path: SP/SpAdmin/ChSp/ChSpGrAlloc/Ch[k]/ChSp[1..8]).

To select between the compatibility mode (spindle groups) and the channel spindle mode, refer to [chapter 9.4.1 "General information" on page 152](#).

Assigned system spindle

	Assigned system spindle	SpIndChSp
	Assigns a system spindle Sp[1..60] to a channel spindle ChSp[1..8] in the channel Ch[k].	
	This preset can be changed in the part program by means of RemSpindle (RSP) and GetSpindle (GSP). The function DefSpindle (DSP) restores the preset.	
Path	SP/SpAdmin/ChSp/ChSpGrAlloc/Ch[k]/ChSp[1..8]/SpIndChSp	
Value range	0.. 60	
Pre-assignment	1	
Unit	-	
Description of parameter values	0	No channel spindle defined
	>0	Number of system spindles Sp[1..60] that become channel spindles ChSp[1..8] in the channel Ch[k].
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7020 00020	
Data type	Sp_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	
	<i>Example:</i>	
	System spindle assignment to channel spindles Sx	
	The following setting applies to channel Ch[2]:	
	ChSp[1]/SpIndChSp = 1 S1 is the 1st system spindle	
	ChSp[2]/SpIndChSp = 5 S2 is the 5th system spindle	
	ChSp[3]/SpIndChSp = 7 S3 is the 7th system spindle	

9.4.3 Spindles in the "Sp" selection (compatibility mode)

System spindles[1..60] - Spindle group[1..4]

In the compatibility mode, the system spindles Sp[1..60] are always assigned to one spindle group [1..4]. The assignment is carried out in the parameter **SpGrInd** (path: SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]).

To select between the compatibility mode (spindle groups) and the channel spindle mode, refer to [chapter 9.4.1 General information, page 152](#).

Assigned spindle group

Assigned spindle group	SpGrInd

Defines for each system spindle Sp[1..60] whether they are assigned to spindle groups and, if so, to which.



This parameter is only effective in compatibility mode (SP/SpAdmin/ → Sp)

No spindle may be assigned to more than one spindle group. A spindle group may include Sercos, analog and external spindles.

All spindles of a spindle group can be activated (programmed) together with the suitable spindle group syntax.

Restriction:

"External spindles" can only be programmed via the spindle group if the function concerned (e.g. "Spindle stop") has been parameterized as follows:

- as an auxiliary function (see AUXF)
- as spindle syntax (see SP/SpFunc/...)

Path	SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/SpGrInd	
Value range	0.. 4	
Pre-assignment	1	
Unit	-	
Description of parameter values	0	System spindle Sp[1..60] is not assigned to any spindle group
	>0	Spindle group 1, 2, 3 or 4 is assigned to system spindle Sp[1..60]
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00002	
Data type	SpGr_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

9.4.4 Block switching control

The block switching control allows enabling waiting conditions in connection with the spindle functions. These conditions delay the start of a path movement or the exiting of an NC block.

The following waiting conditions are possible:

- Waiting for "Speed reached" during active clockwise and counterclockwise rotation
- Waiting for "Spindle standstill" during active stop command
- Waiting for "Spindle oriented" during active spindle orientation
- Waiting for "Gear range changed"

- There are three waiting strategies: one predefined strategy and two strategies that can be configured by the machine manufacturer. Each strategy comprises individual settings for the mentioned waiting conditions.
- Behavior of the waiting strategies** The behavior of waiting strategy 1 is fixedly specified. For the waiting strategies 2 and 3, the respective behavior is configured in the folders **VarStrategy2** and **VarStrategy3** (path: SP/SpAdmin/BlkSwiCtrl/WaitingPoints/).
- Selecting the waiting strategy** One waiting strategy can be selected and activated in each channel for the individual system spindles/ channel spindles 1...8. This is carried out by the parameters **SelBlkSwiStrategy** and **EnablBlkSwiStrategy** (SP/SpAdmin/BlkSwiCtrl/Ch[k]/Sp_ChSp[1..8]/).

9.4.5 Waiting points in time for block switching control with strategy 2

Strategy 2: Speed reached

Strategy 2: Speed reached	Strat2WaitSpSpeed

It defines the waiting points in time of strategy 2 for the waiting on "Speed reached" during active clockwise or counterclockwise direction. There are entries for waiting while positioning, exiting the positioning and during machining.

Path	SP/SpAdmin/BlkSwiCtrl/WaitingPoints/VarStrategy2/Strat2WaitSpSpeed	
Value range	000:	never wait
	001:	Posit.:--, after pos. mode:--, machining: SP
	002:	Posit.:--, after pos. mode:--, machining: EP
	020:	Posit.:--, after pos. mode: EP, machining:--
	021:	Posit.:--, after pos. mode: EP, machining: SP
	022:	Posit.:--, after pos. mode: EP, machining: EP
	100:	Posit.: SP, after pos. mode: --, machining: --
	101:	Posit.: SP, after pos. mode: --, machining: SP
	102:	Posit.: SP, after pos. mode: --, machining: EP
	120:	Posit.: SP, After pos. mode: EP, machining:--
	121:	Posit.: SP, After pos. mode: EP, machining: SP
	122:	Posit.: SP, After pos. mode: EP, machining: EP
	200:	Posit.: EP, After pos. mode:--, machining.: --
	201:	Posit.: EP, After pos. mode:--, machining.: SP
	202:	Posit.: EP, After pos. mode:--, machining.: EP
	220:	Posit.: EP, After pos. mode: EP, machining:--
	221:	Posit.: EP, After pos. mode: EP, machining: SP
	222:	Posit.: EP, After pos. mode: EP, machining: EP
Pre-assignment	021:	Posit.:--, after pos. mode: EP, machining: SP
Unit	-	

Description of parameter values The values contain the following information:

Spindles (SP) - Basic parameters

- Hundredths: Wait while positioning (G0)
- Tenths: Wait when exiting the positioning (switching from G0 to G1/G2/...)
- Units digit: Wait while machining (G1, G2, ...)

The digits mean:

- 0: Do not wait
- 1: Wait at starting point (not possible for waiting when exiting the positioning)
- 2: Wait at end point

Combinations allowed:

- 000: **never wait**
Do not wait at any point.
- 001: **Posit.:--, after pos. mode:--, machining: SP**
Wait at starting point while machining.
- 002: **Posit.:--, after pos. mode:--, machining: EP**
Wait at end point while machining.
- 020: **Posit.:--, after pos. mode: EP, machining:--**
Wait at end point when exiting the positioning.
- 021: **Posit.:--, after pos. mode: EP, machining: EP**
Wait at end point when exiting the positioning and at starting point while machining.
- 022: **Posit.:--, after pos. mode: EP, machining: EP**
Wait at end point when exiting the positioning and while machining.
- 100: **Posit.: SP, after pos. mode: --, machining: --**
Wait at starting point when positioning.
- 101: **Posit.: SP, after pos. mode: --, machining: SP**
Wait at starting point while positioning and machining.
- 102: **Posit.: SP, after pos. mode: --, machining: EP**
Wait at starting point while positioning and at end point while machining.
- 120: **Posit.: SP, After pos. mode: EP, machining:--**
Wait at starting point while positioning and at end point when exiting the positioning.
- 121: **Posit.: SP, After pos. mode: EP, machining: SP**
Wait at starting point while positioning, at end point when exiting the positioning and at starting point while machining.
- 122: **Posit.: SP, After pos. mode: EP, machining: EP**
Wait at starting point while positioning, at end point when exiting the positioning and while machining.
- 200: **Posit.: EP, After pos. mode:--, machining.: --**
Wait at end point while positioning.
- 201: **Posit.: EP, After pos. mode:--, machining.: SP**
Wait at end point while positioning and at starting point while machining.

	202:	Posit.: EP, After pos. mode:--, machining.: EP Wait at end point while positioning and machining.
	220:	Posit.: EP, After pos. mode: EP, machining: -- Wait at end point while positioning and when exiting the positioning.
	221:	Posit.: EP, After pos. mode: EP, machining: SP Wait at end point while positioning, when exiting the positioning and at starting point while machining.
	222:	Posit.: EP, After pos. mode: EP, machining: EP Wait at end point while positioning, when exiting the positioning and while machining.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00300[1]	
Data type	UnsignedShort_t 2 byte integer, unsigned for MACODA compatibility	

Strategy 2: Spindle standstill

Strategy 2: Spindle standstill		Strat2WaitSpStop
	It defines the waiting points in time of strategy 2 for the waiting on "Spindle standstill" during active stop command. There are entries for waiting while positioning, exiting the positioning and during machining.	
Path	SP/SpAdmin/BlkSwiCtrl/WaitingPoints/VarStrategy2/Strat2WaitSpStop	
Value range	as for Strat2WaitSpSpeed, see Value range, page 155	
Pre-assignment	020:	Posit.:--, after pos. mode: EP, machining:--
Unit	-	
Description of parameter values	The values contain the following information: - Hundredths: Wait while positioning (G0) - Tenths: Wait when exiting the positioning (switching from G0 to G1/G2/...) - Units digit: Wait while machining (G1, G2, ...) The digits mean: 0: Do not wait 1: Wait at starting point (not possible for waiting when exiting the positioning) 2: Wait at end point Combinations allowed: as for Strat2WaitSpSpeed, see Meaning of parameter values, page 155	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Spindles (SP) - Basic parameters

Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00300[2]
Data type	UnsignedShort_t 2 byte integer, unsigned for MACODA compatibility

Strategy 2: Spindle oriented

Strategy 2: Spindle oriented		Strat2WaitSpOri
It defines the waiting points in time of strategy 2 for the waiting on "Spindle oriented" during active spindle orientation. There are entries for waiting while positioning, exiting the positioning and during machining.		
Path	SP/SpAdmin/BlkSwiCtrl/WaitingPoints/VarStrategy2/Strat2WaitSpOri	
Value range	as for Strat2WaitSpSpeed, see Value range, page 155	
Pre-assignment	020: Posit.:--, after pos. mode: EP, machining:--	
Unit	-	
Description of parameter values	The values contain the following information: • Hundredths: Wait while positioning (G0) • Tenths: Wait when exiting the positioning (switching from G0 to G1/G2/...) • Units digit: Wait while machining (G1, G2, ...) The digits mean: 0: Do not wait 1: Wait at starting point (not possible for waiting when exiting the positioning) 2: Wait at end point Permitted combinations: As for Strat2WaitSpSpeed, see Meaning of parameter values, page 155	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00300[3]	
Data type	UnsignedShort_t 2 byte integer, unsigned for MACODA compatibility	

Strategy 2: Gear step changed

Strategy 2: Gear range changed		Strat2WaitSpGear
Defines the waiting points in time of strategy 2 for waiting for "Gear stage changed". There are entries for waiting while positioning, exiting the positioning and during machining.		
Path	SP/SpAdmin/BlkSwiCtrl/WaitingPoints/VarStrategy2/Strat2WaitSpGear	
Value range	as for Strat2WaitSpSpeed, see Value range, page 155	
Pre-assignment	021: Posit.:--, after pos. mode: EP, machining:SP	
Unit	-	

Description of parameter values	The values contain the following information: - Hundredths: Wait while positioning (G0) - Tenths: Wait when exiting the positioning (switching from G0 to G1/G2/...) - Units digit: Wait while machining (G1, G2, ...) The digits mean: 0: Do not wait 1: Wait at starting point (not possible for waiting when exiting the positioning) 2: Wait at end point Permitted combinations: As for Strat2WaitSpSpeed, see Meaning of parameter values, page 155
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00300[4]
Data type	UnsignedShort_t 2 byte integer, unsigned for MACODA compatibility

9.4.6 Waiting points in time for block switching control with strategy 3

Strategy 3: Speed reached

Strategy 3: Speed reached	Strat3WaitSpSpeed
Path SP/SpAdmin/BlkSwiCtrl/WaitingPoints/VarStrategy3/Strat3WaitSpSpeed Value range as for Strat2WaitSpSpeed, see Value range, page 155 Pre-assignment 021: Posit.--, after pos. mode: EP, machining: SP Unit -	
Description of parameter values	The values contain the following information: - Hundredths: Wait while positioning (G0) - Tenths: Wait when exiting the positioning (switching from G0 to G1/G2/...) - Units digit: Wait while machining (G1, G2, ...) The digits mean: 0: Do not wait 1: Wait at starting point (not possible for waiting when exiting the positioning) 2: Wait at end point Permitted combinations: As for Strat2WaitSpSpeed, see Meaning of parameter values, page 155

Spindles (SP) - Basic parameters

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00300[1]	
Data type	UnsignedShort_t	
	2 byte integer, unsigned for MACODA compatibility	

Strategy 3: Spindle standstill

Strategy 3: Spindle standstill		Strat3WaitSpStop
It defines the waiting points in time of strategy 3 for the waiting on "Spindle standstill" during active stop command. There are entries for waiting while positioning, exiting the positioning and during machining.		
Path	SP/SpAdmin/BlkSwiCtrl/WaitingPoints/VarStrategy3/Strat3WaitSpStop	
Value range	as for Strat2WaitSpSpeed, see Value range, page 155	
Pre-assignment	020: Posit.--, after pos. mode: EP, machining:--	
Unit	-	
Description of parameter values	The values contain the following information:	
	• Hundredths:	Wait while positioning (G0)
	• Tenths:	Wait when exiting the positioning (switching from G0 to G1/G2/...)
	• Units digit:	Wait while machining (G1, G2, ...)
	The digits mean:	
	0:	Do not wait
	1:	Wait at starting point (not possible for waiting when exiting the positioning)
	2:	Wait at end point
	Combinations allowed:	
	as for Strat2WaitSpSpeed, see Meaning of parameter values, page 155	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00300[2]	
Data type	UnsignedShort_t	
	2 byte integer, unsigned for MACODA compatibility	

Strategy 3: Spindle oriented

Strategy 3: Spindle oriented	Strat3WaitSpOri
It defines the waiting points in time of strategy 3 for the waiting on "Spindle oriented" during active spindle orientation. There are entries for waiting while positioning, exiting the positioning and during machining.	

Path	SP/SpAdmin/BlkSwiCtrl/WaitingPoints/VarStrategy3/Strat3WaitSpOri
Value range	as for Strat2WaitSpSpeed, see Value range, page 155
Pre-assignment	021: Posit.:--, after pos. mode: EP, machining: SP
Unit	-
Description of parameter values	The values contain the following information: - Hundredths: Wait while positioning (G0) - Tenths: Wait when exiting the positioning (switching from G0 to G1/G2/...) - Units digit: Wait while machining (G1, G2, ...) The digits mean: 0: Do not wait 1: Wait at starting point (not possible for waiting when exiting the positioning) 2: Wait at end point Combinations allowed: as for Strat2WaitSpSpeed, see Meaning of parameter values, page 155
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1040 00300[3]
Data type	UnsignedShort_t
	2 byte integer, unsigned for MACODA compatibility

Strategy 3: Gear range changed

Strategy 3: Gear range changed		Strat3WaitSpGear
Defines the waiting points in time of strategy 3 for waiting for "Gear stage changed". There are entries for waiting while positioning, exiting the positioning and during machining.		
Path	SP/SpAdmin/BlkSwiCtrl/WaitingPoints/VarStrategy3/Strat3WaitSpGear	
Value range	as for Strat2WaitSpSpeed, see Value range, page 155	
Pre-assignment	021: Posit.:--, after pos. mode: EP, machining:SP	
Unit	-	
Description of parameter values	The values contain the following information: - Hundredths: Wait while positioning (G0) - Tenths: Wait when exiting the positioning (switching from G0 to G1/G2/...) - Units digit: Wait while machining (G1, G2, ...) The digits mean: 0: Do not wait 1: Wait at starting point (not possible for waiting when exiting the positioning) 2: Wait at end point	

Spindles (SP) - Basic parameters

	Combinations allowed:	
	as for Strat2WaitSpSpeed, see Meaning of parameter values, page 155	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1040 00300[4]	
Data type	UnsignedShort_t	
	2 byte integer, unsigned for MACODA compatibility	

9.4.7 Strategy Determination for Block Switching Control

Selecting the Block Switching Strategy

Selection of block switching strategy		SelBlkSwiStrategy
	It selects the strategy of the block switching control channel-specifically for the respective system spindle/channel spindle 1...8. The block switching control allows enabling waiting conditions in connection with the spindle functions. These conditions delay the start of a path movement or the exiting of an NC block.	
Path	SP/SpAdmin/BlkSwiCtrl/Ch[k]/Sp_ChSp[1..8]/SelBlkSwiStrategy	
Value range	0	No waiting
	1	Strategy 1 (fixedly set)
	2	Strategy 2 (variable)
		Strategy 3 (variable)
Pre-assignment	1	Strategy 1 (fixedly set)
Unit	-	
Description of parameter values	0	No waiting Not expected during block switching.
	1	Strategy 1 (fixedly set) It selects the fixedly set block switching strategy 1. In case of strategy 1, it is waited for the reaching of the programmed spindle state during the positioning mode when exiting the positioning mode. In case of an active processing, the path motion is stopped if there is a gear shift.
	2	Strategy 2 (variable) It selects the block switching strategy 2 defined by the machine manufacturer (see SP/SpAdmin/BlkSwiCtrl/WaitingPoints/VarStrategy2/...).
	3	Strategy 3 (variable) It selects the block switching strategy 3 defined by the machine manufacturer (see SP/SpAdmin/BlkSwiCtrl/WaitingPoints/VarStrategy3/...).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Activation	System reset (SR)
Dependency	A block switching strategy selected here is only applied if it is enabled with EnablBlkSwiStrategy or with the NC function SpindleWaitMode/SWM.
Macoda ID	7020 00030
Data type	Byte0_3_t 1 byte integer, unsigned for MACODA compatibility

Enabling Block Switching Strategy

Enable block switching strategy		EnablBlkSwiStrategy
It enables the strategy of the block switching control channel-specifically for the respective system spindle/channel spindle 1...8. The block switching control allows enabling waiting conditions in connection with the spindle functions. These conditions delay the start of a path movement or the exiting of an NC block.		
Path	SP/SpAdmin/BlkSwiCtrl/Ch[k]/Sp_ChSp[1..8]/EnablBlkSwiStrategy	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		It disables the waiting strategy selected in SelBlkSwiStrategy.
	1	Yes
		It enables the waiting strategy selected in SelBlkSwiStrategy.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	The block switching strategy is selected in SelBlkSwiStrategy.	
Macoda ID	7020 00040	
Data type	Byte0_1_t	
	1 byte integer, unsigned for MACODA compatibility	

10 Sercos communication (SCS) - Basic parameters -

10.1 General information

This chapter describes the settings of the Sercos interface between drives and MTX:

- [Sercos ring, page 165](#)
- [Sercos data handling, page 166](#)
- [Virtual operation, page 167](#)
- [Test operation, page 168](#)

10.2 Sercos Ring (ScsIf)

10.2.1 General information

The MTX only supports one Sercos ring (ring 1). The general configuration is with the parameters in the folder SCS/ScsIf[1].

10.2.2 Sercos Cycle Time (= Interpolation Cycle Time)

Sercos Cycle Time (= Interpolation Cycle Time)	ScsCycTime

Determines the interpolation cycle of the MTX.

The lower limit of the value range (100 µs) as well as the upper limit (64000 µs) are not important for the entry. The lowest value that can be entered depends on the output class of the control (e.g. MTX performance: 250 µs).

The greatest value that can be entered with the MTX is generally 8000 µs.

Path SCS/ScsIf[1]/ScsCycTime

Value range 100 .. 64000 (please note the explanations above!)

Pre-assignment 4000

Unit µs

Description of parameter values Time pattern for output of the interpolator command values.



The Sercos cycle time that is used is based on this value.

The following applies: Sercos cycle time in µs = NC cycle time in µs

Access rights **MTB:** Write

Setter: Read

Operator: Read

Activation System restart (CS)

Dependency -

Macoda ID 9030 00001

Data type CycTime_t (unsignedInt)
2 byte integer, unsigned for MACODA compatibility

10.2.3 Time monitoring for Sercos startup

Time monitoring for Sercos startup	ScsTimeOut

Defines for the Sercos ring how long the control shall wait until the ring is started up (i.e. until all devices in the ring have reached "Phase 4").

Sercos communication (SCS) - Basic parameters -

 The time required for Sercos startup depends on the number of drives configured.	
Path	SCS/ScsIf[1]/ScsTimeOut
Value range	0 .. 1,000 0 .. 999 for MACODA compatibility
Pre-assignment	60
Unit	s
Description of parameter values	Control timeout after which the command value can be output over the Sercos interface.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1050 00021
Data type	Int0_1T_t (unsignedInt) 4 byte integer, unsigned for MACODA compatibility

10.3 Sercos Data Handling (Data Handl)

10.3.1 Downloading Sercos Data in Phase 3

Download Sercos data in phase 3		ModeScsFiles
For Sercos phase 3, the control saves parameters in the drive in order to accelerate the Sercos startup.		
Path	SCS/DataHandl/ModeScsFiles	
Value range	0	execute always
	1	only after a change
Pre-assignment	0	execute always
Unit	-	
Description of parameter values	0	execute always Parameterization is with SCS files.
	1	only after a change Parameters are saved in the drive; SCS files are downloaded only when parameters are modified.



Parameters saved in the drive are retained when the control is switched off if the command "Save main memory" is executed beforehand.

In "DIAGNOSTICS" mode, press the buttons:

Save <Drive diagnostics> <Sercos monitor> <Drive data>
back up.

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1050 00022	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

10.3.2 Sercos ID for NC command values

Sercos ID for NC command Values		ScsIdCmdVal
Specifies for each axis Dr[i] with which Sercos parameters the position command values from the NC shall be transmitted to the drive.		
Path	SCS/DataHandl/Dr[i]/ScsIdCmdVal	
Value range	0	S-0-0047
	1	P-0-1370
Pre-assignment	0	S-0-0047
Unit	-	
Description of parameter values	0	S-0-0047
		Transmission of the NC command value via Sercos ID S-0-0047
	1	P-0-1370
		Transmission of the NC command value via Sercos ID P-0-1370
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1050 00050	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

10.4 Virtual mode (VirtMode)

10.4.1 General information

The settings in "Virtual mode" (in the folder SCS/VirtMode/) are drive-specific (Dr[i]).

10.4.2 Position of the reference point

Position of reference point	RefPos
Specifies for every virtual drive (MAIN/Dr[i]/EnablVirtMode = Yes) the reference point for the functions:	
Move to reference point coordinates (G74) and	

Sercos communication (SCS) - Basic parameters -

		move to reference point in drive-controlled mode (asynchronously and via "G74(Home)").
Path	SCS/VirtMode/Dr[i]/RefPos	
Value range	-1000000.000 .. 1000000.000 -999999.998 .. 999999.998, 999999.999 (MACODA compatibility)	
Pre-assignment	0.000	
Unit	mm or ° (degrees)	
Description of parameter values	999999.999	No reference point position of the virtual drive Dr[i] exists
	Otherwise	Reference point position of the virtual drive Dr[i] exists
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1015 00110	
Data type	Pos_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

10.5 Test mode

10.5.1 Enabling the test mode

Enabling test mode		EnablTestMode
This parameter allows the control to be immediately switched to Test mode while it is starting up. In addition, the interrupt load that causes the Sercos interrupt can be simulated for the test mode. When this parameter is set to "Yes" or to "Yes, and simulate Sercos interrupt load", time monitoring of the Sercos startup is switched off. During simulation of the Sercos interrupt load, the running time of the interrupt routine is determined depending on the axis configuration.		
Path	SCS/TestMode/EnablTestMode	
Value range	0	No
	1	Yes
	2	No, but simulate Sercos interrupt load
	3	(Yes, and simulate Sercos interrupt load
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No Test mode not activated If this is set, the control switches to Test mode only if not all Sercos devices have attained Sercos phase 4 after the waiting time set in parameter ScsTimeOut has elapsed. The Sercos interrupt load is not simulated.
	1	Yes Test mode activated

			The control immediately switches to Test mode at startup. The Sercos interrupt load is not simulated.
	2		No, but simulate Sercos interrupt load Test mode not activated
			If this is set, the control switches to Test mode only if not all Sercos devices have attained Sercos phase 4 after the waiting time set in parameter ScsTimeOut has elapsed. The Sercos interrupt load is simulated only when switching to test mode.
	3		(Yes, and simulate Sercos interrupt load Test mode activated The control immediately switches to Test mode and simulates the Sercos interrupt load at startup.
Access rights	MTB:	Write	
	Setter:	Write	
	Operator:	Read	
Activation	System restart (CS)		
Dependency	-		
Macoda ID	1050 00020		
Data type	Byte0_3_t (byte)		
	1 byte integer, unsigned for MACODA compatibility		

11 Sercos parameters (SCSP) - Basic parameters -

11.1 General information

This chapter describes the drive-specific settings MTX require for automatic parameterization of Sercos drives:

- [General Sercos settings, page 171](#)
- [Dynamic limitations, page 173](#)
- [Telegrams, page 174](#)

The SCSP parameters are only used in the "IndraDrive" drive type (see machine parameter MAIN/Dr[i]/Scs/DrType, Macoda-ID 1050 00001). In this case, manually created scs files (see documentation "MTX Commissioning") are not required. The scs parameters are generated automatically using the machine parameters and are written to the drive.

11.2 General Sercos settings

11.2.1 Storage mode

Storage mode		StorMode
	Specifies whether and how the MTX sets the saving mode in the drive (S-0-0269) during startup. Depending on the selected setting in the drive, changed drive parameters are automatically saved as resident or not saved as resident.	
Path	SCSP/Dr[i]/Gen/StorMode	
Value range	0	as preset in drive
	1	After each change
	2	with command S-0-0264
Pre-assignment	0	as preset in drive
Unit	-	
Description of parameter values	0	as preset in drive
		With this setting, the MTX does not apply any changes to parameter S-0-0269 in the drive.
	1	After each change
		With this setting, parameter S-0-0269 saves changed drive parameters as resident in the drive.
	2	with command S-0-0264
		With this setting, parameter S-0-0269 does not save changed drive parameters as resident in the drive. This means, the drive parameters will be lost after switch off. Execute the command "S-0-0264, C2200 Command Backup RAM" to retain the parameter settings in case of power failure.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	-	

Sercos parameters (SCSP) - Basic parameters -

Sercos ID S-0-0269

Data type Byte0_2_t (byte)

11.2.2 Encoder selection

Encoder selection		Enco
Specifies if the drive uses a motor encoder or an external encoder.		
Path	SCSP/Dr[i]/Gen/Enco	
Value range	0	Motor encoder
	1	External encoder
Pre-assignment	0	Motor encoder
Unit	-	
Description of parameter values	The encoder selection (motor encoder or external encoder) influences several parameters in the automatic parameterization of Sercos drives by the MTX.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	-	
Data type	Byte0_1_t (byte)	

11.2.3 Multiplex channel

Multiplex channel		MpxCh
Specifies if a multiplex channel is to be created for the drive.		
Path	SCSP/Dr[i]/Gen/MpxCh	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No No multiplex channel is created
	1	Yes A cyclic multiplex container is automatically created. The user only has to configure the parameter lists S-0-0370 and S-0-0371 (see "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P), chapter "Functions for drive parameterization").
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	-	
Data type	Byte0_1_t (byte)	

11.3 Dynamic limitations

11.3.1 Position controller Kv factor

Servo gain factor (Kv) of position controller		PosLoopKv
Specifies whether and how the MTX sets the Kv factor of the position controller in the drive (S-0-0104) during startup.		
Path	SCSP/Dr[i]/Lim/PosLoopKv	
Value range	0.00 .. 655.35	
Pre-assignment	0.00	
Unit	1000/min	
Description of parameter values	0.00	The MTX does not apply any changes of drive parameter S-0-0104
	Otherwise	The drive parameter S-0-0104 is set to this start value
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	-	
Sercos ID	S-0-0104	
Data type	PosLoopKv_t (double, 2 decimal places)	

11.3.2 Torque limit value

Torque limit value		TrqLim
Specifies whether and how the MTX sets the torque limit value in the drive (torque/force limit value bipolar, S-0-0092) during startup.		
Path	SCSP/Dr[i]/Lim/TrqLim	
Value range	0.0 .. 1000.0	
Pre-assignment	0.0	
Unit	%	
Description of parameter values	0.00	The MTX does not apply any changes of drive parameter S-0-0092
	Otherwise	The drive parameter S-0-0092 is set to this start value
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	-	
Sercos ID	S-0-0092	
Data type	TrqLim_t (double, 1 decimal place)	

11.4 Telegrams

11.4.1 User parameter display

User parameter display		DispUsrPara
	Enables or disables the additional user-specific master data telegram settings, the drive telegram, the signal status word and the signal control word for the relevant drive. The default configuration lists are generated automatically. The user can add user-defined additional list elements as long as the maximum list length is complied with.	
Path	SCSP/Dr[i]/Teleg/DispUsrPara	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No
	The additional user-specific settings of the configuration lists are disabled.	
	The "MDT" (master data telegram) folder, "AT" (drive telegram), "SigStatW" (signal status word) and "SigCtrlW" (signal control word) are grayed out and cannot be opened.	
	1	Yes
	The additional user-specific settings of the configuration lists are enabled.	
	The "MDT" (master data telegram) folder, "AT" (drive telegram), "SigStatW" (signal status word) and "SigCtrlW" (signal control word) can be opened and edited.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	-	
Data type	Byte0_1_t (byte)	

11.4.2 Master data telegram

Master data telegram entry

Master data telegram entry		MDTConf
	Contains the user-specific settings of the additive MDT configuration list (S-0-1050.0.6).	
Path	SCSP/Dr[i]/Teleg/MDT/MDTConf[1..16]	
Value range	<S-x-xxxx.xxx.xxx> (x = [0..9], max. 16 characters)	
	<P-x-xxxx.xxx.xxx>	
	<xxxx> short for <S-0-xxxx>	
Pre-assignment	<EmptyString>	
Unit	-	

Description of parameter values	Entry (Sercos ID) for the additive MDT configuration list
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	-
Sercos ID	S-0-1050.0.6
Data type	CfgSCSP_ScsIdn_t (16 characters + string end)

11.4.3 Drive telegram

Drive telegram entry

Drive telegram entry		ATConf
	Contains the user-specific settings of the additive AT configuration list (S-0-1050.1.6).	
Path	SCSP/Dr[i]/Teleg/AT/ATConf[1..16]	
Value range	<S-x-xxxx.xxx.xxx> (x = [0..9], max. 16 characters) <P-x-xxxx.xxx.xxx> <xxxx> short for <S-0-xxxx>	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Entry (Sercos ID) for the additive AT configuration list	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	-	
Sercos ID	S-0-1050.1.6	
Data type	CfgSCSP_ScsIdn_t (16 characters + string end)	

11.4.4 Signal status word

Signal status word configuration

Configuration signal status word		SigStatWConf
	Contains the user-specific settings of the additive signal status word configuration list (S-0-0026).	
Path	SCSP/Dr[i]/Teleg/SigStatW/Conf/SigStatWConf[1..16]	
Value range	<S-x-xxxx.xxx.xxx> (x = [0..9], max. 16 characters) <P-x-xxxx.xxx.xxx> <xxxx> short for <S-0-xxxx>	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Entry (Sercos ID) for the additive signal status word configuration list	

Sercos parameters (SCSP) - Basic parameters -

Access rights	MTB: Write
	Setter: Read
	Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	-
Sercos ID	S-0-0026
Data type	CfgSCSP_ScsIdn_t (16 characters + string end)

Bit assignments signal status word

Bit assignment signal status word		SigStatWAssign
Contains the user-specific settings of the additive bit assignments for the signal status word (S-0-0328).		
Path	SCSP/Dr[i]/Teleg/SigStatW/Assign/SigStatWAssign[1..16]	
Value range	0 .. 65535	
Pre-assignment	0	
Unit	-	
Description of parameter values	Bit position for the additive signal status word configuration	
Access rights	MTB: Write	
	Setter: Read	
	Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	-	
Sercos ID	S-0-0328	
Data type	CfgSCSP_U2Int_t (2 byte)	

11.4.5 Signal control word

Signal control word configuration

Configuration signal status word		SigCtrlWConf
Contains the user-specific settings of the additive signal control word configuration list (S-0-0027).		
Path	SCSP/Dr[i]/Teleg/SigCtrlW/Conf/SigCtrlWConf[1..16]	
Value range	<S-x-xxxx.xxx.xxx> (x = [0..9], max. 16 characters) <P-x-xxxx.xxx.xxx> <xxxx> short for <S-0-xxxx>	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Entry (Sercos ID) for the additive signal control word configuration list	
Access rights	MTB: Write	
	Setter: Read	
	Operator: Read	
Activation	System restart (CS)	

Dependency	-
Macoda ID	-
Sercos ID	S-0-0027
Data type	CfgSCSP_ScsIdn_t (16 characters + string end)

Bit assignment signal control word

	<table><tr><th>Bit assignment signal control word</th><th>SigCtrlWAssign</th></tr></table>	Bit assignment signal control word	SigCtrlWAssign
Bit assignment signal control word	SigCtrlWAssign		
	Contains the user-specific settings of the additive bit assignments for the signal control word (S-0-0329).		
Path	SCSP/Dr[i]/Teleg/SigCtrlW/Assign/SigCtrlWAssign[1..16]		
Value range	0 .. 65535		
Pre-assignment	0		
Unit	-		
Description of parameter values	Bit position for the additive signal control word configuration		
Access rights	MTB: Write Setter: Read Operator: Read		
Activation	System restart (CS)		
Dependency	-		
Macoda ID	-		
Sercos ID	S-0-0329		
Data type	CfgSCSP_U2Int_t (2 byte)		

12 NC/PLC interface (PLC) - Basic parameters

12.1 Overview

This chapter explains settings for the NC/PLC interface:

- [PLC selection, page 179](#)
- [Digital inputs/outputs, page 182](#)
- [Analog inputs/outputs, page 185](#)
- [NC function-specific bit interface, page 189](#)
- [Machine state display MZA, page 197](#)

12.2 PLC selection (Plc)

12.2.1 General information

*In the MTX, only **IndraLogic** is provided as PLC application for configuration purposes. The settings are located in the following subfolder:*

- **IL**

(see [chapter 12.2.2 IndraLogic, page 179](#))

If another subfolder than the **IL** subfolder is displayed, select the **PLC/Plc** folder and right-click on the on **IL** using the "Change to" function.

Macoda ID 2060 00200 (5 = IL; not relevant anymore: 1 = KnsTr, 2 = KnsStandard)

Furthermore, a channel-dependent resetting of CPL customer outputs to the PLC can be configured in case of a channel reset.

The settings are located in the following subfolder:

- **ResetCplIf**

(see [Chapter 12.2.3 IndraLogic, page 180](#))

12.2.2 IndraLogic

Path of the PLC application

Path of the PLC application	PlcSymbFilePath

Defines the directory path of the saved PLC application.

"/plc/" is assigned to this parameter by default.

To swap data to the CF card, enter the mount directory "/cf_oem/plc/".

Other directory paths are not permitted.

Path PLC/Plc/IL/PlcSymbFilePath

Value range {Text}

Pre-assignment /plc/

Unit -

Description of parameter values The directory path consists of single directory names.

The following applies:

- The path has to begin and end with a "/".
- A "/" has to be written as well between the directory names.
- The path may not contain any blanks.

Access rights **MTB:** Write

NC/PLC interface (PLC) - Basic parameters

	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00209	
Data type	Str64_t (isoLatin1String)	
	8 * 8 byte string for MACODA compatibility	

Minimum size of the PLC user program memory

Minimum size of the PLC user program memory		PlcMemIL
To compute the reserved memory size for the complete PLC.		
Path	PLC/Plc/IL/PlcMemIL	
Value range	0 .. 100000	
Pre-assignment	3000	
Unit	kB	
Description of parameter values	A useful value for PlcMemIL is twice the total size of code range and data range displayed in a message window when creating the PLC project in IndraWorks Engineering. If the memory space is insufficient, the PLC program is not downloaded and the PLC is not started. This parameter is not significant upon the control start-up in startup mode 3 (secure startup). In this case, the PLC never starts running.	
	 Changes are only applied after the control startup. The size of the user program memory can be limited due to licensing!	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00210	
Data type	Int0_100T_t (int)	
	4 byte integer, unsigned for MACODA compatibility	

12.2.3 Resetting the CPL customer outputs to the PLC

Start of range

Start of range		CplIfStartBit
Defines the start of the range of the CPL customer signals in the channel-related interface signals (iCh_Cpl01 to iCh_Cpl16) that are to be reset via channel reset.		
Path	PLC/Plc/ResetCplIf/Ch[k]/CplIfStartBit	
Value range	0	No CPL customer output reset
	1.. 16	CPL customer output 1 ... 16
Pre-assignment	0	No CPL customer output reset
Unit	-	

Description of parameter values	0	No CPL customer output range selected.
	1.. 16	Beginning of range "CPL customer output 1" (iCh_Cpl01) selected
	...	Beginning of range "CPL customer output 16" (iCh_Cpl16) selected



Also take the end of the range from the parameter "CplIfEndBit" in the folder PLC/Plc/ResetCplIf/Ch[k] into account.

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Activation System restart (CS)

Dependency -

Macoda ID 7070 00020

Data type int

4 byte integer for MACODA compatibility

End of range

End of range	CplIfEndBit

Defines the end of the range of the CPL customer signals in the channel-related interface signals (iCh_Cpl01 to iCh_Cpl16) to be reset via channel reset.

Path PLC/Plc/ResetCplIf/Ch[k]/CplIfEndBit

Value range	0	No CPL customer output reset
	1..16	CPL customer output 1 ... 16

Pre-assignment 0 No CPL customer output reset

Unit -

Description of parameter values	0	No CPL customer output range selected.
	1.. 16	End of range "CPL customer output 1" (iCh_Cpl01)
	...	End of range "CPL customer output 16" (iCh_Cpl16)



Also note the beginning of the range from the "CplIfStartBit" parameter in the PLC/Plc/ResetCplIf/Ch[k]/ folder.

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Activation System restart (CS)

Dependency -

Macoda ID 7070 00020

Data type int

4 byte integer for MACODA compatibility

12.3 Digital inputs/outputs (DigIO)

12.3.1 General information

Fast input/outputs are provided for the MTX:

- 8 digital inputs DigIOAllocIn[1..8]
(refer to the function assignment of digital inputs)
- 8 digital outputs DigIOAllocOut[1..8]
(refer to the function assignment of digital outputs)

12.3.2 Hardware selection

Hardware selection		DigIOHw
Specifies whether the digital high-speed or the local I/Os are linked via Profibus-DP. For rack-based controls, the local I/Os are assigned to the on-board I/Os and for PC-based controls, they are assigned to the I/Os of the additional plug-in card "MTX High-Speed I/O".		
Path	PLC/DigIO/DigIOHw	
Value range	0	High-speed I/Os via local I/Os
	1	High-speed I/Os via Profibus-DP
Pre-assignment	0	High-speed I/Os via local I/Os
Unit	-	
Description of parameter values	0	High-speed I/Os via local I/Os
	1	High-speed I/Os via Profibus-DP
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4075 00200	
Data type	Byte0_1_t	
	1 byte integer, unsigned for MACODA compatibility	

12.3.3 Synchronization type of the high-speed I/Os

Synchronization type of the high-speed I/Os		HSDigIOSynchrMode
Specifies whether digital high-speed I/Os are exchanged synchronously in the interpolator cycle or in the PLC cycle.		
Path	PLC/DigIO/HSDigIOSynchrMode	
Value range	0	Hardware-dependent Presetting
	1	Interpolator cycle
	2	PLC cycle
Pre-assignment	0	Hardware-dependent Presetting
Unit	-	
Description of parameter values	0	Hardware-dependent Presetting
	The preset synchronization type is used for the hardware. Digital high-speed I/Os are exchanged for all rack-based con-	

		trols in the PLC cycle and for all PC-based controls in the interpolator cycle.
	1	Interpolator cycle The digital high-speed I/Os are exchanged in the interpolator cycle.
	2	PLC cycle The digital high-speed I/Os are exchanged in the PLC cycle.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4075 00100	
Data type	Byte0_2_t (byte) 1 byte integer, unsigned for MACODA compatibility	

12.3.4 Initial address of digital inputs

Initial address of digital inputs		DigIOStartAddrIn
	Specifies the byte in the input range of the PLC on which the digital inputs are mapped.	
Path	PLC/DigIO/DigIOStartAddrIn	
Value range	{Text}	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	The digital inputs are not used by the PLC.
	E0 .. E8191	Input byte 0 to 8191 (IndraLogic, KNS)
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4075 00105	
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility	

12.3.5 Initial address of digital outputs

Initial address of digital outputs		DigIOStartAddrOut
	Specifies the byte in the output range of the PLC on which the digital outputs are mapped.	
Path	PLC/DigIO/DigIOStartAddrOut	
Value range	{Text}	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	The digital outputs are not used by the PLC.

NC/PLC interface (PLC) - Basic parameters

	A0 .. A8191	Output byte 0 to 8191 (IndraLogic, KNS)
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4075 00106	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

12.3.6 Function assignment of digital inputs

Assigning digital inputs[1..8]

Assign digital inputs[1..8]		DigIOAllocIn[d]
Assigns a digital input DigIOAllocIn[1..8] to a specific function unit.		
Path	PLC/DigIO/FuncAllocIn/DigIOAllocIn[1..8]	
Value range	0	Not used
	1	Punching
	2	PLC
	3	Block switching via HS signal
	<0	Reserved for OEM
Pre-assignment	0	Not used
Unit	-	
Description of parameter values	0	Not used
		Digital input is not used.
	1	Punching
		The digital input is assigned to the "Punch" function (also refer to "StrokeMonTime" in the PUN/ folder).
	2	PLC
		The digital input is assigned to the PLC (also refer to "DigIOStartAddrIn").
	3	Block switching via HS signal
		The digital input is assigned to the "HsBlkSwitch" (HSB) function (block switching via high-speed signal).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4075 00101	
Data type	IOFuncAlloc_t	
	4 byte integer, signed for MACODA compatibility	

12.3.7 Function assignment of digital outputs

Assigning digital outputs [1..8]

	Assign digital outputs [1..8]	DigIOAllocOut[d]
	Assigns a digital output DigIOAllocOut[1..8] to a specified function unit.	
Path	PLC/DigIO/FuncAllocOut/DigIOAllocOut[1..8]	
Value range	0	Not used
	1	Punching
	2	PLC
	6	Digitizing
	7	Fast actions (FA)
	501..560	"pos.-dep. HS output" for channel 1... 60
	<0	Reserved for OEM
Pre-assignment	0 (not used)	
Unit	-	
Description of parameter values	0	Not used
		Digital output is not used
	1	Punching
		The digital output is assigned to the "Punch" function (also refer to "StrokeMonTime" in the PUN/ folder)
	2	PLC
		The digital output is assigned to the PLC (also refer to "Dig-IOStartAddrOut")
	6	Digitizing
		The digital output is assigned to the "Digitizing" function
	7	Fast actions (FA)
		Digital output is assigned to the "Fast actions" function
	501..560	"pos.-dep. HS output" for channel 1... 60
		Channels 1-60 use this output for the "position-dependent HS output" (PosDepHSOut)
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4075 00102	
Data type	IOFuncAlloc_t	
	4 byte integer, signed for MACODA compatibility	

12.4 Analog inputs/outputs (AnalO)

12.4.1 General information

The MTX can control four analog outputs AnalOAllocOut[1..4] (refer to the function assignment "analog outputs"), but no analog inputs.



The four analog outputs with 16 bit data width each are always connected via Profibus DP.

Analog inputs are **not** available in the MTX.

12.4.2 Initial address of digital outputs

Initial address of analog outputs	AnalOStartAddrOut

Specifies the initial address of the MTX analog outputs in the output range of the PLC.



The analog outputs use four consecutive data words of 16 bits each in the output range of the PLC.

Path	PLC/AnalO/AnalOStartAddrOut	
Value range	{Text}	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	The analog outputs are not used by the PLC.
	A0 .. A8184	Output byte 0 to 8184
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4075 00108	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

12.4.3 Function assignment of analog outputs

Assigning analog inputs[1..4]

Assign analog outputs [1..4]	AnalOAllocOut[d]

Assigns an analog output AnalOAllocOut[1..4] to a specific function.

Path	PLC/AnalO/FuncAllocOut/AnalOAllocOut[1..4]	
Value range	0	
	2	
	401 .. 460	
	701 .. 760	
	< 0 (reserved for OEM)	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	The analog output is not used
	2	The analog output is assigned to the PLC (also refer to "AnalOStartAddrOut")
	401 .. 460	Spindle (1 .. 60) is an analog spindle

		(also see parameter "SpType", path: SP/SpFunc/Sp[1..60]/Base/SpType, Macoda_ID: 1040 00001)
	701 .. 760	Channel (1..60) uses this output for the "Laser power control" function
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4075 00104	
Data type	IOFuncAlloc_t	
	4 byte integer, signed for MACODA compatibility	

12.4.4 Offset specification

Offset of the D/A converter[1..4]

	Offset of the D/A converter[1..4]	DigAnaConvOffs[1..4]
	Compensates the offset of the D/A converter of the four analog outputs of the MTX.	
Path	PLC/AnaIO/ConvOffs/DigAnaConvOffs[1..4]	
Value range	-10.000 .. 10.000	
Pre-assignment	0.000	
Unit	V	
Description of parameter values	Offset compensation in Volts	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4075 00201	
Data type	Float-10_10_t (float)	
	4 byte float for MACODA compatibility	

12.4.5 Gain specification

Gain of the D/A converter[1..4]

Gain of D/A converter[1..4]	DigAnaConvGain[1..4]
Specifies, as in the following figure, the gain of the D/A converter for the four analog outputs of the MTX:	

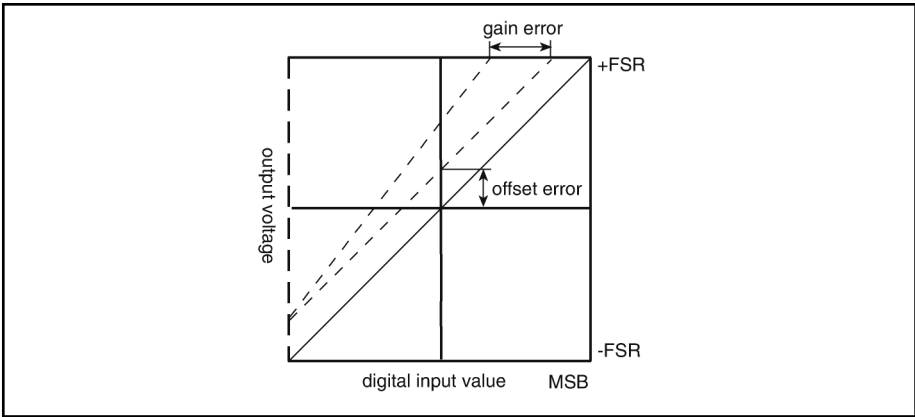


Fig. 12-1: Gain of D/A converter

Path	PLC/AnaIO/ConvOffs/DigAnaConvGain[1..4]		
Value range	-10.000 .. 10.000		
Pre-assignment	0.000		
Unit	V		
Description of parameter values	Gain in volts		
Access rights	MTB:	Write	
	Setter:	Read	
	Operator:	Read	
Activation	System restart (CS)		
Dependency	-		
Macoda ID	4075 00202		
Data type	Float-10_10_t (float)		
	4 byte float for MACODA compatibility		

12.4.6Type selection

D/A converter[1..4] type

D/A converter[1..4] type		DigAnaConvType[1..4]
Specifies the D/A converter type for the four analog outputs of the MTX.		
Path	PLC/AnaIO/ConvOffs/DigAnaConvType[1..4]	
Value range	0	16 bits, -10 to 10 V, dual complement
	1	12 bits, -10 to 10 V, dual complement, left-justified
	2	12 bits, -10 to 10 V, dual complement, right-justified
	10	16 bits, 0..10 V
	11	12 bits, 0..10 V, left-justified
	12	12 bits, 0..10 V, right-justified
Pre-assignment	0	16 bits, -10 to 10 V, dual complement
Unit	V	
Description of parameter values	D/A converter type (see "Value range")	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Activation	System restart (CS)
Dependency	-
Macoda ID	4075 00203
Data type	int
	1 byte integer, unsigned for MACODA compatibility

12.5 NC function-specific bit interface (NcFuncBitIf)

12.5.1 General information

Several applications in the PLC program have to be informed on the activity of individual NC functions. When the NC function is enabled for the PLC, the NC has to set a corresponding status signal and has to disable it at deletion. Since not all of the NC functions can be applied to the bit interface, up to 24 NC functions can be mapped at the bit interface (refer to the table [Modal NC functions for the bit interface, page 191](#)).

These signals exist in the PLC channel input interface with the names "iCh_Act- Func01..iCh_ActFunc24". Parameters assign at control startup. This assignment can only be changed upon the next startup.

12.5.2 NC function for the bit interface iCh_ActFunc[1..24]

NC function for the bit interface iCh_ActFunc[1..24]		NcFunc[1..24]
Describes the NC function at the bit interface for the bit iCh_ActFunc[1..24].		
Path	PLC/NcFuncBitIf/NcFunc[1..24]	
Value range	{NC syntax}, max. 16 characters	
Pre-assignment	G70	PLC/NcFuncBitIf/NcFunc[1]
	OVD	PLC/NcFuncBitIf/NcFunc[2]
	G95	PLC/NcFuncBitIf/NcFunc[3]
	NcfNoAxTrafo2	PLC/NcFuncBitIf/NcFunc[4]
	SETPOS	PLC/NcFuncBitIf/NcFunc[5]
	TST	PLC/NcFuncBitIf/NcFunc[6]
	G63	PLC/NcFuncBitIf/NcFunc[7]
	G33	PLC/NcFuncBitIf/NcFunc[8]
	ED	PLC/NcFuncBitIf/NcFunc[9]
	ED	PLC/NcFuncBitIf/NcFunc[10]
	ED	PLC/NcFuncBitIf/NcFunc[11]
	ED	PLC/NcFuncBitIf/NcFunc[12]
	ED	PLC/NcFuncBitIf/NcFunc[13]
	-	PLC/NcFuncBitIf/NcFunc[14..17]
	G96	PLC/NcFuncBitIf/NcFunc[18]
	-	PLC/NcFuncBitIf/NcFunc[19..24]
Unit	-	

NC/PLC interface (PLC) - Basic parameters

Description of parameter values

G70, OVD, G95, NcfNoAxTrafo, SETPOS, TST, G63, G33, G96

"Single-bit signals" for the NC function state "Active"/"Inactive":

The single parameter <n> assigns an NC function each to the iCh_ActFunc signal.

Example: PLC/NcFuncBitIf/NcFunc[1] = G70

Part program:

N10 G70 iCh_ActFunc01 = TRUE

N20 G1 X10 Y20 iCh_ActFunc01 = TRUE

N30 G71 iCh_ActFunc01 = FALSE

ED **"Multi-bit values"** that map the programmed value of the function contain the same NC function (only for "D" and "ED") if there are directly consecutive single parameters. Thus, the programmed value of the NC function is assigned to these bits as binary value.

The value range results from the number of assigned bits n: $0..2^n-1$.

The control limits higher values to the value modulo 2^n .

Example: PLC/NcFuncBitIf/NcFunc[9..12] = ED

Part program:

N10 ED10 The binary value 1010 is output on the signals
iCh_ActFunc09..12 (iCh_ActFunc10 = TRUE and
iCh_ActFunc12 = TRUE)

N20 ED16 The binary value 10000 deletes with $16 \bmod 24 = 0$
iCh_ActFunc09..12

N30 ED15 The binary value 1111 sets all signals
iCh_ActFunc09..12 = TRUE

N40 ED0 All signals iCh_ActFunc09..12 are deleted.

Access rights **MTB:** Write
 Setter: Read
 Operator: Read

Activation System restart (CS)

Dependency -

Macoda ID 3020 00001 (characters 1..8)
 3020 00002 (characters 9..16)

Data type Str16_t (isoLatin1String)
 8 byte string for MACODA compatibility

The following tables show the identifier to be parameterized to display the corresponding NC function at the bit interface. Synonym identifiers are listed in one line separated by commas.

It is differentiated between:

- local NC functions and
- modal NC functions.

The table [Local NC functions for the bit interface, page 191](#) is complete, that means only these local NC functions can be output at the bit interface.

The table [Modal NC functions for the bit interface, page 191](#) can be extended, that means that the user can output each added modal NC function at

the bit interface. The user can parameterize the bit by using the syntax of the modal NC function as identifier.

Furthermore, there are identifiers that deviate from the regular NC syntax as in the coordinate transformations.

Local NC function	Identifier
Tapping	G63 (The signal parameterized for G63 is also set with the retract programming "TappRet2")
Set program position	PLC, SetPos, SETPOS
G33 auxiliary function: TST for TST(TCI1) / TST(TCI0)	TST, ThreadSet
Travel against first touch probe	G75.1, G75
Travel against second touch probe	G75.2
Flying Measurement	FME, FlyMeas

Tab. 12-1: Local NC function for the bit interface

Modal NC functions (groups)	Identifier
Modal drilling cycles	ModSubProgOff
	ModSubProgNr1
	ModSubProgNr2
	ModSubProgNr3
	ModSubProgNr4
	ModSubProgNr5
	ModSubProgNr6
	ModSubProgNr7
	ModSubProgNr8
	ModSubProgNr9
	ModSubProgNr10
	ModSubProgNr11
	ModSubProgNr12
	ModSubProgNr13
	ModSubProgNr14
	ModSubProgNr15
Interpolation	G00, G0
	G01, G1
	G2, G02
	G3, G03
	G5, G05
	G06, G6
	G33
	HandwheelWCS

NC/PLC interface (PLC) - Basic parameters

Modal NC functions (groups)	Identifier
Path slope	G9, G09
	CFD, ConstFeed
	LNU, LinUpFeed
	LND, LinDownFeed
	SNU, SinUpFeed
	SND, SinDownFeed
	S2U, Sin2UpFeed
	S2D, Sin2DownFeed
	G8, G08
Absolute/relative measure	G90
	G91
Plane selection	G17
	G18
	G19
	G20
	G16
Feed programming	G94
	G93
	G95
Feed 100%	OVD, OvrDis
	OVE, OvrEna
Inch/metric	G71
	G70
CO programming	KVP, KvProg
Feed forward	FFW, FeedForward
Axis acceleration	AAC, AxAcc
Exact stop	G61
	G62
Path acceleration	PAC, PathAcc
Tool correction assignment	G79
	G78
Tool correction activation	G47
	G48

Modal NC functions (groups)	Identifier
Zero point offset bank 1	G53.1
	G52, G52.1
	G54, G54.1
	G55, G55.1
	G56, G56.1
	G57, G57.1
	G58, G58.1
	G59, G59.1
Zero point offset bank 2	G53.2
	G52.2
	G54.2
	G55.2
	G56.2
	G57.2
	G58.2
	G59.2
Zero point offset bank 3	G53.3
	G52.3
	G54.3
	G55.3
	G56.3
	G57.3
	G58.3
	G59.3
Contour offset	SHT, Shift
Mirror coordinates	MIR, Mirror
Scale coordinates	SCL, Scale
Rotate coordinates	ROT, Rotate
Polar coordinates	PLS, PoleSet
Setting-up correction (workpiece position correction 0)	BCR, BcsCorr

NC/PLC interface (PLC) - Basic parameters

Modal NC functions (groups)	Identifier
Inclined plane of bank 1 (workpiece position correction 1)	G153.1
	G151, G151.1
	G152, G152.1
	G154, G154.1
	G155, G155.1
	G156, G156.1
	G157, G157.1
	G158, G158.1
	G159, G159.1
Positioning mode (rotary axis)	PMD, PosMode
Transition angle	DTT, DefTangTrans
Minimum spindle speed	SMN, SMin
Maximum spindle speed	SMX, SMax, SMAX
Spindle programming	G97
	G96
Zero point offset bank 4	G53.4
	G52.4
	G54.4
	G55.4
	G56.4
	G57.4
	G58.4
	G59.4
Zero point offset bank 5	G53.5
	G52.5
	G54.5
	G55.5
	G56.5
	G57.5
	G58.5
	G59.5

Modal NC functions (groups)	Identifier
Inclined plane of bank 2 (workpiece position correction 2)	G153.2
	G151.2
	G152.2
	G154.2
	G155.2
	G156.2
	G157.2
	G158.2
	G159.2
Inclined plane of bank 3 (workpiece position correction 3)	G153.3
	G151.3
	G152.3
	G154.3
	G155.3
	G156.3
	G157.3
	G158.3
	G159.3
Inclined plane of bank 4 (workpiece position correction 4)	G153.4
	G151.4
	G152.4
	G154.4
	G155.4
	G156.4
	G157.4
	G158.4
	G159.4
Inclined plane of bank 5 (workpiece position correction 5)	G153.5
	G151.5
	G152.5
	G154.5
	G155.5
	G156.5
	G157.5
	G158.5
	G159.5

NC/PLC interface (PLC) - Basic parameters

Modal NC functions (groups)	Identifier
Chamfer/rounding	RND, Rounding
	RNE, RoundEps
	CHL, ChLength
	CHS, ChSection
D-correction	D0
	D
Tool edge selection	ED
	ED0
Feed computation	FAD, FeedAd
Absolute program coordinate offset	TRS, Trans
Additive program coordinate offset	ATR, ATrans
Accuracy programming	PRP, PrecProg
Axis coupling	AXC, AxCouple
Tool orientation	TTL, TangTool
Punching/nibbling	PUN, Punch
	NIB, Nibble
Tang. Tool orientation	TTO, TangToolOri
Range monitoring enabled	ONA
	OFA
Radius/diameter programming	RAD
	DIA
AND stop conditions enabled	ONSTPA, ONSTOPA
	OFSTPA, OFFSTOPA
OR stop condition enabled	ONSTPO, ONSTOPO
	OFSTPO, OFFSTOPO
Laser power control	LFP
Coordinate coupling	SCC, SelCrdCouple
3D radius correction	G140
	G141
	G142
Travel limit enabled	TRL, TravLim
Programmable path split	Split, SPLIT
Coordinate conversion	PLP
	EPC, EndPosCouple
Geometry compensation	GCT, GeoComp
Radius correction of the milling cutter	G40
	G41
	G42

Modal NC functions (groups)	Identifier
Insertion strategy (transition circle/intersection)	G43
	G44
Contact velocity of the milling cutter	G45
	G46
Collision monitoring	CLN, CollisionCheck

Tab. 12-2: Modal NC function for the bit interface

12.6 Machine state display MZA (MachStatDiag)

12.6.1 Cycle time

Cycle time	MachStatDiagCycTime

The MZA cycle time covers

- the frequency at which MZA files are requested and evaluated and
- the updating frequency in the MZA dialog.

With MZA, machine errors, warnings or notes can be generated.

Path	PLC/MachStatDiag/MachStatDiagCycTime	
Value range	0 .. 100000 0 .. 999 for MACODA compatibility	
Pre-assignment	0	
Unit	ms	
Description of parameter values	0	MZA is not used.
	>0	Interval after which MZA data is updated.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00090	
Data type	Int0_100T_t (int) 2 byte integer, unsigned for MACODA compatibility.	

13 Auxiliary functions (AUXF) – Basic parameters -

13.1 Auxiliary function basics

The auxiliary functions

- are components of the NC syntax that can be programmed both in a part program and in "Setup" mode;
- are used to exchange information with the PLC via markers, e.g. to switch machine functions on or off, to switch from one machine function to another, or to transmit or receive numeric data;
- may or may not require acknowledgement. For auxiliary functions requiring acknowledgement, the control waits until a positive acknowledgement is received (e.g. by flags being reset by the PLC) before processing the next program block or changing to "READY" state;
- can be assigned to an auxiliary function group upon mutual exclusion of their activity or display.

Auxiliary function classes

The MTX provides various different auxiliary function classes. What class to be assigned to an auxiliary function depends on the kind of information to be exchanged and the desired range of action (channel-dependent or channel-independent).

The MTX supports the following auxiliary function classes:

- Bit-coded, channel-dependent
- 32-bit BCD-coded, channel dependent
- 32-bit BCD-coded, channel independent
- Integer-coded, channel-dependent
- Integer-coded, channel-independent
- Float-coded, channel-dependent
- Float-coded, channel-independent
- Double-coded, channel-dependent
- Double-coded, channel-independent

Auxiliary function classes also always include the auxiliary function type and range of action.

Channel-dependent and channel-independent auxiliary functions

The differences between the two ranges of action are:

- Channel-independent auxiliary functions exist only once within the system. Therefore, their current states are identical on each applied channel.
- Channel-dependent auxiliary functions exist once per applied channel. Thus, the state of such auxiliary functions can differ in all applied channels.

Bit-coded auxiliary functions:

This type of auxiliary function can be used to exchange individual binary information with the PLC (e.g., to switch on or off machine functions or to switch from one machine function to another). Bit-coded auxiliary functions are assigned to a **bit array** with a "start bit address" and an "end bit address". The individual bits of any such bit array are programmed, using NC syntax, with the respective name of the auxiliary function (letters only!) directly followed by an integer. The value range of the integer is specified by the "start code" and the "end code".



In the MTX, bit-coded auxiliary functions are first configured as channel-dependent auxiliary functions as a rule.

Auxiliary functions (AUXF) – Basic parameters -

Non-bit-coded Auxiliary Functions

Non-bit-coded auxiliary functions are used if numerical information should be exchanged with the PLC, e.g. to transfer additional values to functions. The name of this type of auxiliary function can contain letters, digits and the character "_". The name is followed directly by the data value.

Example:

Auxiliary function name "T" and data value "12345"

The auxiliary function is programmed as "T12345".

32-bit-BCD-coded auxiliary functions

This type of auxiliary function is used when numerical information must be exchanged with the PLC in BCD format (Binary Coded Decimal).

A 32-bit BCD-coded auxiliary function must have an 8-byte marker area (= 64 bits) and can transmit up to 8-digit data values.

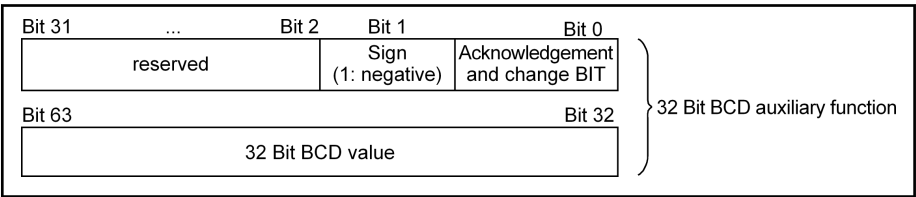


Fig. 13-1: Marker area of a 32-bit BCD auxiliary function

Integer-coded auxiliary functions

This type of auxiliary function is used when numerical information has to be exchanged with the PLC in integer format.

An integer auxiliary function has to have an 8-byte marker area (= 64 bits) and can transmit numerical values between -2147483648 and +2147483647.

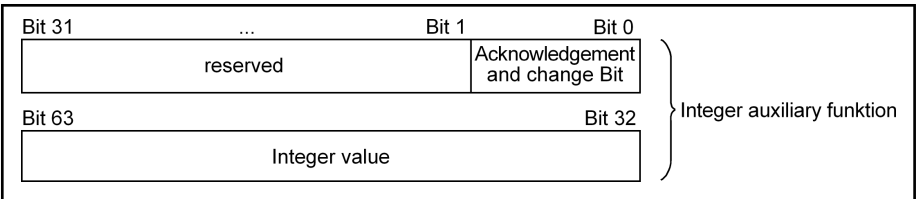


Fig. 13-2: Marker area of an integer auxiliary function

Float-coded auxiliary functions

This type of auxiliary function is used when numerical information has to be exchanged with the PLC in float format.

A float auxiliary function has to have an 8-byte marker area (= 64 bits) and can transmit floating point values between 1.175494351E-38 and 3.402823466E+38 and between -1.175494351E-38 and -3.402823466E+38 with an accuracy of six decimal places.

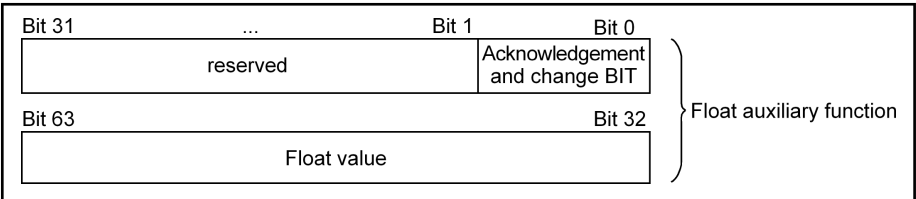


Fig. 13-3: Marker area of a float auxiliary function

Double-coded auxiliary functions

This type of auxiliary function is used when numerical information has to be exchanged with the PLC in double format.

A double auxiliary function has to have an 12-byte marker area (= 96 bits) and can transmit floating point values between 2.2250738585072014E-308 and -1.7976931348623158E+308 and between -2.2250738585072014E-308 and 1.7976931348623158E+308 with an accuracy of 15 decimal places.

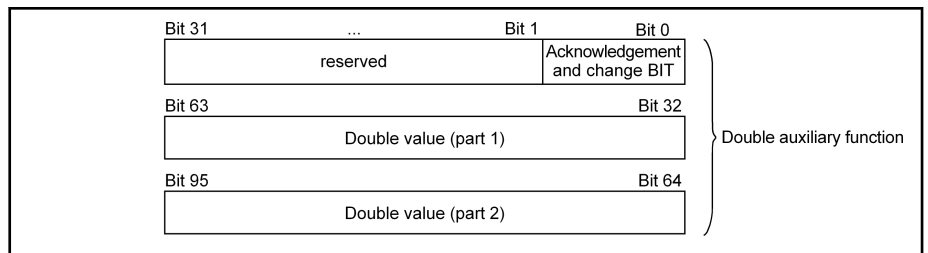


Fig. 13-4: Marker area of a double auxiliary function

Auxiliary function groups

Auxiliary functions that mutually exclude one another in the display and/or their activity can be grouped into auxiliary function groups. This is done with syntax groups in the machine parameters.

The control is able to determine the last enabled auxiliary function of an auxiliary function group. This way, the last M-code of the group "M3, M4, M5, M19" or the last T word, for example, can be displayed.

Auxiliary function groups are defined as either channel-dependent or channel-independent. Auxiliary functions of the same name may only be assigned to a group once. Channel-independent groups may only contain channel-independent auxiliary functions, and channel-dependent groups may only contain channel-dependent auxiliary functions.

13.2 Overview

The following settings are available in the "AUXF" data group to define auxiliary functions :

- [Auxiliary function class definition -ClassDef-](#) , page 201
- [Acknowledgement behavior for bit-coded auxiliary functions -BitAckn-](#) , page 213
- [Group definition -GrDef-](#) , page 216
- [Global output behavior -OutBehav-](#) , page 217
- [Number of auxiliary functions -NofAuxFunc-](#) , page 218
- [Addresses of auxiliary functions in the PLC interface -Plclf-](#) , page 223

Application examples are explained in the following:

- Definition of a new auxiliary function
(chapter 13.9 "Defining a new auxiliary function" on page 228) und
- parameterization of auxiliary functions
(chapter 13.10 "Parameterizing example: Auxiliary functions" on page 231)

13.3 Auxiliary function class definition (ClassDef)

13.3.1 General information

Up to 30 auxiliary functions can be defined. Here, the auxiliary function type, name and action range and parameters specific to each auxiliary function are defined.

13.3.2 Auxiliary function types

When defining auxiliary function classes, the "auxiliary function type" has to be selected first.

To do so, select the AUXF/ClassDef/Class[1..30]/ folder and, with the right mouse button, select the appropriate entry with the function "Convert to..." from the context menu :

- Bit
- Bcd32
- Int
- Float
- Double

Macoda ID3010 00040

(1 = Bit, 2 = Bcd32 channel-independent, 4 = Bcd32 channel-dependent, 6 = Int channel-independent, 7 = Float channel-independent, 8 = Double channel-independent, 9 = Int channel-dependent, 10 = Float channel-dependent, 11 = Double channel-dependent)



The entry for the auxiliary function type currently enabled for Class[1-30] is not shown in the context menu.

The auxiliary functions are described below:

Settings such as "Enable auxiliary function", "Name" and "Channel dependency", which apply equally to all auxiliary functions, are described in the General section (General parameters of auxiliary function classes Class [1..30]).

Following this are the special parameters for each auxiliary function type (bit, bcd32, int, float and double-coded auxiliary functions of auxiliary function classes [1-30]).

13.3.3 General parameters of auxiliary function classes class [1-30]

Enabling auxiliary function

Enable auxiliary function		EnablAuxFunc
<i>Enables/disables the auxiliary function from class [1..30] named "AuxFunc-Name", relating to auxiliary function type:</i>		
• Bit auxiliary function • BCD32 auxiliary Function • Integer auxiliary function • Float auxiliary function • Double auxiliary function		
Path	AUXF/ClassDef/Class[1-30]/EnablAuxFunc	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No Auxiliary function with name in .../Class[1..30]/AuxFuncName not enabled.
	1	Yes Auxiliary function with name in .../Class[1..30]/AuxFuncName enabled.

Access rights	MTB: Write
	Setter: Read
	Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	3010 00040
Data type	Byte0_1_t (byte)
	4 byte integer, unsigned for MACODA compatibility

Name

Name	AuxFuncName

Specifies an auxiliary function "Name" for each class [1..30] in one of the auxiliary function types named below:

- Bit auxiliary function
- BCD32 auxiliary Function
- Integer auxiliary function
- Float auxiliary function
- Double auxiliary function



Names of channel-dependent bit-coded auxiliary functions can also be used more than once (see example below).

Please note the following when assigning names to auxiliary functions:

For "channel-dependent" bit-coded auxiliary functions (see "AuxFuncCh-Dep"), only **letters** are allowed in the name.

Path	AUXF/ClassDef/Class[1..30]/AuxFuncName	
Value range	{Text}	
Pre-assignment	M	.../Class[1]/AuxFuncName
	S?	.../Class[2]/AuxFuncName
	T	.../Class[1]/AuxFuncName
	-	.../Class[4..30]/AuxFuncName
Unit	-	

Description of parameter values Name of auxiliary function (dependent on auxiliary function type)

Access rights	MTB: Write
	Setter: Read
	Operator: Read

Activation	System restart (CS)
Dependency	-

Macoda ID 3010 00030

Data type Str8_t (isoLatin1String)
8 byte string for MACODA compatibility

Example:

Auxiliary functions

You need access to two bit memory areas of 20 bits each. It should be possible to program both of these bit memory areas under the name TEST

Auxiliary functions (AUXF) – Basic parameters -

(TEST1 through TEST20 and TEST1001 through TEST1020). Gaps in the programming code are possible only if the name TEST may be used repeatedly. Otherwise, more storage space would have to be provided in the PLC.

Therefore, parameterize two auxiliary functions, both with the name TEST, that have different starting and end bit addresses as well as a different programming codes:

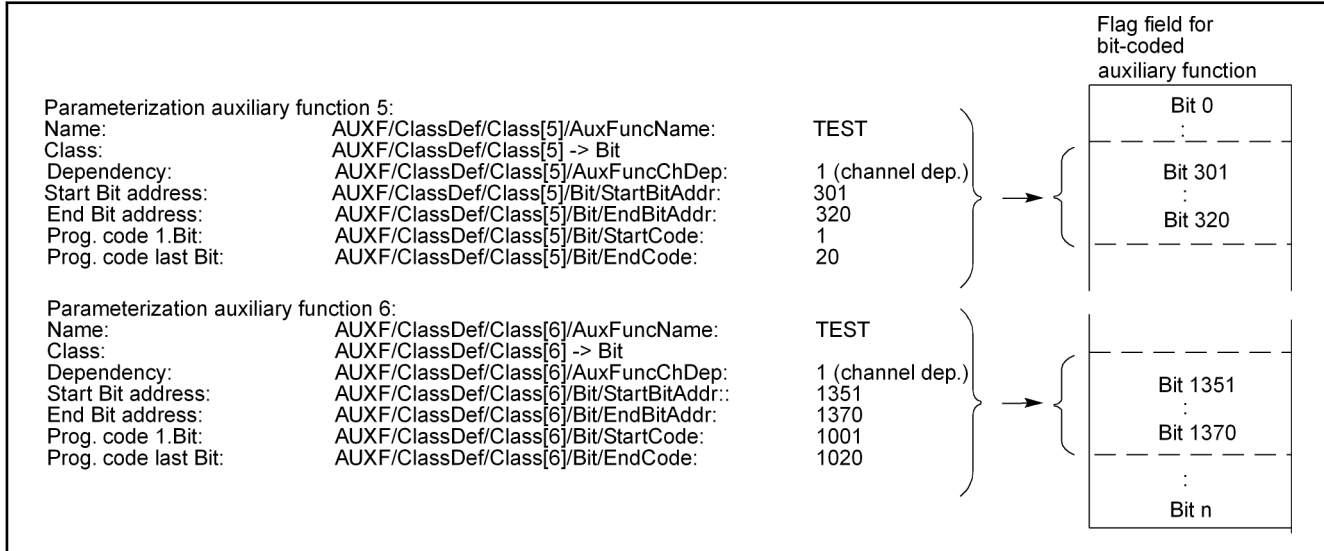


Fig. 13-5: Example 1 with two homonymous auxiliary functions

Example:

Auxiliary Function Program Code Areas

For matters of compatibility, a bit memory area should be accessible through 2 different programming code areas. According to specifications, the name of the auxiliary function may not be changed for this. This problem can be solved only because the name TEST may be assigned more than once.

Set the parameters of 2 auxiliary functions, both called TEST, that differ only in respect to their programming code areas:

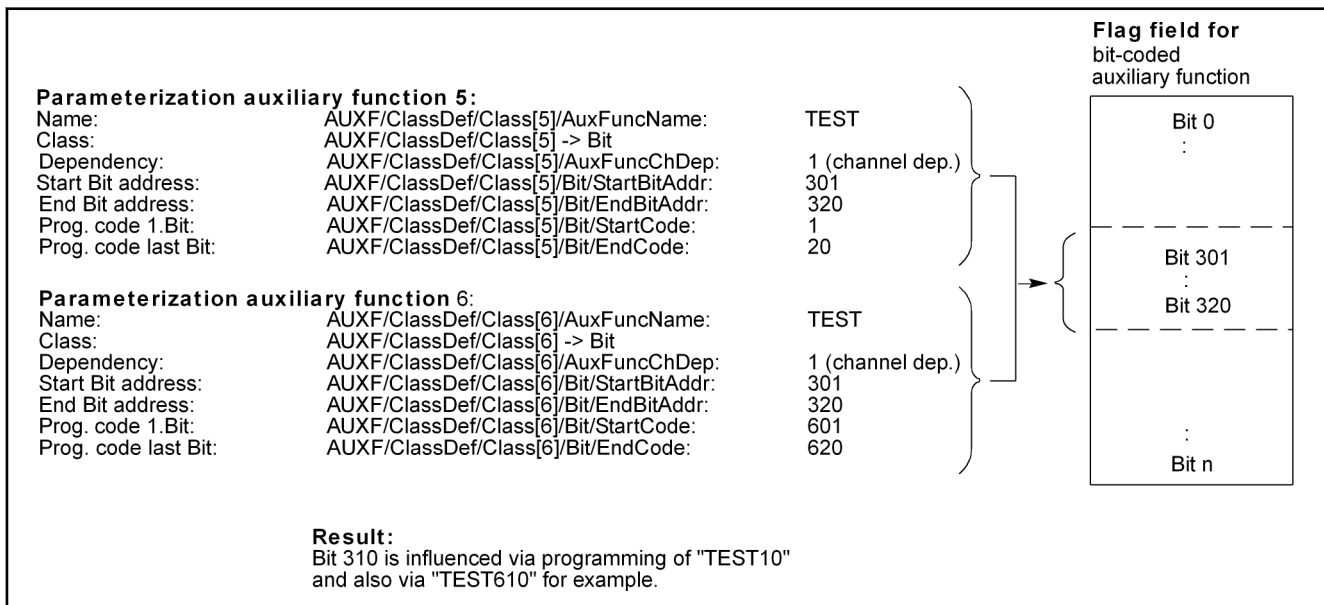


Fig. 13-6: Example 2 with two homonymous auxiliary functions

Channel dependency

Channel dependency		AuxFuncChDep
<i>Specifies whether the auxiliary function named in AUXF/ClassDef/Class[1..30]/AuxFuncName is "channel-dependent", relating to auxiliary function type:</i>		
• BCD32 auxiliary Function • Integer auxiliary function • Float auxiliary function • Double auxiliary function		
Path	AUXF/ClassDef/Class[1..30]/AuxFuncChDep	
Value range	0 (channel-independent auxiliary function) 1 (channel-dependent auxiliary function)	
Pre-assignment	0 (channel-independent auxiliary function)	
Unit	-	
Description of parameter values	0 Auxiliary function from the corresponding auxiliary function type is channel-independent 1 Auxiliary function from the corresponding auxiliary function type is channel-dependent	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00040	
Data type	Byte0_1_t (byte) 4 byte integer, unsigned for MACODA compatibility	

13.3.4 Bit-coded auxiliary functions of auxiliary function classes class [1..30]

Start code

Start code		StartCode
Defines which digit the first bit (start code) can be addressed with for a bit-coded auxiliary function defined under .../Class[1..30]/Bit/ that is assigned to a whole bit array.		
Start and end codes specify how the bits of a bit-coded auxiliary function can be addressed using NC syntax (see also second example of the parameter AuxFuncChDep).		
Path	AUXF/ClassDef/Class[1..30]/Bit/StartCode	
Value range	0 .. 65535	
Pre-assignment	0	
Unit	-	
Description of parameter values	Start code (first bit)	
Access rights	MTB: Write Setter: Read Operator: Read	

Auxiliary functions (AUXF) – Basic parameters -

Activation	System reset (SR)
Dependency	-
Macoda ID	3010 00080
Data type	Int0_65535_t (int)
	2 byte integer, unsigned for MACODA compatibility

End code

End code		EndCode
Defines which digit the last bit (end code) can be addressed with for a bit-coded auxiliary function defined under .../Class[1..30]/Bit/ that is assigned to a whole bit array.		
Start and end codes specify how the bits of a bit-coded auxiliary function can be addressed using NC syntax (see also second example of the parameter AuxFuncChDep).		
Path	AUXF/ClassDef/Class[1..30]/Bit/EndCode	
Value range	0 .. 65535	
Pre-assignment	299	.../Class[1]/Bit/EndCode
	0	.../Class[2-30]/Bit/EndCode
Unit	-	
Description of parameter values	End code (last bit)	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00090	
Data type	Int0_65535_t (int)	
	2 byte integer, unsigned for MACODA compatibility	

Start bit address

Start bit address		StartBitAddr
<i>Defines for an under .../Class[1..30]/Bit/ defined</i>		
• Bit-coded auxiliary function assigned to an entire bit array is the address of the first bit. • Bit-coded auxiliary function assigned to a single bit, that is the address of this bit.		
Path	AUXF/ClassDef/Class[1..30]/Bit/StartBitAddr	
Value range	0 .. 65535	
Pre-assignment	0	.../Class[1..30]/Bit/StartBitAddr
Unit	-	
Description of parameter values	Address of the first bit (bit array) or address of the bit (single bit)	
 All addresses relating to the starting address of the bit auxiliary function marker area are offset values (see also folder AUXF/NofAuxFunc/UsabAuxFunc/..., single parameters).		

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00060	
Data type	Int0_65535_t (int)	
	2 byte integer, unsigned for MACODA compatibility	

End bit address

End bit address		EndBitAddr
Defines the address of the last bit (end code) for a bit-coded auxiliary function defined under .../Class[1..30]/Bit/		
Path	AUXF/ClassDef/Class[1..30]/Bit/EndBitAddr	
Value range	0 .. 65535	
Pre-assignment	299	.../Class[1]/Bit/EndBitAddr
	0	.../Class[2..60]/Bit/EndBitAddr
Unit	-	
Description of parameter values	Address of the last bit (bit array)	
 All addresses relating to the end address of the bit auxiliary function marker area are offset values (see also folder AUXF/NofAuxFunc/UsabAuxFunc/..., single parameters).		
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00070	
Data type	Int0_65535_t (int)	
	2 byte integer, unsigned for MACODA compatibility	

13.3.5 BCD32-coded auxiliary functions of auxiliary function classes class [1..30]

Acknowledgement requirement

Acknowledgement requirement		AcknBcd32
Specifies whether a BCD32-coded auxiliary function defined under Class[1..30]/Bcd32 requires acknowledgement.		
Path	AUXF/ClassDef/Class[1..30]/Bcd32/AcknBcd32	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No

Auxiliary functions (AUXF) – Basic parameters -

		Auxiliary function does not require acknowledgement
	1	Yes
		Auxiliary function requires acknowledgement
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00090	
Data type	Byte0_1_t (byte)	
	2 byte integer, unsigned for MACODA compatibility	

Integer places

Integer places		PrelPlaceBcd32
Defines the number of integer places for a Bcd32-coded auxiliary function defined under Class[1..30]/Bcd32.		
The total number of integer places and decimal places cannot exceed 8 (see also parameter in folder AUXF/ClassDef/Class[1..30]/Bcd32/).		
Path	AUXF/ClassDef/Class[1..30]/Bcd32/PrelPlaceBcd32	
Value range	0 .. 8	
Pre-assignment	8	../Class[1..30]/Bcd32/PrelPlaceBcd32
Unit	-	
Description of parameter values	Number of places to the left of the decimal point	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00060	
Data type	Byte0_8_t (byte)	
	2 byte integer, unsigned for MACODA compatibility	
	Example:	

Decimal places and integer places of a BCD32-coded auxiliary function

Having 4 places to the left and 4 places to the right of the decimal point, the auxiliary function **T1234.56** would be stored in the PLC memory as follows:

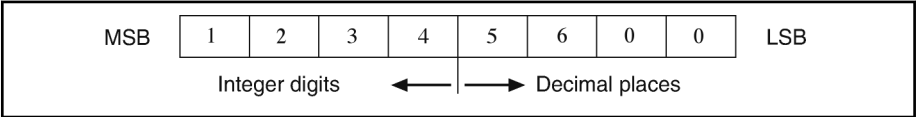


Fig. 13-7: BCD32-coded auxiliary function

Decimal places

Decimal places	DeciPlaceBcd32
Defines the number of places to the left of the decimal point for a Bcd32-coded auxiliary function defined under ../Class[1..30]/Bcd32.	

The total number of places to the left and to the right of the decimal point cannot exceed 8 (see also parameter in folder AUXF/ClassDef/Class[1..30]/Bcd32/).

Path	AUXF/ClassDef/Class[1..30]/Bcd32/DeciPlaceBcd32	
Value range	0 .. 8	
Pre-assignment	0	.../Class[1..30]/Bcd32/PrelPlaceBcd32
Unit	-	
Description of parameter values	Number of decimal places	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00070	
Data type	Byte0_8_t (byte)	
	2 byte integer, unsigned for MACODA compatibility	

Example:

Decimal places and integer places of a BCD32-coded auxiliary function

Having 4 places to the left and 4 places to the right of the decimal point, the auxiliary function **T1234.56** would be stored in the PLC memory as follows:

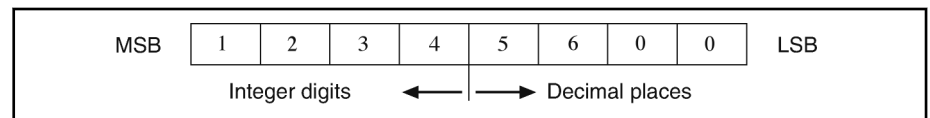


Fig. 13-8: Bcd32-coded auxiliary function

Index

index	IndBcd32

Specifies the "index" of a BCD32-coded auxiliary function defined under .../Class[1..30]/Bcd32.

Their index determines the order in which several auxiliary functions belonging to a specific class of auxiliary functions are stored in the marker filed of class Class[1..30]/Bcd32.

Path	AUXF/ClassDef/Class[1..30]/Bcd32/IndBcd32	
Value range	0 .. 32	
Pre-assignment	0	.../Class[1..30]/Bcd32/IndBcd32
Unit	-	
Description of parameter values	Index of the BCD32-coded auxiliary function.	
	The highest index is dependent on the number of existing BCD32-coded auxiliary functions in the class Class[1..30]/Bcd32 (parameter "Usable auxiliary functions" UsabAuxFunc in the folder AUXF/NofAuxFunc/).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	

Auxiliary functions (AUXF) – Basic parameters -

Dependency	-
Macoda ID	3010 00080
Data type	Byte0_32_t (byte)
	2 byte integer, unsigned for MACODA compatibility

13.3.6 Integer-coded auxiliary functions of auxiliary function classes [1..30]

Acknowledgement requirement

	Acknowledgement requirement	AcknInt
	Specifies whether an integer-coded auxiliary function defined under Class[1..30]/Int requires acknowledgement.	
Path	AUXF/ClassDef/Class[1..30]/Int/AcknInt	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
	Auxiliary function does not require acknowledgement	
	1	Yes
	Auxiliary function requires acknowledgement	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00090	
Data type	Byte0_1_t (byte)	
	2 byte integer, unsigned for MACODA compatibility	

Index

	index	IndInt
	Specifies the "index" of an integer-coded auxiliary function defined under ../Class[1..30]/Int.	
	Their index determines the order in which several auxiliary functions belonging to a specific class of auxiliary functions are stored in the marker filed of class Class[1..30]/Int.	
Path	AUXF/ClassDef/Class[1..30]/Int/IndInt	
Value range	0 .. 32	
Pre-assignment	0	../Class[1..30]/Int/IndInt
Unit	-	
Description of parameter values	Index of the integer-coded auxiliary function.	
	The highest index is dependent on the number of existing integer-coded auxiliary functions in the class Class[1..30]/Int (parameter "UsabAuxFunc" in the folder AUXF/NofAuxFunc/).	
Access rights	MTB:	Write

Setter:	Read
Operator:	Read
Activation	System reset (SR)
Dependency	-
Macoda ID	3010 00080
Data type	Byte0_32_t (byte)
	2 byte integer, unsigned for MACODA compatibility

13.3.7 Float-coded auxiliary functions of auxiliary function classes class [1..30]

Acknowledgement requirement

Acknowledgement requirement		AcknFloat
	Specifies whether a float-coded auxiliary function defined under Class[1..30]/Float requires acknowledgement.	
Path	AUXF/ClassDef/Class[1..30]/Float/AcknFloat	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Auxiliary function does not require acknowledgement
	1	Yes
		Auxiliary function requires acknowledgement
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00090	
Data type	Byte0_1_t (byte)	
	2 byte integer, unsigned for MACODA compatibility	

Index

index		IndFloat
	Specifies the "index" of a float-coded auxiliary function defined under ../Class[1..30]/Float.	
	Their index determines the order in which several auxiliary functions belonging to a specific class of auxiliary functions are stored in the marker filed of class Class[1..30]/Float.	
Path	AUXF/ClassDef/Class[1..30]/Float/IndFloat	
Value range	0 .. 32	
Pre-assignment	0	../Class[1..30]/Float/IndFloat
Unit	-	

Auxiliary functions (AUXF) – Basic parameters -

Description of parameter values	Index of the float-coded auxiliary function. The highest index is dependent on the number of existing float-coded auxiliary functions in the class Class[1..30]/Float (parameter "UsabAuxFunc" in the folder AUXF/NofAuxFunc/).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00080	
Data type	Byte0_32_t (byte) 2 byte integer, unsigned for MACODA compatibility	

13.3.8 Double-coded auxiliary functions of auxiliary function classes class [1..30]

Acknowledgement requirement

Acknowledgement requirement		AcknDouble
Specifies whether an double-coded auxiliary function defined under Class[1..30]/Double requires acknowledgement.		
Path	AUXF/ClassDef/Class[1..30]/Double/AcknDouble	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No Auxiliary function does not require acknowledgement
	1	Yes Auxiliary function requires acknowledgement
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00090	
Data type	Byte0_1_t (byte) 2 byte integer, unsigned for MACODA compatibility	

Index

index	IndDouble
Specifies the "index" of a double-coded auxiliary function defined under ../Class[1..30]/Double.	
Their index determines the order in which several auxiliary functions belonging to a specific class of auxiliary functions are stored in the marker filed of class Class[1..30]/Double.	

Path	AUXF/ClassDef/Class[1..30]/Double/IndDouble	
Value range	0 .. 32	
Pre-assignment	0	.../Class[1..30]/Double/IndDouble
Unit	-	
Description of parameter values	Index of the double-coded auxiliary function. The highest index is dependent on the number of existing double-coded auxiliary functions in the class Class[1..30]/Double (parameter "UsabAuxFunc" in the folder AUXF/NofAuxFunc/).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00080	
Data type	Byte0_32_t (byte) 2 byte integer, unsigned for MACODA compatibility	

13.4 Acknowledgement behavior for bit-coded auxiliary functions (BitAckn)

13.4.1 General information

The acknowledgement requirement of every bit of a bit-coded auxiliary function is specified channel-independently in the parameters **Byte000** to **Byte191**, which cover an address range of 1536 bits.

The address range is divided into 12 sequential segments, each of with 16 bytes, each of which has 8 bits (12x16x8 bits = 1536 bits) (see the figure "bit array for M19, M48 and M90" and [chapter 13.10 "Parameterizing example: Auxiliary functions" on page 231](#)).

13.4.2 Acknowledgement requirement from bit 0000 to bit 0127 (Byte000_015)

Acknowledgment Requirement from Bit 0000 to Bit 0127 (Byte000_015)	Byte000 to Byte015

Specifies channel-independently from bit 0000 to bit 0127 in Byte000_015 the acknowledgement requirement of bit-coded auxiliary functions.

Path	AUXF/BitAckn/Byte000_015/Byte000 to .../Byte015	
Explanations	-	
Value range	00000000 .. 11111111	
Pre-assignment	00000000	AUXF/BitAckn/Byte000_015/Byte000 (bit 0007-0000)
	00000000	AUXF/BitAckn/Byte000_015/Byte001 (bit 0015-0008)
	00001000	AUXF/BitAckn/Byte000_015/Byte002 (bit 0023-0016)
	00000000	AUXF/BitAckn/Byte000_015/Byte003 (bit 0031-0024)
	00000000	AUXF/BitAckn/Byte000_015/Byte004 (bit 0039-0032)
	00011111	AUXF/BitAckn/Byte000_015/Byte005 (bit 0047-0040)
	00000001	AUXF/BitAckn/Byte000_015/Byte006 (bit 0055-0048)

Auxiliary functions (AUXF) – Basic parameters -

	00000000	AUXF/BitAckn/Byte000_015/Byte007 (bit 0063-0056)
	...	...
	00000000	AUXF/BitAckn/Byte000_015/Byte015 (bit 0127-0120)
Unit	-	
Description of parameter values	Bit mask (8 bit) of byte xxx (see following example): A "0" in one place in the bit mask means "acknowledgement not required" A "1" in one place in the bit mask means "acknowledgement required"	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00020	
Data type	Str8_8_t (isoLatin1String)	
	4 byte integer, unsigned for MACODA compatibility	
	<i>Example:</i>	
	Bit-coded auxiliary functions. Acknowledgement requirement for M19, M48 and M90 Defined is a bit-coded auxiliary function named "M" and a bit array size of 96 bits: Start Bit Address: 0 (AUXF/ClassDef/ClassDef[1]/Bit/StartBitAddr); End bit address: 95 (AUXF/ClassDef/ClassDef[1]/Bit/EndBitAddr). The individual bits can be programmed using the NC syntax "M0" up to "M95": Programming code of first bit: 0 (AUXF/ClassDef/ClassDef[1]/Bit/StartCode); Programming code of last bit: 95 (AUXF/ClassDef/ClassDef[1]/Bit/EndCode) M19, M48 and M90 are to require acknowledgement:	

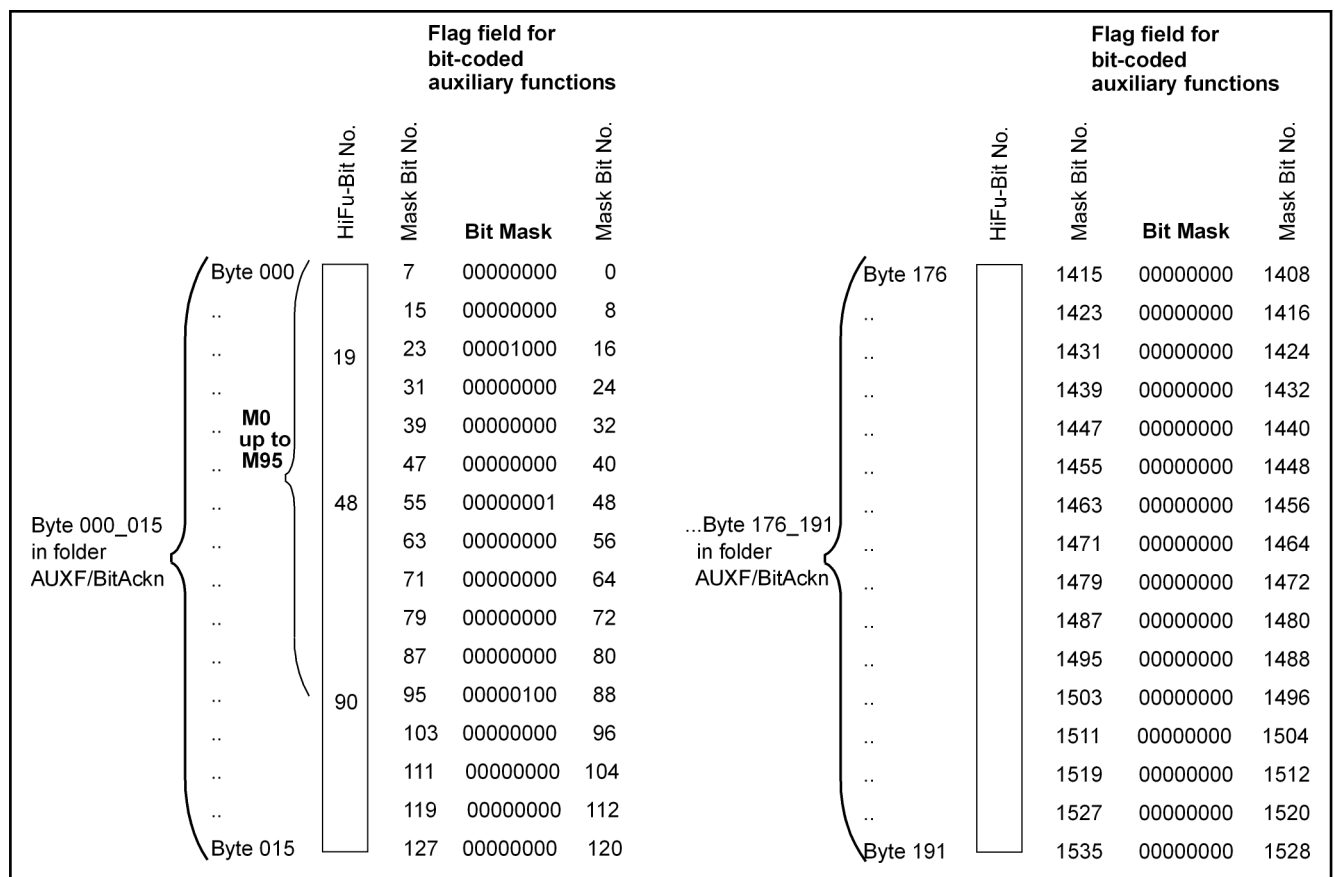


Fig. 13-9: Bit array for M19, M48 and M90

13.4.3 Acknowledgement requirement from bit 0128 to bit 1535 (Byte016 to Byte191)

Acknowledgement requirement bit 0128 to bit 1535 (Byte016_31 to Byte176_191)	Byte016 to Byte191

Specifies channel-independently from bit 0128 to bit 1535 in Byte016 to Byte191 the acknowledgement requirement of bit-coded auxiliary functions.

Path AUXF/BitAckn/Byte016_031 to .../Byte176_191

Value range 00000000 .. 11111111

Pre-assignment 00000000 AUXF/BitAckn/Byte016_031/Byte016 (bit 0135-0128)

...

00000000 AUXF/BitAckn/Byte176_191/Byte191 (bit 1535-1528)

Unit -

Description of parameter values Bit mask (8 bit) of byte xxx:

A "0" in one place in the bit mask means "acknowledgement not required"

A "1" in one place in the bit mask means "acknowledgement required"

Access rights **MTB:** Write

Setter: Read

Operator: Read

Activation System reset (SR)

Dependency	-
Macoda ID	3010 00020
Data type	Str8_8_t (isoLatin1String) 4 byte integer, unsigned for MACODA compatibility

13.5 Auxiliary function class definition (GrDef)

13.5.1 General information

Up to 64 auxiliary function groups Gr[1..64] can be defined here and individual auxiliary functions GrAuxFunc[1..15] assigned to these auxiliary function groups.

13.5.2 Channel dependency

Channel dependency		GrChDep
Defines whether each auxiliary function group Gr[1..64] is channel-independent or channel-dependent.		
Path	AUXF/GrDef/Gr[1..64]/GrChDep	
Value range	0	behavior specified automatically
	1	Channel-independent auxiliary function
	2	Channel-dependent auxiliary function (MACODA compatibility: 0 ≙ <empty>, 1 ≙ "A:", 2 ≙ "S:")
Pre-assignment	1 (channel-independent auxiliary function)	→ AUXF/GrDef/Gr[1]/GrChDep
	1 (channel-independent auxiliary function)	→ AUXF/GrDef/Gr[2]/GrChDep
	2 (channel-dependent auxiliary function)	→ AUXF/GrDef/Gr[3]/GrChDep
	0 (behavior specified automatically)	→ AUXF/GrDef/Gr[4..64]/GrChDep
Unit	-	
Description of parameter values	0	Behavior of the auxiliary function group Gr[1..64] is set automatically, depending on whether the contained auxiliary functions are channel-independent or channel-dependent. When mixed, the group cannot be used, and a warning is generated.
	1	Auxiliary function group Gr[1..64] is channel-independent
	2	Auxiliary function group Gr[1..64] is channel dependent
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3010 01000 [1] .. 3010 01063 [1]	
Data type	Byte0_2_t (byte) 8 byte string for MACODA compatibility	

13.5.3 Auxiliary function[1..15]

AuxiliaryFunction[1..15]		GrAuxFunc[1..15]
Defines the syntax of the individual auxiliary functions GrAuxFunc[1..15] in the auxiliary function groups Gr[1..64].		
Path	AUXF/GrDef/Gr[1..64]/GrAuxFunc[1..15]	
Value range	{Text}	
Pre-assignment	M3	.../Gr[1]/GrAuxFunc[1]
	M4	.../Gr[1]/GrAuxFunc[2]
	M5	.../Gr[1]/GrAuxFunc[3]
	M19	.../Gr[1]/GrAuxFunc[4]
	-	.../Gr[1]/GrAuxFunc[5..15]
	S	.../Gr[2]/GrAuxFunc[1]
	-	.../Gr[2]/GrAuxFunc[2..15]
	T	.../Gr[3]/GrAuxFunc[1]
	-	.../Gr[3]/GrAuxFunc[2..15]
	-	.../Gr[4..64]/GrAuxFunc[1..15]
Unit	-	
Description of parameter values	Name of the auxiliary function [1..15] in the auxiliary function group Gr[1..64].	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3010 01000 [2..16] .. 3010 01063 [2..16]	
Data type	Byte0_2_t (byte)	
	8 byte string for MACODA compatibility	

13.6 Global output behavior (OutBehav)

13.6.1 General information

The parameter **OutBehavMode** specifies how programmed auxiliary functions are transferred to the PLC application program.

13.6.2 Output type

Output type		OutBehavMode
Defines the method of transferring the auxiliary functions programmed in an NC block to the PLC application program.		
Path	AUXF/OutBehav/OutBehavMode	
Value range	0	Only bit-coded auxiliary functions synchronously output to PLC.
	1	Output all auxiliary functions synchronously to PLC.
Pre-assignment	1	Output all auxiliary functions synchronously to PLC.
Unit	-	

Auxiliary functions (AUXF) – Basic parameters -

Description of parameter values	0	Only the bit-coded auxiliary functions of an NC block are collected and then transferred together to the PLC. All auxiliary functions coded otherwise are transferred immediately.
	1	All auxiliary functions of an NC block are transferred synchronously to the PLC.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3010 00015	
Data type	Byte0_1_t (byte)	
	2 byte integer, unsigned for MACODA compatibility	

13.7 Number of usable auxiliary functions (NofAuxFunc)

13.7.1 General information

Parameter "Usable auxiliary functions" UsabAuxFunc (in folder AUXF/NofAuxFunc/) reduces the usable number of all auxiliary functions defined in folders AUXF/ClassDef/Class[1..30] as needed. This way, auxiliary functions which are, for instance, elements of extended machine functions can be quickly disabled or enabled.

The remaining parameters in the folders AUXF/NofAuxFunc/ChDep and AUXF/NofAuxFunc/ChIndep specify how many channel-dependent and channel-independent auxiliary functions from the individual auxiliary function classes can be used. For bit-coded auxiliary functions, the number of all individual bits is specified.

13.7.2 Number of auxiliary functions

Usable auxiliary functions

Usable auxiliary functions	UsabAuxFunc

Specifies how many auxiliary functions are used in the individual auxiliary function classes AUXF/ClassDef/Class[1..30].

With this parameter, set the usable number of all auxiliary functions defined in the individual auxiliary function classes AUXF/ClassDef/Class[1..30] as needed. This way, auxiliary functions which are, for instance, elements of extended machine functions can be quickly disabled or enabled.

Path	AUXF/NofAuxFunc/UsabAuxFunc
Value range	0 .. 1536
Pre-assignment	3
Unit	-
Description of parameter values	Total number of all usable auxiliary functions defined in the auxiliary function classes in the system.
	If the number of auxiliary functions existing there exceeds the number specified here, only the initially defined auxiliary functions become effective (related to the index of the individual parameters).

"Usable auxiliary functions": UsabAuxFunc	"Name" of the bit auxiliary function: AuxFuncName	Auxiliary function is effective
3	M	Yes
	M500	Yes
	M999	Yes
	O	No
	S?	No

Tab. 13-1: Example of reducing usable auxiliary functions

Access rights	MTB: Write
	Setter: Read
	Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	3010 00010 [1]
Data type	NofAuxFunc_t (int)
	4 byte integer, unsigned for MACODA compatibility

13.7.3 Channel-dependent auxiliary functions (ChDep)

Bit auxiliary function

Bit auxiliary function		ChDepBit
Specifies how many individual bits of the bit-coded auxiliary functions are used.		
Path	AUXF/NofAuxFunc/ChDep/ChDepBit	
Value range	0 .. 1536	
Pre-assignment	304	
Unit	-	
Description of parameter values	Number of individual bits of the bit-coded auxiliary functions per channel.	
Access rights	MTB: Write	
	Setter: Read	
	Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3010 00010 [2]	
Data type	NofAuxFunc_t (int)	
	4 byte integer, unsigned for MACODA compatibility	

BCD32 auxiliary function

BCD32 auxiliary function	ChDepBcd32
Specifies the number of channel-dependent Bcd32-coded auxiliary functions used.	
Path	AUXF/NofAuxFunc/ChDep/ChDepBcd32

Auxiliary functions (AUXF) – Basic parameters -

Value range	0 .. 1536
Pre-assignment	0
Unit	-
Description of parameter values	Number of channel-dependent Bcd32-coded auxiliary functions.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	3010 00010 [5]
Data type	NofAuxFunc_t (int) 4 byte integer, unsigned for MACODA compatibility

Integer auxiliary function

Integer auxiliary function		ChDepInt
	Specifies the number of channel-dependent integer-coded auxiliary functions used.	
Path	AUXF/NofAuxFunc/ChDep/ChDepInt	
Value range	0 .. 1536	
Pre-assignment	1	
Unit	-	
Description of parameter values	Number of channel-dependent integer-coded auxiliary functions	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3010 00010 [10]	
Data type	NofAuxFunc_t (int) 4 byte integer, unsigned for MACODA compatibility	

Float auxiliary function

Float auxiliary function		ChDepFloat
	Specifies the number of channel-dependent float-coded auxiliary used.	
Path	AUXF/NofAuxFunc/ChDep/ChDepFloat	
Value range	0 .. 1536	
Pre-assignment	0	
Unit	-	
Description of parameter values	Number of channel-dependent float-coded auxiliary functions.	
Access rights	MTB: Write Setter: Read Operator: Read	

Activation	System restart (CS)
Dependency	-
Macoda ID	3010 00010 [11]
Data type	NofAuxFunc_t (int) 4 byte integer, unsigned for MACODA compatibility

Double auxiliary function

Double auxiliary function		ChDepDouble
Specifies the number of channel-dependent double-coded auxiliary functions used.		
Path	AUXF/NofAuxFunc/ChDep/ChDepDouble	
Value range	0 .. 1536	
Pre-assignment	0	
Unit	-	
Description of parameter values	Number of channel-dependent double-coded auxiliary functions	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3010 00010 [12]	
Data type	NofAuxFunc_t (int) 4 byte integer, unsigned for MACODA compatibility	

13.7.4 Channel-independent auxiliary functions (ChInDep)

BCD32 auxiliary function

BCD32 auxiliary function		ChInDepBcd32
Specifies the number of channel-independent Bcd32-coded auxiliary functions used.		
Path	AUXF/NofAuxFunc/ChInDep/ChInDepBcd32	
Value range	0 .. 1536	
Pre-assignment	0	
Unit	-	
Description of parameter values	Number of channel-independent Bcd32-coded auxiliary functions	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3010 00010 [3]	
Data type	NofAuxFunc_t (int) 4 byte integer, unsigned for MACODA compatibility	

Auxiliary functions (AUXF) – Basic parameters -

Integer auxiliary function

Integer auxiliary function		ChIndepInt
	Specifies the number of channel-independent integer-coded auxiliary functions used.	
Path	AUXF/NofAuxFunc/ChInDep/ChIndepInt	
Value range	0 .. 1536	
Pre-assignment	0	
Unit	-	
Description of parameter values	Number of channel-independent integer-coded auxiliary functions.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3010 00010 [7]	
Data type	NofAuxFunc_t (int) 4 byte integer, unsigned for MACODA compatibility	

Float auxiliary function

Float auxiliary function		ChIndepFloat
	Specifies the number of channel-independent float-coded auxiliary functions used.	
Path	AUXF/NofAuxFunc/ChInDep/ChIndepFloat	
Value range	0 .. 1536	
Pre-assignment	1	
Unit	-	
Description of parameter values	Number of channel-independent float-coded auxiliary functions.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3010 00010 [8]	
Data type	NofAuxFunc_t (int) 4 byte integer, unsigned for MACODA compatibility	

Double auxiliary function

Double auxiliary function		ChIndepDouble
	Specifies the number of channel-independent double-coded auxiliary functions used.	
Path	AUXF/NofAuxFunc/ChInDep/ChIndepDouble	
Value range	0 .. 1536	
Pre-assignment	0	

Unit	-
Description of parameter values	Number of channel-independent double-coded auxiliary functions.
Access rights	MTB: Write
	Setter: Read
	Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	3010 00010 [9]
Data type	NofAuxFunc_t (int)
	4 byte integer, unsigned for MACODA compatibility

13.8 Addresses of auxiliary functions in the PLC interface (PlcIf)

13.8.1 General information

The parameter in the folders

AUXF/PlcIf/ChDep/ChIf[1..60]/.. and AUXF/PlcIf/ChIndep/..

specify the addresses in the PLC interface for all auxiliary function classes.

13.8.2 Addresses in the channel interface[1..60] of channel-dependent auxiliary functions

Address of the bit-coded auxiliary functions

Address of the bit-coded auxiliary functions	AddrChDepBit
Specifies the starting addresses of the ranges for bit auxiliary functions for each channel (channel interface ChIf[1..60]).	
When specifying individual address ranges, plan gaps for a future expansion of bit-coded auxiliary functions. Thus, addressing of existing bit-coded auxiliary functions in the PLC program do not have to be adjusted if additional bit-coded auxiliary functions are implemented in the system.	
When additional channels are applied, the corresponding new marker fields can be placed right in some unused PLC address ranges.	
	In the absence of sufficiently large gaps, ensure that existing ranges do not overlap with new ones when adding new auxiliary functions!

Path	AUXF/PlcIf/ChDep/ChIf[1..60]/AddrChDepBit	
Value range	-1, total integer range	
Pre-assignment	4700	.../ChDep//ChIf[1]/AddrChDepBit
	4738	.../ChDep//ChIf[2]/AddrChDepBit
	4776	.../ChDep//ChIf[3]/AddrChDepBit
	-1	.../ChDep//ChIf[4..60]/AddrChDepBit
Unit	-	
Description of parameter values	-1	The interface is not used; signals are not copied
	>= 0	Starting addresses for channel-dependent bit-coded auxiliary functions
Access rights	MTB:	Write

Auxiliary functions (AUXF) – Basic parameters -

	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00009	
Data type	int	
	4 byte integer, unsigned for MACODA compatibility	

Example:

Starting addressing relating to channel interface in AddrChDepBit

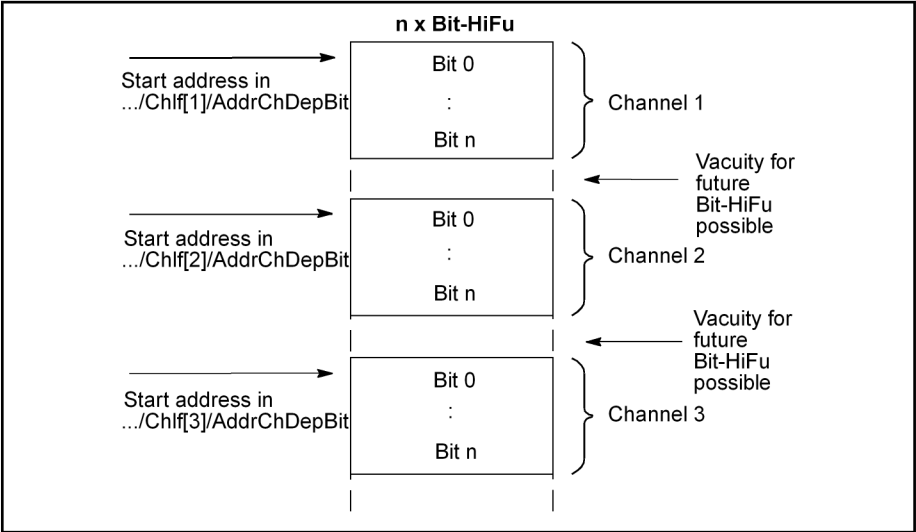


Fig. 13-10: Extension options for bit-coded auxiliary functions

Address of the BCD32-coded auxiliary functions

Address of BCD32-coded auxiliary functions		AddrChDepBcd32
Specifies the starting addresses of the ranges for BCD32 auxiliary functions for each channel (channel interface ChIf[1..60]).		
Path	AUXF/Plclf/ChDep/ChIf[1..60]/AddrChDepBcd32	
Value range	-1, total integer range	
Pre-assignment	-1 .../ChDep/ChIf[1..60]/AddrChDepBcd32	
Unit	-	
Description of parameter values	-1	The interface is not used; signals are not copied
	>= 0	Starting addresses for channel-dependent Bcd32-coded auxiliary functions
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00010	
Data type	int	
	4 byte integer, unsigned for MACODA compatibility	

Address of the integer-coded auxiliary functions

Address of the integer-coded auxiliary functions		AddrChDepInt
Specifies the starting addresses of the ranges for integer auxiliary functions for each channel (channel interface ChIf[1..60]).		
Path	AUXF/PlcIf/ChDep/ChIf[1..60]/AddrChDepInt	
Value range	-1, total integer range	
Pre-assignment	6200	.../ChDep/ChIf[1]/AddrChDepInt
	6216	.../ChDep/ChIf[2]/AddrChDepInt
	6232	.../ChDep/ChIf[3]/AddrChDepInt
	6248	.../ChDep/ChIf[4]/AddrChDepInt
	6264	.../ChDep/ChIf[5]/AddrChDepInt
	6280	.../ChDep/ChIf[6]/AddrChDepInt
	6296	.../ChDep/ChIf[7]/AddrChDepInt
	6312	.../ChDep/ChIf[8]/AddrChDepInt
	6328	.../ChDep/ChIf[9]/AddrChDepInt
	6348	.../ChDep/ChIf[10]/AddrChDepInt
	6344	.../ChDep/ChIf[11]/AddrChDepInt
	6360	.../ChDep/ChIf[12]/AddrChDepInt
	-1	.../ChDep/ChIf[13..60]/AddrChDepInt
Unit	-	
Description of parameter values	-1	The interface is not used; signals are not copied
	>= 0	Starting addresses for channel-dependent integer-coded auxiliary functions
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00012	
Data type	int	
	4 byte integer, unsigned for MACODA compatibility	

Address of the float-coded auxiliary functions

Address of the float-coded auxiliary functions		AddrChDepFloat
Specifies the starting addresses of the ranges for float auxiliary functions for each channel (channel interface ChIf[1..60]).		
Path	AUXF/PlcIf/ChDep/ChIf[1..60]/AddrChDepFloat	
Value range	-1, total integer range	
Pre-assignment	-1	.../ChDep/ChIf[1..60]/AddrChDepFloat
Unit	-	
Description of parameter values	-1	The interface is not used; signals are not copied
	>= 0	Starting addresses for channel-dependent float-coded auxiliary functions

Auxiliary functions (AUXF) – Basic parameters -

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00013	
Data type	int	
	4 byte integer, unsigned for MACODA compatibility	

Address of the double-coded auxiliary functions

Address of the double-coded auxiliary functions		AddrChDepDouble
Specifies the starting addresses of the ranges for double-coded auxiliary functions for each channel (channel interface Chlf[1..60]).		
Path	AUXF/Plclf/ChDep/Chlf[1..60]/AddrChDepDouble	
Value range	-1, total integer range	
Pre-assignment	-1 .../ChDep/Chlf[1..60]/AddrChDepDouble	
Unit	-	
Description of parameter values	-1	The interface is not used; signals are not copied
	>= 0	Starting addresses for channel-dependent double-coded auxiliary functions
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00014	
Data type	int	
	4 byte integer, unsigned for MACODA compatibility	

13.8.3 Addresses of channel-independent auxiliary functions

Address of the BCD32 auxiliary functions

Address of BCD32 auxiliary functions		AddrChIndepBcd32
Path	AUXF/Plclf/ChIndep/AddrChIndepBcd32	
Purpose	Specifies the starting addresses of the PLC address range for channel-independent Bcd32-coded auxiliary functions.	
Explanations	-	
Value range	-1, total integer range	
Pre-assignment	-1	
Unit	-	
Description of parameter values	-1	The interface is not used; signals are not copied
	>= 0	Starting address for channel-independent Bcd32-coded auxiliary functions

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00003	
Data type	int	
	4 byte integer, unsigned for MACODA compatibility	

Address of the integer-coded auxiliary functions

Address of the integer-coded auxiliary functions		AddrChIndepInt
Specifies the starting address of the PLC address range for channel-independent integer-coded auxiliary functions.		
Path	AUXF/PlcIf/ChInDep/AddrChIndepInt	
Value range	-1, total integer range	
Pre-assignment	-1	
Unit	-	
Description of parameter values	-1	The interface is not used; signals are not copied
	≥ 0	Starting address for channel-independent integer-coded auxiliary functions
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00020	
Data type	int	
	4 byte integer, signed for MACODA compatibility	

Address of the float-coded auxiliary functions

Address of the float-coded auxiliary functions		AddrChIndepFloat
Specifies the starting address of the PLC address range for channel-independent float-coded auxiliary functions.		
Path	AUXF/PlcIf/ChInDep/AddrChIndepFloat	
Value range	-1, total integer range	
Pre-assignment	6024	
Unit	-	
Description of parameter values	-1	The interface is not used; signals are not copied
	≥ 0	Starting address for channel-independent float-coded auxiliary functions
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	

Auxiliary functions (AUXF) – Basic parameters -

Dependency	-
Macoda ID	2060 00020
Data type	int
	4 byte integer, signed for MACODA compatibility

Address of the double-coded auxiliary functions

Address of the double-coded auxiliary functions		AddrChIndepDouble
Specifies the starting address of the PLC address range for channel-independent double-coded auxiliary functions.		
Path	AUXF/PlcIf/ChInDep/AddrChIndepDouble	
Value range	-1, total integer range	
Pre-assignment	-1	
Unit	-	
Description of parameter values	-1	The interface is not used; signals are not copied
	>= 0	Starting address for channel-independent double-coded auxiliary functions
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2060 00020	
Data type	int	
	4 byte integer, unsigned for MACODA compatibility	

13.9 Defining a new auxiliary function

The following section deals with how to parameterize a new auxiliary function.



An example of parameterization of an entire configuration of auxiliary functions is given in [chapter 13.10 "Parameterizing example: Auxiliary functions" on page 231](#).

Proceed as follows:

1. Open the necessary folder **AUXF/ClassDef/Class[1..30]** and, in the parameter **AuxFuncName**, define the NC syntax of the desired auxiliary function.

Please note the explanations given in [chapter 13.1 "Auxiliary function basics" on page 199](#).

2. To do so, select the necessary folder **AUXF/ClassDef/Class[1..30]/** and, with the right mouse button, select the desired auxiliary function type with the function "Convert to..." from the context menu.

Depending on the selection, one of the following subfolders is displayed:

- Bit
- Bcd32
- Int
- Float
- Double

Always define the auxiliary function type first and then the range of action

3. In the parameter **AuxFuncChDep**, specify whether the auxiliary function should be channel-dependent or channel-independent (range of action).
Note that for bit-coded auxiliary functions, only the "channel-dependent" setting is supported.
4. Open the folder **AUXF/NofAuxFunc** and adjust in the parameter **UsabAuxFunc** the number of all auxiliary functions defined in the folders **AUXF/ClassDef/Class[1..30]**.
. If an auxiliary function or a coherent block of auxiliary functions should be hidden as seen from the highest index, reduce this parameter accordingly.
5. Adjust the number of channel-dependent or channel-independent auxiliary functions in the folder **AUXF/NofAuxFunc/ChDep** or **AUXF/NofAuxFunc/ChIndep**.
For bit-coded auxiliary functions, enter at least the number of all individual bits used (highest bit address used + 1). It is recommended to set this number to the next highest value divisible by 8 (byte raster). Thus, you are also prepared for future extensions to the bit-coded auxiliary functions.
6. *Depending on the auxiliary function type chosen, continue with the following points:*
 - Bit-coded: Point 7
 - 32-bit BCD-coded: Point 9
 - Integer-coded: Point 10
 - Float-coded: Point 11
 - Double-coded: Point 12
7. *For bit-coded auxiliary functions only:*
 - Open the required folder **AUXF/ClassDef/Class[1..30]/Bit**.
 - In the parameter **StartCode**, enter the programming code of the first bit, and in **EndCode**, the programming code of the last bit of the auxiliary function.
 - In the parameter **StartBitAddr**, enter the address of the first bit, and in **EndBitAddr**, the address of the last bit of the auxiliary function.

Example:

Bit-coded auxiliary function named "RUDI"

Address of the first bit: 301

Address of the last bit: 320

Program code of the first bit: 1

Program code of the last bit: 20

⇒ Programming of "RUDI1" sets bit 301 in the flag field.

⇒ Programming of "RUDI20" sets bit 320 in the flag field.

8. For bit-coded auxiliary functions only:
Specify the acknowledgement requirement of the auxiliary function.

Auxiliary functions (AUXF) – Basic parameters -

To do so, open the necessary folder **AUXF/BitAckn** and, in the appropriate subfolder, select the parameter **Bytexxx** in which the bit to be changed is found.

The parameter values begin with the higher order bit.

Continue with point 13.

9. *For 32-bit BCD-coded auxiliary functions only:*

- Open the required folder
AUXF/ClassDef/Class[1..32]/Bcd32.
- In the parameter **AcknBcd32**, specify the acknowledgement requirement of the auxiliary function.
- In the parameter **PrelPlaceBcd32**, specify the number of integer places (to the left of the decimal point), and in **DecPlaceBcd32**, the number of decimal places of the auxiliary function.
- In the parameter **IndBcd32**, enter the index of the auxiliary function. This allows to arrange the order in which several auxiliary functions of the same auxiliary function class are stored in the marker field of this class. The index is independent of the number of channel-dependent or channel-independent Bcd32-coded auxiliary functions in the folder

AUXF/NofAuxFunc/ChDep or **AUXF/NofAuxFunc/ChIndep.**

Example:

The channel-dependent 32-bit BCD auxiliary functions

- STEVE (Index: 2),
- SANDY (Index: 0)

are stored in the marker field of this auxiliary function class in the order

- 0. SANDY,
- 1. <64 bit free>
- 2. STEFAN

.

Please note that STEVE can only contain the index 2 if at least 3 channel-dependent auxiliary functions are defined in the parameter "ChDepBcd32" in the folder AUXF/NofAuxFunc/ChDep.

Continue with point 13.

10. *For integer-coded auxiliary functions only:*

- Open the required folder
AUXF/ClassDef/Class[1..30]/Int.
- In the parameter **AcknInt**, specify the acknowledgement requirement of the auxiliary function.
- In the parameter **IndInt**, enter the index of the auxiliary function. This allows to arrange the order in which several auxiliary functions of the same auxiliary function class are stored in the marker field of this class. The index is independent of the number of channel-dependent or channel-independent integer-coded auxiliary functions in the folder

AUXF/NofAuxFunc/ChDep or **AUXF/NofAuxFunc/ChIndep.**

Continue with point 13.

11. *For float-coded auxiliary functions only:*

- Open the required folder
AUXF/ClassDef/Class[1..30]/Float.
- In the parameter **AcknFloat**, specify the acknowledgement requirement of the auxiliary function.
- In the parameter **IndFloat**, enter the index of the auxiliary function. This allows to arrange the order in which several auxiliary functions of the same auxiliary function class are stored in the marker field of this class. The index is independent of the number of channel-dependent or channel-independent float-coded auxiliary functions in the folder

AUXF/NofAuxFunc/ChDep or **AUXF/NofAuxFunc/ChIndep.**

Continue with point 13.

12. *For double-coded auxiliary functions only:*

- Open the required folder
AUXF/ClassDef/Class[1..30]/Double.
- In the parameter **AcknDouble**, specify the acknowledgement requirement of the auxiliary function.
- In the parameter **IndDouble**, enter the index of the auxiliary function. This allows to arrange the order in which several auxiliary functions of the same auxiliary function class are stored in the marker field of this class. The index is independent of the number of channel-dependent or channel-independent double-coded auxiliary functions in the folder

AUXF/NofAuxFunc/ChDep or **AUXF/NofAuxFunc/ChIndep.**

13. Open the folder **AUXF/OutBehav** and specify in the parameter **OutBehavMode** how the programmed auxiliary function should be transferred to the PLC application program.

14. For a channel-dependent auxiliary function, open the folder

AUXF/PlcIf/ChDep/ChIf[1..60] of the desired channel, or, for a channel-independent auxiliary function, open the folder **AUXF/PlcIf/ChIndep.**

Adjust the address of the auxiliary function in the PLC interface in the corresponding parameter as necessary (e.g. **AddrChDepBcd32**).

15. Only if assignment to an auxiliary function group is desired:

Open the necessary folder **AUXF/GrDef/Gr[1..64]**

and supplement the syntax of the new auxiliary function (e.g. **M19**) in the parameter **GrAuxFunc[i]**.

The definition of an auxiliary function is now completed.

13.10 Parameterizing example: Auxiliary functions

Task Parameterize the auxiliary functions **M (M0..M299)**, **M (M500)**, **M (M999)**, **O**, **S**, **S1**, **S2**, **V** and **T**. The output of all auxiliary functions to the PLC application program is to be synchronous. The PLC interface is designed for maximum 4 channels. Auxiliary functions are not now assigned to auxiliary function groups. The auxiliary functions are to have the following properties in detail:

M (M0..M299): 1. auxiliary function, bit-coded, channel-dependent,
bit array with a length of 300 bits,
Bit addresses from 0..299, programming as **M0..M299**,

Auxiliary functions (AUXF) – Basic parameters -

- M19, M40..M44, M48 and M95 require acknowledgement.
- M (M500):** 2. auxiliary function, bit-coded, channel-dependent, individual bit at bit address 300, programming as M500, no acknowledgement required.
- M (M999):** 3. auxiliary function, bit-coded, channel-dependent, individual bit at bit address 199, programming as M999, no acknowledgement required.
- O:** 4. auxiliary function, bit-coded, channel-dependent, bit array with 20 bits in length, Bit addresses from 301..320, programming as O1..O20, all bits require acknowledgement.
- S:** 5. auxiliary function, float-coded, channel-independent, is in first position of the marker area of the channel-independent float auxiliary functions, i.e. index = 0, no acknowledgement required.
- S1 :** 6. auxiliary function, float-coded, channel-dependent, is in first position of the marker area of the channel-dependent float auxiliary functions, i.e. index = 0, acknowledgement required.
- S2 :** 7. auxiliary function, float-coded, channel-dependent, is in second position of the marker area of the channel-dependent float auxiliary functions, i.e. index = 1, acknowledgement required.
- V:** 8. auxiliary function 32-bit BCD-coded, channel-independent, 8 integer places (to the left of the decimal point), no decimal places, is in first position of the marker area of the channel-independent 32 bit BCD auxiliary functions, i.e. index = 0, no acknowledgement required.
- T:** 9. auxiliary function, integer-coded, channel-dependent, is in first position of the marker area of the channel-dependent integer auxiliary functions, i.e. index = 0, no acknowledgement required.

Required Parameter Settings In NC Configurator, the following settings are made according to the approach described in [chapter 13.9 "Defining a new auxiliary function" on page 228](#) when parameterizing a "new auxiliary function":

Auxiliary Function Names: AUXF/ClassDef/Class[1]/AuxFuncName: M
/Class[2]/AuxFuncName: M
/Class[3]/AuxFuncName: M
/Class[4]/AuxFuncName: O
/Class[5]/AuxFuncName: S?
/Class[6]/AuxFuncName: S1?
/Class[7]/AuxFuncName: S2?
/Class[8]/AuxFuncName: V
/Class[9]/AuxFuncName: T

Auxiliary Function Type: AUXF/ClassDef/Class[1]: Convert to → Bit
/Class[2]: Convert to → Bit
/Class[3]: Convert to → Bit
/Class[4]: Convert to → Bit
/Class[5]: Convert to → Float
/Class[6]: Convert to → Float
/Class[7]: Convert to → Float
/Class[8]: Convert to → Bcd32
/Class[9]: Convert to → Int

The parameter **EnablAuxFunc** is automatically set to **Yes** for this auxiliary function.

Range of action: AUXF/ClassDef/Class[1]/AuxFuncChDep: channel-independent
/Class[2]/AuxFuncChDep: channel-dependent
/Class[3]/AuxFuncChDep: channel-dependent
/Class[4]/AuxFuncChDep: channel-dependent
/Class[5]/AuxFuncChDep: channel-independent
/Class[6]/AuxFuncChDep: channel-dependent
/Class[7]/AuxFuncChDep: channel-dependent
/Class[8]/AuxFuncChDep: channel-independent
/Class[9]/AuxFuncChDep: channel-dependent

Number of Auxiliary Functions: AUXF/NofAuxFunc/UsabAuxFunc: 9
 channel-dependent:
 AUXF/NofAuxFunc/ChDep/ChDepBit: 336



The highest bit address used within all bit auxiliary functions in this example is 320 (see auxiliary function "O").

The parameter **ChDepBit** could therefore be assigned the value 321. In order to be prepared for extensions to the bit auxiliary functions, however, and to set the end of the bit auxiliary function range to a byte raster, **ChDepBit** is assigned the value 336.

AUXF/NofAuxFunc/ChDep/ChDepBcd32: 0
/ChDepInt: 1
/ChDepFloat: 2
/ChDepDouble: 0

channel-independent:

AUXF/NofAuxFunc/ChIndep/ChIndepBcd32: 1
/ChIndepInt: 0
/ChIndepFloat: 1
/ChIndepDouble: 0

Addresses, programming code and acknowledgement requirement of M (M0..M299):

AUXF/ClassDef/Class[1]/Bit/StartCode: 0
/EndCode: 299
/StartBitAddr: 0
/EndBitAddr: 299

Auxiliary functions (AUXF) – Basic parameters -

	AUXF/BitAckn/Byte000_015/Byte002/: 00001000 (Bit 19)/Byte005/: 00011111 (Bits 44..40)/Byte006/: 00000001 (Bit 48)/Byte011/: 10000000 (Bit 95)
Addresses and programming code of M (500):	AUXF/ClassDef/Class[2]/Bit/StartCode: 500/EndCode: 500/StartBitAddr: 300/EndBitAddr: 300
Addresses and programming code of M (999):	AUXF/ClassDef/Class[3]/Bit/StartCode: 999/EndCode: 999/StartBitAddr: 199/EndBitAddr: 199
Addresses, Programming Code and Acknowledgement Requirement of O:	AUXF/ClassDef/Class[4]/Bit/StartCode: 1/EndCode: 20/StartBitAddr: 301/EndBitAddr: 320 AUXF/BitAckn/Byte032_047/Byte037/: 11100000 (Bits 303..301)/Byte038/: 11111111 (Bits 311..304)/Byte039/: 11111111 (Bits 319..312)/Byte040/: 00000001 (Bit 320)
Acknowledgement Requirement and Index of S:	AUXF/ClassDef/Class[5]/Float/AcknFloat: no/IndFloat: 0
Acknowledgement Requirement and Index of S1:	AUXF/ClassDef/Class[6]/Float/AcknFloat: yes/IndFloat: 0
Acknowledgement Requirement and Index of S2:	AUXF/ClassDef/Class[7]/Float/AcknFloat: yes/IndFloat: 1
Acknowledgement Requirement, Integer/Decimal Places and Index of V:	AUXF/ClassDef/Class[8]/Bcd32/AcknBcd32: no/PrelPlaceBcd32: 8/DeciPlaceBcd32: 0/IndBcd32: 0
Acknowledgement Requirement, Integer/Decimal Places and Index of T:	AUXF/ClassDef/Class[9]/Int/AcknInt: no/IndInt: 0
Output Behavior to PLC	AUXF/OutBehav/OutBehavMode: All auxiliary functions synchronous...

Addresses in PLC Interface**Channel-dependent auxiliary functions:**

AUXF/PlcIf/ChDep/Ch[1]/AddrChDepBit: e.g. 5000
/Ch[2]/AddrChDepBit: e.g. 5042
/Ch[3]/AddrChDepBit: e.g. 5084
/Ch[4]/AddrChDepBit: e.g. 5126
/Ch[5..60]/AddrChDepBit: -1
 AUXF/PlcIf/ChDep/Ch[1..60]/AddrChDepBcd32: -1
 AUXF/PlcIf/ChDep/Ch[1]/AddrChDepInt: e.g. 5200
/Ch[2]/AddrChDepInt: e.g. 5208
/Ch[3]/AddrChDepInt: e.g. 5216
/Ch[4]/AddrChDepInt: z. B. 5224
/Ch[5..60]/AddrChDepInt: -1
 AUXF/PlcIf/ChDep/Ch[1]/AddrChDepFloat: e.g. 5300
/Ch[2]/AddrChDepFloat: e.g. 5316
/Ch[3]/AddrChDepFloat: e.g. 5332
/Ch[4]/AddrChDepFloat: e.g. 5348
/Ch[5..60]/AddrChDepFloat: -1
 AUXF/PlcIf/ChDep/Ch[1..60]/AddrChDepDouble: -1

Channel-independent auxiliary functions:

AUXF/PlcIf/ChIndep/AddrChIndepBcd32: e.g. 5400
/AddrChIndepInt: -1
/AddrChIndepFloat: e.g. 5416
/AddrChIndepDouble: -1

Representation of Marker Fields

The following tables show how these auxiliary functions are displayed in the marker area of the PLC.

Auxiliary functions (AUXF) – Basic parameters -

Channel-dependent bit-coded auxiliary functions:

Channel 1	Start address (e.g. 5000) according to AUXF/Plclf/ChDep/Ch[1]/AddrChDepBit	Bit 0	M0
		:	:
		Bit 199	M199 / M999
		:	:
		Bit 299	M299
		Bit 300	M500
		Bit 301	O1
		:	:
		Bit 320	O20
		:	:
		Bit 335	
...	...	...	...
Channel 4	Start address (e.g. 5126) according to AUXF/Plclf/ChDep/Ch[4]/AddrChDepBit	Bit 0	M0
		:	:
		Bit 199	M199 / M999
		:	:
		Bit 299	M299
		Bit 300	M500
		Bit 301	O1
		:	:
		Bit 320	O20
		:	:
		Bit 335	

Tab. 13-2: Bit-coded auxiliary functions M and O

Channel-dependent integer-coded auxiliary functions:

Channel 1	Start address (e.g. 5200) according to AUXF/PlcIf/ChDep/Ch[1]/AddrChDepInt	Bit 0 (acknowledge- ment) Bit 1 : Bit 31 Bit 32 (int value) : Bit 63	T
...	...	...	...
Channel 4	Start address (e.g. 5224) according to AUXF/PlcIf/ChDep/Ch[4]/AddrChDepInt	Bit 0 (acknowledge- ment) Bit 1 : Bit 31 Bit 32 (int value) : Bit 63	T

Tab. 13-3: Integer-coded auxiliary function T

Channel-dependent float-coded auxiliary functions:

Channel 1	Start address (e.g. 5300) according to AUXF/PlcIf/ChDep/Ch[1]/AddrChDepFloat	Bit 0 (acknowledge- ment) Bit 1 : Bit 31 Bit 32 (float value) : Bit 63	S1
		Bit 64 (acknowledge- ment) Bit 65 : Bit 95 Bit 96 (float value) : Bit 127	S2
...	...	...	...

Auxiliary functions (AUXF) – Basic parameters -

Channel 4	Start address (e.g. 5348) according to AUXF/Plclf/ChDep/Ch[4]/AddrChDepFloat	Bit 0 (acknowledgement) Bit 1 : Bit 31 Bit 32 (float value) : Bit 63	S1
		Bit 64 (acknowledgement) Bit 65 : Bit 95 Bit 96 (float value) : Bit 127	S2

Tab. 13-4: Float-coded auxiliary functions S1 and S2

Channel-independent 32-bit BCD-coded auxiliary functions:

Start address (e.g. 5400) according to AUXF/Plclf/ChIndep/AddrChIndepBcd32	Bit 0 (acknowledgement) Bit 1 (sign) : Bit 31 Bit 32 (BCD value) : Bit 63	V

Tab. 13-5: 32-bit BCD-coded auxiliary function V

Channel-independent float-coded auxiliary functions:

Start address (e.g. 5416) according to AUXF/Plclf/ChIndep/AddrChIndepFloat	Bit 0 (acknowledgement) Bit 1 : Bit 31 Bit 32 (float value) : Bit 63	S

Tab. 13-6: Float-coded auxiliary function S

14 Graphical user interface (GUI) - Basic parameters -

14.1 General information

The data group *Graphical User Interface (GUI)* contains the following settings for display and operation:

- [Measuring unit in the display, page 239](#)
- [Channel display, page 239](#)
- [Position display, page 241](#)
- [Technology display for feed and speed, page 253](#)
- [Modal NC function group display, page 256](#)
- [Modal M-function group display, page 257](#)
- [Display of zero offsets, page 257](#)
- [Display of placements, page 258](#)
- [Table display, page 258](#)

14.2 Measuring unit for display (DispUnit)

Measuring unit for display (DispUnit)	DispUnit

Specifies the "unit" used for positions in the display (mm or inches).

This setting also affects derived quantities.

Path	GUI/DispUnit	
Value range	0	automatically with G70/G71
	1	mm
	2	inch
Pre-assignment	0	automatically with G70/G71
Unit	-	
Description of parameter values	0	Unit selected automatically according to G70/G71
	1	Unit "mm"
	2	Unit "inches"
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6020 00030	
Data type	Byte0_3_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

14.3 Channel display (ChDisp)

14.3.1 General information

The "channel display" is used to specify which channel is active in the display after startup. This is where the axes/spatial coordinates that have their position defined under "Position display" (see [chapter 14.4 "Position display \(PosDisp\)" on page 241](#)) are displayed.

14.3.2 Active channel displayed after switch-on

Active channel displayed after switch-on		ActCh
Specifies which (machining) channel is selected as "active" after the GUI has been started.		
Path	GUI/ChDisp/ActCh	
Value range	0 .. 60	
Pre-assignment	1	
Unit	-	
Description of parameter values	0	After the user interface has been started, the latest active channel is displayed in the display window.
	1 .. 60	After the user interface has been started, the specified channel is displayed in the display window.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6001 00010	
Data type	Ch_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

14.3.3 Hide channel within the interface

Hide channel in the interface		SupprChDisp
Specifies separately for each applied channel Ch[k] whether it shall be hidden in the GUI.		
Hidden channels can neither be displayed in the channel dialog nor selected by changing the channel.		
Path	GUI/ChDisp/Suppr/Ch[k]/SupprChDisp	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No Channel display of channel Ch[k] exists.
	1	Yes Channel display of channel Ch[k] does not exist.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00010	
Data type	Byte0_1_t (byte)	

1 byte integer, unsigned for MACODA compatibility

14.4 Position display (PosDisp)

14.4.1 General information

In the MTX user interface, up to 8 axes are shown at the same time in the "Channel display" by default.

The axis display shows the current positions (position display) of the following elements:

- **Axes** (synchronous axes of the channel and auxiliary axes)
- **Spatial coordinates** (axes linked with a transformation) in so-called channel coordinate packets

When axis transformation is active, the "spatial coordinates" will be displayed instead of the axes linked with a transformation, since that is what is generally programmed. The "axis transformation type" TrafoType (in the folder TRA/AxTrafo[1..20]/) defines the axes that are linked to the maximum possible number of spatial coordinates (X, Y, z-coordinates and 3 orientation coordinates).

The names of the linear coordinates (spatial coordinates) are entered channel-specifically in LinCoord[1-3] (in the folder CHAN/Ch[k]/Coord/Wcs/). All active spatial coordinates of the channel form the channel coordinate packet (CCP).

The following settings are possible for the position display of axes Dr[i] and channel coordinate packets ChCoordPack[1..60]:

- "Coordinate system selection" for each channel Ch[k] (see coordinate system selection)
- "Axis order" for each axis Dr[i] and/or channel coordinate packet ChCoordPack[1..60] in each channel Ch[k] (see axis order)
- "Display precision" for each axis Dr[i] (see display precision)



Axes Dr[i] that run as spindles (i.e. no C-axis operation) **cannot** be configured here (for position display)!

14.4.2 Updating actual values in workpiece coordinates

Updating actual workpiece coordinate values	UpdateTimePosDisp

Specifies whether or not, and if so, after how many interpolator cycles the controller computes actual workpiece coordinate values to be displayed when kinematics axis transformation is enabled.

When kinematics axis transformations and/or coordinate transformations from the current actual axis values are enabled, the MTX can generate, on the basis of the current actual axis values, the display of the actual values with reference to the machine coordinate system MCS (Cartesian actual values).

The NC computes these actual values by an interrupt in the course of the interpolation process. New values are therefore available after the following interval at the earliest:

Interval = UpdateTimePosDisp * ScsCycTime

(ScsCycTime is located in folder SCS/ScsIf[1])

Path GUI/PosDisp/UpdateTimePosDisp

Graphical user interface (GUI) - Basic parameters -

Value range	0 .. 1000	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	No computation
	1 .. 1000	Number of interpolator cycles
 This parameter has no impact on the GUI display cycle time.		
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	9030 00002	
Data type	Int0_1_t (unsignedInt)	
	4 byte integer, signed for MACODA compatibility	

14.4.3 Coordinate system selection

World coordinate system (WCS)

Workpiece coordinate system (WCS)		WcsPosDisp
Specifies for channel Ch[k] whether the position display shall be based on the workpiece coordinate system (WCS).		
Path	GUI/PosDisp/CoordSys/Ch[k]/WcsPosDisp	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No Workpiece coordinate system not displayable
	1	Yes Workpiece coordinate system displayed
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00100	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Machine coordinate system (MCS)

Machine coordinate system (MCS)	McsPosDisp
Specifies for channel Ch[k] whether the position display shall be based on the machine coordinate system (MCS).	

Path	GUI/PosDisp/CoordSys/Ch[k]/McsPosDisp	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Machine coordinate system not displayable
	1	Yes
Access rights		Machine coordinate system displayed
	MTB:	Write
	Setter:	Read
Activation	Operator:	Read
	System restart (CS)	
Dependency	-	
Macoda ID	6005 00100	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Basic coordinate system (BCS)

Basic coordinate system (BCS)		BcsPosDisp
Specifies for channel Ch[k] whether the position display shall be based on the basic coordinate system (BCS).		
Path	GUI/PosDisp/CoordSys/Ch[k]/BcsPosDisp	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Basic coordinate system not displayable
	1	Yes
Access rights		Basic coordinate system displayed
	MTB:	Write
	Setter:	Read
Activation	Operator:	Read
	System restart (CS)	
Dependency	-	
Macoda ID	6005 00100	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Axis coordinate system (ACS)

Axis coordinate system (ACS)	AcsPosDisp
Specifies for channel Ch[k] whether the position display shall be based on the axis coordinate system (ACS).	

Graphical user interface (GUI) - Basic parameters -

Path	GUI/PosDisp/CoordSys/Ch[k]/AcsPosDisp	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No
		Axis coordinate system not displayable
	1	Yes
Access rights		Axis coordinate system displayed
	MTB:	Write
	Setter:	Read
Activation	Operator:	Read
	System restart (CS)	
	-	
Dependency	-	
Macoda ID	6005 00100	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

14.4.4 Display order of axes and channel coordinate packages

General information

- Display order** *Conditions for sequence of axes and channel coordinate packets (CCPs) in the axis display:*
1. **With the axis transformation inactive**, the **axes of the current channel** are displayed first. The "order of the axes in the channel window" according to PrioAxChWin (in the folder GUI/PosDisp/Ord/Ch[k]/ChWin/Dr[i]/) applies.
 2. **With active axis transformation**, the **channel coordinate package of the current channel** (ChCoordPack[k]) is displayed **first**. Below this are **further axes of the current channel** that are not involved in the axis transformation but are selected in "order of axes in the channel window" PrioAxChWin (in the folder GUI/PosDisp/Ord/Ch[k]/ChWin/Dr[i]/).
- The display position of the CCP can be shifted into the area of the entire sequence with "Display channel coord. packet in additional window" ModeChPack (in the folder GUI/PosDisp/Ord/Ch[k]/AddWin/ChCoordPack[1..60]/), but the order of spatial coordinates within the CCP cannot be changed.
3. Finally, there are the **elements of the entire sequence** (these are the remaining axes of the channel, asynchronous axes, axes of other channels and CCPs of other channels):
 - **Axes:**
"Display axis in the auxiliary window" ModeAddAx
(in folder GUI/PosDisp/Ord/Ch[k]/AddWin/AddAx/Dr[i]/),
 - **CCPs:**
"channel coordinate package in additional window" ModeChPack
(in Ordner GUI/PosDisp/Ord/Ch[k]/AddWin/ChCoordPack[1..60]/)

Note: *If the same priorities are assigned to different axes/CCPs in the parameters concerned, the following applies:*

- The order of the axes with the same priority results form the specified physical axis index
- The order of the CCPs with the same priority results form the channel index
- CCPs from ModeChPack ("Display channel coordinate in additional window") are displayed before the axes from ModeAddAx ("Display axis in additional window") are displayed

Default setting: The default settings can be seen in the following figure:

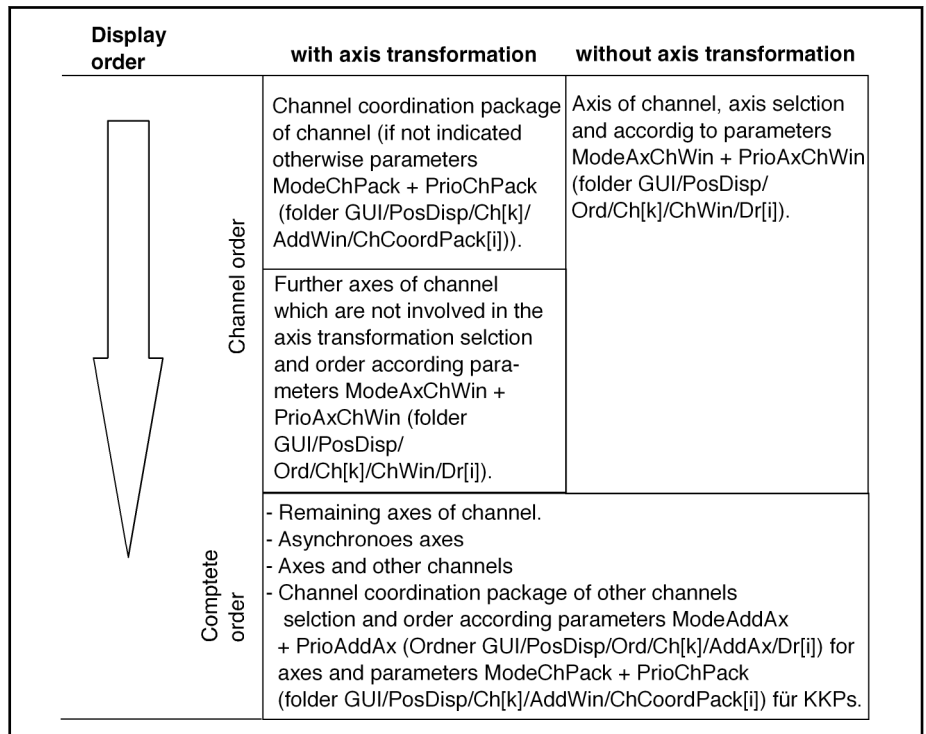


Fig. 14-1: Default setting of position display

Examples The following figures show a configuration of a possible display order for the displaying of channel 1.

The number of the axes (or coordinates) to be displayed is limited to 8 (by default).

Example:

Axes in the channel display

Axes of channel 1 (channel sequence)	Other axes and CCPs (total sequence)
Axis 1	Axis 4
Axis 2	CCP of channel 3
Axis 3	Axis 13 (is to be displayed always)
Axis 10	Axis 14
Axis 8	

Tab. 14-1: Axes in the position display

Graphical user interface (GUI) - Basic parameters -

To display the axes of channel 1 in the desired order in channel 1, the "Order of the axis in the channel window" ModeAxChWin (in folder GUI/PosDisp/Ord/Ch[1]/ChWin/Dr[i]/) is to be parameterized as described below:

Axis	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15...
PrioAxCh-Win	1	1	1	0	0	0	0	2	0	1	0	0	0	0	0
Mod-eAxChWin	Yes	Yes	Yes	No	No	No	No	Yes	No	Yes	No	No	No	No	No

Tab. 14-2: Order of axes in the channel window

To display the channel coordinate package of channel 3 in the desired location in channel 1, the "Display channel coordinate package in the additional window" ModeAxChWin (in folder GUI/PosDisp/Ord/Ch[1]/AddWin/ChCoord-Pack[1..60]/) is to be parameterized as described below:

Channel	1	2	3	4	5	6	7	8	9	10	11	12...	
PrioCh-Pack	0	0	2	0	0	0	0	0	0	0	0	0	
ModeCh-Pack	No	No	Yes	No	No	No	No	No	No	No	No	No	

Tab. 14-3: Display channel coordinate packet in additional window

To display the other axes in the desired location in channel 1, the "Display axis in additional window" ModeAddAx (in folder GUI/PosDisp/Ord/Ch[k]/AddWin/AddAx/Dr[i]/) is to be parameterized as described below:

Axis	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15...
PrioAddAx	0	0	0	1	0	0	0	0	0	0	0	0	3	4	0
ModeAddAx	No	No	No	if axis is in channel	No	No	No	No	No	No	No	No	Al-ways	if axis is in channel	No

Tab. 14-4: Order of the axes in the auxiliary window

Example:

Display without axis transformation

Display without axis transformation, if

- axes 1, 2, 3 and 8 are assigned to channel 1,
- axis 13 is assigned to channel 2,
- the axes 4 and 14 are asynchronous,
- not transformation is active in the channels 1 and 2.

Display sequence	Defined by
Synchronous axis 1	Channel sequence
Synchronous axis 2	Channel sequence
Synchronous axis 3	Channel sequence
Synchronous axis 8	Channel sequence
Asynchronous axis 4	Entire sequence

Display sequence	Defined by
Synchronous axis 13	Entire sequence
Asynchronous axis 14	Entire sequence

Tab. 14-5: Display order without axis transformation

Since it does not belong to channel 1, axis 10 is not displayed. In addition, there are no statements as to whether it belongs to another channel or has an asynchronous property.

Example:

Display with axis transformation

Display with axis transformation, if

- axes 1, 2, 3, 4 and 8 are assigned to channel 1,
- a transformation is active in channel 1, with coordinates x, y, z, psi and axes 1, 2, 4 and 8,
- a transformation is active in channel 3, with coordinates x3 and y3
- axis 13 is assigned to channel 2,
- axis 14 is asynchronous.

Display sequence	Defined by
x	CCP of channel 1
y	
z	
psi	
Synchronous axis 3	Channel sequence
x3	CCP of channel 3 (entire sequence)
y3	
Synchronous axis 13	Entire sequence

Tab. 14-6: Display order with axis transformation

The asynchronous axis 14 is not displayed anymore if the display is limited to 8 axes (by default).

Maximum number of axes in the axis display

Maximum number of axes in axis display	NofAxPosDisp

Specifies the maximum number of axes that shall appear in the axis display of a channel Ch[k].

The parameter value defines the number of axes according to the display sequence defined in the following machine parameters:

- "Display axis in the auxiliary window" ModeAddAx
(in folder GUI/PosDisp/Ord/Ch[k]/AddWin/AddAx/Dr[i]/)
- "Display channel coordinate packet in additional window" ModeChPack
(in the GUI/PosDisp/Ord/Ch[k]/AddWin/ChCoordPack[1..60]/) folder and
- "Axis sequence in the Channel Window" ModeAxChWin
(in Ordner GUI/PosDisp/Ord/Ch[k]/ChWin/Dr[i]/)

Graphical user interface (GUI) - Basic parameters -

	Up to 25 axis values can be displayed in the channel. The default setting "8" ensures compatibility with the default display.	
Path	GUI/PosDisp/Ord/Ch[k]/NofAxPosDisp	
Value range	0 .. 25	
Pre-assignment	8	
Unit	-	
Description of parameter values	Number of axes to be displayed.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00023	
Data type	NofAxPosDisp_Dr_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Displaying the axis in the channel window

Display axis in channel window		ModeAxChWin
Specifies for each channel Ch[k] for each applied axis Dr[i] whether it shall be displayed within the axis display for its channel.		
The display is in the upper part of the axis position display of the current channel.		
Path	GUI/PosDisp/Ord/Ch[k]/ChWin/Dr[i]/ModeAxChWin	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No
		Do not display axis in channel window.
	1	Yes
Access rights		Display axis in channel window.
	MTB:	Write
	Setter:	Read
Activation	Operator:	Read
	System restart (CS)	
Dependency	-	
Macoda ID	6005 00022	
Data type	Byte0_1_t (byte)	
	4 byte integer for MACODA compatibility	

Order of the axis in the channel window

Order of the axis in the channel window	PrioAxChWin
Specifies for each channel Ch[k] for each applied axis Dr[i] in what position it shall be displayed within the axis display for its channel (channel sequence).	

	The display is in the upper part of the axis position display of the current channel.	
Path	GUI/PosDisp/Ord/Ch[k]/ChWin/Dr[i]/PrioAxChWin	
Value range	0 .. 250	
Pre-assignment	1	
Unit	-	
Description of parameter values	"Position in the display order" of the axis in the channel window.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00022	
Data type	Dr_t (byte)	
	4 byte integer for MACODA compatibility	

Displaying the axis in the additional window

Display axis in the additional window		ModeAddAx
Specifies for each channel Ch[k] for each applied axis Dr[i] (auxiliary and machining axes) whether it shall be displayed in the entire sequence within the axis display for its channel.		
The display is in the lower part of the axis position display of the current channel.		
Path	GUI/PosDisp/Ord/Ch[k]/AddWin/AddAx/Dr[i]/ModeAddAx	
Value range	0	No
	1	if axis is in channel
	2	Always
Pre-assignment	1	if axis is in channel
Unit	-	
Description of parameter values	0	No
	1	Do not display axis in entire sequence in channel window.
	2	If the axis is a synchronous axis of the channel or an auxiliary axis, then it will be displayed in the channel window.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00020	
Data type	Byte0_2_t (byte)	
	4 byte integer for MACODA compatibility	

Order of the axis in the additional window

Order of the axis in the additional window		PrioAddAx
Specifies for each channel Ch[k] for each applied axis Dr[i] (auxiliary and machining axes) in what position it shall be displayed in the entire sequence within the axis display for its channel.		
The display is in the lower part of the axis position display of the current channel.		
Path	GUI/PosDisp/Ord/Ch[k]/AddWin/AddAx/Dr[i]/PrioAddAx	
Value range	0 .. 250	
Pre-assignment	1	
Unit	-	
Description of parameter values	Position number in the entire sequence within the axis display of the current channel window.	
	Axes with identical position number are displayed in the order of their physical axis index i of Dr[i].	
		Since the position numbers are interpreted as "priorities", there are no gaps in the display sequence. "The Order of axes in the channel window" PrioAxChWin (in folder GUI/PosDisp/Ord/Ch[k]/ChWin/Dr[i]/) has priority over "Display axis in additional window" ModeAddAx (in folder GUI/PosDisp/Ord/Ch[k]/AddWin/AddAx/Dr[i]/).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00020	
Data type	Dr_t (byte)	
	4 byte integer for MACODA compatibility	

Displaying channel coordination in the additional window

Display channel coord. package in the additional window		ModeChPack
Specifies channel-specifically Ch[k] whether the channel coordinate packet ChCoordPack[1..60] shall be displayed in the entire sequence.		
The display is in the lower part of the axis position display of the current channel.		
Path	GUI/PosDisp/Ord/Ch[k]/AddWin/ChCoordPack[1..60]/ModeChPack	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
	CCP not displayed in the additional window of the current channel. If the CCP refers to the current channel, it is still au-	

		tomatically displayed in the channel window of the current channel.
	1	Yes
		CCP is displayed in the additional window of the current channel. If the CCP refers to the current channel, it is not displayed in the channel window of the current channel.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00021	
Data type	Byte0_1_t (byte)	
	4 byte integer for MACODA compatibility	

Order of the channel coordination package in the additional window

Order of channel coord. package in the additional window		PrioChPack
	Specifies channel-specifically Ch[k] in which position in the axis display the channel coordinate packet ChCoordPack[1..60] of the channels shall be displayed in the entire sequence.	
	The display is in the lower part of the axis position display of the current channel.	
Path	GUI/PosDisp/Ord/Ch[k]/AddWin/ChCoordPack[1..60]/PrioChPack	
Value range	0 .. 250	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	The CCP (ChCoordPack[1..60]) is displayed at the first position in the currently active channel, but not above the entire sequence in the other channels.
	>0	Position number of the CCP (ChCoordPack[1..60]) in the entire sequence.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00021	
Data type	Dr_t (byte)	
	4 byte integer for MACODA compatibility	

14.4.5 Display precision

Number of decimal places for metric display

Number of decimal places for metric Display	PrecPosMetr
Specifies for each axis Dr[j] the number of decimal places for the position display when using the metric unit system.	

Graphical user interface (GUI) - Basic parameters -

The parameter always applies to rotary and endless axis. Their position value is always displayed in ° (degrees) and is indicated by the trailing character °.

The metric system is active for

- linear **machining axes**, as long as it is selected in the channel "Programming mode" (all operation modes except for "Jog" and "Move to reference point" modes) and metric programming is active (G71).
If "Jog" or "Move to reference point" mode is selected, the signal "Inch increments" (qAx_JogInch) must not be set on the corresponding axis interface
(qAx_JogInch) does not have to be set.
- linear **auxiliary axes**, if the axis is in the programming mode, i.e. the "Jog" and "Move to reference point" axis modes are not applied to the axis interface.
If these signals are applied, the "Inch increments" (qAx_JogInch) signal may not be set at the corresponding axis interface.

Path	GUI/PosDisp/Prec/Dr[i]/PrecPosMetr
Value range	0 .. 7
Pre-assignment	3
Unit	-
Description of parameter values	Number of decimal places for position display in the metric unit system.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	6020 00011
Data type	Prec_t (byte) 4 byte integer, unsigned for MACODA compatibility

Number of decimal places for non-metric display

Number of decimal places for non-metric Display	PrecPosInch
Specifies for each axis Dr[i] the number of decimal places for the position display when using the inches unit system.	
 Since they are always programmed in ° (degrees), rotary and endless axes are not affected by this parameter. Their number of decimal places is defined with "Decimal places" PrecPosMetr in the folder GUI/PosDisp/Prec/Dr[i]/.	

The inches system is active for

- linear **machining axes**, as long as it is selected in the channel "Programming mode" (all operation modes except for "Jog" and "Move to reference point" modes) and inch programming is active (G70).
If "Jog" or "Move to reference point" mode is selected, the signal "Inch increments" (qAx_JogInch) must be set on the corresponding axis interface.
- linear **auxiliary axes**, if the axis is in the setup mode, i.e. the "Jog" and "Move to reference point" axis modes are not applied to the axis interface and the signal "Inch increments" (qAx_JogInch) is set.



The measuring system is always "metric" for linear auxiliary axes, if the "programming mode" axis mode is active.

Path	GUI/PosDisp/Prec/Dr[i]/PrecPosInch	
Value range	0 .. 7	
Pre-assignment	4	
Unit	-	
Description of parameter values	Number of decimal places for position display in the inches unit system.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6020 00012	
Data type	Prec_t (byte)	
	4 byte integer, unsigned for MACODA compatibility	

14.5 Technology display for feed and speed

14.5.1 General information

The technology display refers to the feed and speed display for **spindles or spindle groups**. The settings for spindle or spindle group display are selected in the folder ../SpSpGrDisp[1..8]/...

*To do so, select the folder **GUI/TechDisp/Ch[k]/SpSpGrDisp[1..8]** and, with the right mouse button, select one of the entries from the context menu with the function "convert to..."*

- SpDisp
- SpGrDisp

Macoda ID 6005 00030

(<0 = Spindle group display, 0 = no display, >0 spindle display)

14.5.2 Feed display

Number of decimal places for metric display

Number of decimal places for metric Display		PrecFeedMetr
Specifies the number of decimal places shown for the channel feed display (for spindles or spindle groups) when G71 (metric programming) is active.		
Path	GUI/TechDisp/FeedDisp/PrecFeedMetr	
Value range	0 .. 7	
Pre-assignment	1	
Unit	-	
Description of parameter values	Number of decimal places displayed for the channel feed display in the "metric" unit system.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Graphical user interface (GUI) - Basic parameters -

Activation	System restart (CS)
Dependency	-
Macoda ID	6020 00001
Data type	Prec_t (byte) 1 byte integer, unsigned for MACODA compatibility

Number of decimal places for non-metric display

Number of decimal places for non-metric Display		PrecFeedInch
Specifies the number of decimal places shown for the channel feed display (for spindles or spindle groups) when G70 (inch programming) is active.		
Path	GUI/TechDisp/FeedDisp/PrecFeedInch	
Value range	0 .. 7	
Pre-assignment	2	
Unit	-	
Description of parameter values	Number of decimal places displayed for the channel feed display in "inches".	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6020 00002	
Data type	Prec_t (byte) 1 byte integer, unsigned for MACODA compatibility	

14.5.3 Speed display**Number of decimal position for spindle speed**

Number of decimal position for spindle speed		PrecSpSpeed
Specifies the number of decimal position for spindle speeds.		
Path	GUI/TechDisp/SpeedDisp/PrecSpSpeed	
Value range	0 .. 7	
Pre-assignment	1	
Unit	-	
Description of parameter values	Number of decimal position for spindle speeds.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6020 00005	
Data type	Prec_t (byte) 1 byte integer, unsigned for MACODA compatibility	

Number of the spindle to be displayed

Number of the spindle to be displayed		SpIndSpDisp
Specifies channel-specifically Ch[k] which applied spindle (see "Assigned system spindles" SP/SpAdmin/ChSp/ChSpGrAlloc/Ch[k]/ChSp[cs]/SpIndChSp) shall be displayed in which position within the spindle/spindle group display SpSpGrDisp[1..8].		
In this way, the spindle/spindle group display allows visualization of up to 8 spindles in the corresponding display section.		
Path	GUI/TechDisp/Ch[k]/SpSpGrDisp[sp]/SpDisp/SpIndSpDisp	
Value range	0 .. 60	
Pre-assignment	1	
Unit	-	
Description of parameter values	0	The spindle/spindle group display position is not in use.
	>0	Number of system spindle to be displayed in the spindle/spindle group display SpSpGrDisp[1..8].
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00030	
	(>0 = spindle display, 0 = no display, <0 spindle group display)	
Data type	Sp_t (byte)	
	4 byte integer for MACODA compatibility	

Number of the spindle group to be displayed

Number of the spindle group to be displayed		SpGrIndSpGrDisp
Specifies channel-specifically Ch[k] which applied spindle group (see "Assigned spindle group" SP/SpAdmin/Sp/SpGrAlloc/Sp[1..60]/SpGrInd) shall be displayed in which position within the spindle/spindle group display SpSpGrDisp[1..8].		
In this way, the spindle/spindle group display allows visualization of up to 8 spindles in the corresponding display section.		
Path	GUI/TechDisp/Ch[k]/SpSpGrDisp[sp]/SpGrDisp/SpGrIndSpGrDisp	
Value range	0 .. 4	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	The spindle/spindle group display position is not in use.
	>0	Number of system spindle group to be displayed in the spindle/spindle group display SpSpGrDisp[1..8].
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	

Macoda ID 6005 00030
 (<0 = Spindle group display, 0 = no display, >0 spindle display)
Data type SpGr_t (byte)
 4 byte integer for MACODA compatibility

14.6 Modal NC function group display (NcCodeDisp)

14.6.1 NC function for group[1..32]

	NC function for group[1..32]	NcCodeGr[1..32]
	Specifies channel-specifically Ch[k] which modal NC functions NcCodeGr[1..32] are displayed in the NC function group display.	
Path	GUI/NcCodeDisp/Ch[k]/NcCodeGr[1..32]	
Value range	{NC syntax of an NC function}	
Pre-assignment	G80	.../NcCodeGr[1]
	G1	.../NcCodeGr[2]
	G17	.../NcCodeGr[3]
	G90	.../NcCodeGr[4]
	G94	.../NcCodeGr[5]
	G8	.../NcCodeGr[6]
	G62	.../NcCodeGr[7]
	G48	.../NcCodeGr[8]
	G40	.../NcCodeGr[9]
	G43	.../NcCodeGr[10]
	G45	.../NcCodeGr[11]
	G71	.../NcCodeGr[12]
	AAC	.../NcCodeGr[13]
	PAC	.../NcCodeGr[14]
	OVE	.../NcCodeGr[15]
	-	.../NcCodeGr[16-32]
Unit	-	
Description of parameter values	By specifying an NC function, all modal NC functions belonging to the same group will be shown in the NC function group display.	
	 All unassigned parameters in NcCodeGr[1..32] have to be empty .	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00040	
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility	

14.7 Modal M-function group display (AuxFuncDisp)

14.7.1 M-function for group[1..32]

	M-function for group[1..32]	AuxCodeGr[1..32]
	Specifies channel-specifically Ch[k] which modal auxiliary functions AuxCodeGr[1..32] shall be displayed in the M-function group display.	
Path	GUI/AuxFuncDisp/Ch[k]/AuxCodeGr[1..32]	
Value range	{NC-Syntax for an auxiliary function}	
Pre-assignment	M3	AuxCodeGr[1]
	-	AuxCodeGr[2-32]
Unit	-	
Description of parameter values	By specifying an auxiliary function, all modal auxiliary functions belonging to the same group will be shown in the M-function group display.	
	 All unassigned parameters in AuxCodeGr[1..32] have to be empty .	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00050	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

14.8 Display of zero point offsets (ZotDisp)

14.8.1 Number of zero point offset banks to be displayed

	Number of zero point offset banks to be displayed	NofZotSets
	Specifies the number of zero point offset tables NofZotSets to be displayed on the MTX GUI for each channel Ch[k].	
Path	GUI/ZotDisp/Ch[k]/NofZotSets	
Value range	0 .. 5	
Pre-assignment	5	
Unit	-	
Description of parameter values	Number of zero point offset banks to be displayed.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00061	
Data type	Byte0_5_t (byte)	

1 byte integer, unsigned for MACODA compatibility

14.9 Number of placements (PmtDisp)

14.9.1 Number of placement banks to be displayed

Number of placement banks to be displayed		NofPmtSets
Specifies the number of placements banks NofPmtSets to be displayed on the MTX GUI for each channel Ch[k].		
Path	GUI/PmtDisp/Ch[k]/NofPmtSets	
Value range	0 .. 5	
Pre-assignment	5	
Unit	-	
Description of parameter values	Number of placement banks to be displayed.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6005 00071	
Data type	Byte0_5_t (byte) 1 byte integer, unsigned for MACODA compatibility	

14.10 Table display (TabDisp)

14.10.1 Decimal position for linear axes for metric table display

Decimal position for linear axes for metric displaying of tables		PrecLinMetr
Specifies the number of decimal places that can be entered and will be displayed in the zero point offset and geometry correction tables for linear axes (editor with metric input).		
Path	GUI/TabDisp/PrecLinMetr	
Value range	0 .. 7	
Pre-assignment	3	
Unit	-	
Description of parameter values	Number of decimal places that can be entered and will be displayed in the zero point offset and geometry correction tables for linear axes	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6020 00021	
Data type	Prec_t (byte) 1 byte integer, unsigned for MACODA compatibility	

14.10.2 Decimal position for rotary axes for metric table display

Decimal position for rotary axes for metric displaying of tables		PrecRotMetr
	Specifies the number of decimal places that can be entered and will be displayed in the zero point offset table for rotary axes (editor with metric input).	
Path	GUI/TabDisp/PrecRotMetr	
Value range	0 .. 7	
Pre-assignment	3	
Unit	-	
Description of parameter values	Number of decimal places that can be entered and will be displayed in the zero point offset table for rotary axes.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6020 00021	
Data type	Prec_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

14.10.3 Decimal position for linear axes for non-metric table display

Decimal position for linear axes for non-metric displaying of tables		PrecLinInch
	Specifies the number of decimal places that can be entered and will be displayed in the zero point offset and geometry correction tables for linear axes (editor with inch input).	
Path	GUI/TabDisp/PrecLinInch	
Value range	0 .. 7	
Pre-assignment	4	
Unit	-	
Description of parameter values	Number of decimal places that can be entered and will be displayed in the zero point offset and geometry correction tables for linear axes.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6020 00022	
Data type	Prec_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

14.10.4 Decimal positions for rotary axes for non-metric table display

Decimal positions for rotary axes for non-metric displaying of tables	PrecRotInch

Graphical user interface (GUI) - Basic parameters -

	Specifies the number of decimal places that can be entered and will be displayed in the zero point offset table for rotary axes (editor with inch input).	
Path	GUI/TabDisp/PrecRotInch	
Value range	0 .. 7	
Pre-assignment	4	
Unit	-	
Description of parameter values	Number of decimal places that can be entered and will be displayed in zero point offset tables for rotary axes.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	6020 00022	
Data type	Prec_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

15 NC programming (NCP) - Basic parameters

15.1 General information

The data group NC programming (NCP) covers the following settings:

- [Subroutine technique, page 261](#)
- [Polar coordinate programming, page 273](#)
- [Corner rounding, page 274](#)
- [Diameter programming, page 275](#)
- [Tool path correction, page 277](#)

15.2 Subroutine technology (SubProg)

15.2.1 General information

Subroutine technology is divided into

- Non-modal G functions
(Non-modal G-functions GFunc[1..80])
- Non-modal M-functions
(Non-modal M-functions MFunc[1...8])
- Modal subroutines
(Modal subroutines SwiOn[1..63] / SwiOff).

There are 80 "non-modal G-functions" GFunc[1..80] and a total of 8 "non-modal M-functions" MFunc[1..8]. For the "modal subroutines", there are 63 switch on functions SwiOn[1..63] and one switch off function SwiOff.

15.2.2 Non-modal G-functions GFunc[1..80]

G-code number

G-code number	GFunc[1..80]/GFuncCode
Defines up to 80 G-functions GFunc[1..80]/GFuncCode, with which non-modal subroutines can be called up.	
If the control detects a G-function defined in the currently active NC block, the control starts the program that is entered in the corresponding individual parameter of "GFunc[1..80]/GFuncSubProg" g. This allows to include own G-functions in the NC syntax. "Non-modal" means in this context that the respective subroutine is called once and only once.	
Path	NCP/SubProg/NonModalGFunc/GFunc[1..80]/GFuncCode
Value range	0.. 10000 0.. 9999 for MACODA compatibility
Pre-assignment	The entries are preassigned with G-code numbers of the MTX standard cycles: 111 .../GFunc[1]/GFuncCode 112 .../GFunc[2]/GFuncCode 113 .../GFunc[3]/GFuncCode 114 .../GFunc[4]/GFuncCode 115 .../GFunc[5]/GFuncCode 116 .../GFunc[6]/GFuncCode

NC programming (NCP) - Basic parameters

160	.../GFunc[7]/GFuncCode
161	.../GFunc[8]/GFuncCode
162	.../GFunc[9]/GFuncCode
166	.../GFunc[10]/GFuncCode
167	.../GFunc[11]/GFuncCode
168	.../GFunc[12]/GFuncCode
169	.../GFunc[13]/GFuncCode
170	.../GFunc[14]/GFuncCode
171	.../GFunc[15]/GFuncCode
172	.../GFunc[16]/GFuncCode
173	.../GFunc[17]/GFuncCode
174	.../GFunc[18]/GFuncCode
191	.../GFunc[19]/GFuncCode
192	.../GFunc[20]/GFuncCode
196	.../GFunc[21]/GFuncCode
197	.../GFunc[22]/GFuncCode
201	.../GFunc[23]/GFuncCode
202	.../GFunc[24]/GFuncCode
213	.../GFunc[25]/GFuncCode
214	.../GFunc[26]/GFuncCode
215	.../GFunc[27]/GFuncCode
219	.../GFunc[28]/GFuncCode
224	.../GFunc[29]/GFuncCode
226	.../GFunc[30]/GFuncCode
230	.../GFunc[31]/GFuncCode
301	.../GFunc[32]/GFuncCode
303	.../GFunc[33]/GFuncCode
311	.../GFunc[34]/GFuncCode
315	.../GFunc[35]/GFuncCode
317	.../GFunc[36]/GFuncCode
318	.../GFunc[37]/GFuncCode
319	.../GFunc[38]/GFuncCode
321	.../GFunc[39]/GFuncCode
323	.../GFunc[40]/GFuncCode
324	.../GFunc[41]/GFuncCode
327	.../GFunc[42]/GFuncCode
328	.../GFunc[43]/GFuncCode
329	.../GFunc[44]/GFuncCode
331	.../GFunc[45]/GFuncCode
333	.../GFunc[46]/GFuncCode
501	.../GFunc[47]/GFuncCode
511	.../GFunc[48]/GFuncCode
541	.../GFunc[49]/GFuncCode

	542	.../GFunc[50]/GFuncCode														
	543	.../GFunc[51]/GFuncCode														
	591 *)	.../GFunc[52]/GFuncCode														
	592	.../GFunc[53]/GFuncCode														
	628	.../GFunc[54]/GFuncCode														
	629	.../GFunc[55]/GFuncCode														
	0	.../GFunc[56..80]/GFuncCode														
	*) The cycle G591 is not supported anymore from MTX 14VRS. The cycle G592 is replaced.															
Unit	-															
Description of parameter values	0	No G-function defined														
	>0	G-code number of G-function														
Access rights	MTB:	Write														
	Setter:	Read														
	Operator:	Read														
Activation	System restart (CS)															
Dependency	-															
Macoda ID	3090 00001															
Data type	Int0_10T_t (int)															
	4 byte integer, unsigned for MACODA compatibility															
	<i>Example:</i>															
	2 non-modal G-functions															
	Specification:															
	.../GFunc[1]/GFuncCode: 111															
	.../GFunc[1]/GFuncSubProg: STC_G111															
	Result:															
	Programming "G111" starts the standard cycle															
	"STC_G111" (Hole circle rotary axis).															
	Specification:															
	.../GFunc[52]/GFuncCode: 1234															
	.../GFunc[52]/GFuncSubProg: Prog1234															
	Result:															
	Programming "G1234" starts the subroutine "Prog1234".															
Subroutine name																
	<table><tr><th>Subroutine name</th><th>GFuncSubProg</th></tr><tr><td colspan="2">Defines up to 80 names GFunc[1..80]/GFuncSubProg of non-modal subroutines which are called up in dependency of GFunc[1..80]/GFuncCode.</td></tr><tr><td colspan="2">NCP/SubProg/NonModalGFunc/GFunc[1..80]/GFuncSubProg</td></tr><tr><td colspan="2">{Text}</td></tr><tr><td colspan="2">The entries are preassigned with names of the MTX standard cycles:</td></tr><tr><td>STC_G111</td><td>.../GFunc[1]/GFuncSubProg</td></tr><tr><td></td><td>Hole circle, rotary axis</td></tr></table>		Subroutine name	GFuncSubProg	Defines up to 80 names GFunc[1..80]/GFuncSubProg of non-modal subroutines which are called up in dependency of GFunc[1..80]/GFuncCode.		NCP/SubProg/NonModalGFunc/GFunc[1..80]/GFuncSubProg		{Text}		The entries are preassigned with names of the MTX standard cycles:		STC_G111	.../GFunc[1]/GFuncSubProg		Hole circle, rotary axis
Subroutine name	GFuncSubProg															
Defines up to 80 names GFunc[1..80]/GFuncSubProg of non-modal subroutines which are called up in dependency of GFunc[1..80]/GFuncCode.																
NCP/SubProg/NonModalGFunc/GFunc[1..80]/GFuncSubProg																
{Text}																
The entries are preassigned with names of the MTX standard cycles:																
STC_G111	.../GFunc[1]/GFuncSubProg															
	Hole circle, rotary axis															
Path	NCP/SubProg/NonModalGFunc/GFunc[1..80]/GFuncSubProg															
Value range	{Text}															
Pre-assignment	The entries are preassigned with names of the MTX standard cycles:															
	STC_G111	.../GFunc[1]/GFuncSubProg														
		Hole circle, rotary axis														

NC programming (NCP) - Basic parameters

STC_G112	.../GFunc[2]/GFuncSubProg Hole circle, main spindle
STC_G113	.../GFunc[3]/GFuncSubProg Hole circle, plane
STC_G114	.../GFunc[4]/GFuncSubProg Hole pattern, line
STC_G115	.../GFunc[5]/GFuncSubProg Hole pattern, matrix
STC_G116	.../GFunc[6]/GFuncSubProg Hole pattern, frame
STC_G160	.../GFunc[7]/GFuncSubProg Longitudinal turning
STC_G161	.../GFunc[8]/GFuncSubProg Face turning
STC_G162	.../GFunc[9]/GFuncSubProg Tapered turning
STC_G166	.../GFunc[10]/GFuncSubProg Thread sequence
STC_G167	.../GFunc[11]/GFuncSubProg Thread turning
STC_G168	.../GFunc[12]/GFuncSubProg Face thread
STC_G169	.../GFunc[13]/GFuncSubProg Tapered thread
STC_G170	.../GFunc[14]/GFuncSubProg Contour finishing
STC_G171	.../GFunc[15]/GFuncSubProg Contour roughing, longitudinal
STC_G172	.../GFunc[16]/GFuncSubProg Contour roughing, plane
STC_G173	.../GFunc[17]/GFuncSubProg Contour roughing, parallel to contour
STC_G174	.../GFunc[18]/GFuncSubProg Contour plunge turning
STC_G191	.../GFunc[19]/GFuncSubProg Groove Turning
STC_G192	.../GFunc[20]/GFuncSubProg Groove and turning
STC_G196	.../GFunc[21]/GFuncSubProg Thread undercut according to DIN 76 or free form
STC_G197	.../GFunc[22]/GFuncSubProg Thread relief groove according to DIN 509 or free form
STC_G201	.../GFunc[23]/GFuncSubProg

	Linear text engraving
STC_G202	.../GFunc[24]/GFuncSubProg
	Circular text engraving
STC_G213	.../GFunc[25]/GFuncSubProg
	Hole circle punching
STC_G214	.../GFunc[26]/GFuncSubProg
	Hole pattern line punching
STC_G215	.../GFunc[27]/GFuncSubProg
	Hole pattern matrix punching
STC_G219	.../GFunc[28]/GFuncSubProg
	Multiple part punching
STC_G224	.../GFunc[29]/GFuncSubProg
	Nibble line
STC_G226	.../GFunc[30]/GFuncSubProg
	Nibble rectangular
STC_G230	.../GFunc[31]/GFuncSubProg
	Swiveling
STP_G301	.../GFunc[32]/GFuncSubProg
	Probe tool length
STP_G303	.../GFunc[33]/GFuncSubProg
	Probe tool radius
STP_G311	.../GFunc[34]/GFuncSubProg
	Calibrate probe - Tool length
STP_G315	.../GFunc[35]/GFuncSubProg
	Calibrate probe - Tool radius
STP_G317	.../GFunc[36]/GFuncSubProg
	Calibrate probe - Point
STP_G318	.../GFunc[37]/GFuncSubProg
	Calibrating the probe - Adjustment ring
STP_G319	.../GFunc[38]/GFuncSubProg
	Calibrate probe - Spigot
STP_G321	.../GFunc[39]/GFuncSubProg
	Probe point
STP_G323	.../GFunc[40]/GFuncSubProg
	Check workpiece availability
STP_G324	.../GFunc[41]/GFuncSubProg
	Probe boring
STP_G327	.../GFunc[42]/GFuncSubProg
	Probe spigot
STP_G328	.../GFunc[43]/GFuncSubProg
	Ball head measuring
STP_G329	.../GFunc[44]/GFuncSubProg
	Probe land

NC programming (NCP) - Basic parameters

STP_G331	.../GFunc[45]/GFuncSubProg Probe groove
STP_G333	.../GFunc[46]/GFuncSubProg Probe angle
STC_G501	.../GFunc[47]/GFuncSubProg T,F,S input dialog
STC_G511	.../GFunc[48]/GFuncSubProg Tool change, milling cycle
STP_G541	.../GFunc[49]/GFuncSubProg Tolerance monitoring
STP_G542	.../GFunc[50]/GFuncSubProg Average
STP_G543	.../GFunc[51]/GFuncSubProg Percentage compensation
STC_G591 *)	.../GFunc[52]/GFuncSubProg High-speed milling
STC_G592	.../GFunc[53]/GFuncSubProg High-speed milling (with new spline variant for an improved free form surface milling)
STP_G628	.../GFunc[54]/GFuncSubProg Computing plane orientation
STP_G629	.../GFunc[55]/GFuncSubProg Ball head measuring in space
<Blank>	.../GFunc[56..80]/GFuncSubProg

*) The cycle G591 is not supported anymore from MTX 14VRS. The cycle G592 is replaced.

Unit

-

Description of parameter values Subroutine name of the G-function

Access rights

MTB: Write

Setter: Read

Operator: Read

Activation System restart (CS)**Dependency** -**Macoda ID** 3090 00002

Data type Str8_t (isoLatin1String)

8 byte string for MACODA compatibility

15.2.3 Non-modal M-functions MFunc[1...8]

M-code number

M-code number	MFuncCode

Defines up to 8 M-functions MFunc[1..8]/MFuncCode, over which non-modal subroutines can be called up.

If it finds a G function defined in the NC block currently active, the control starts the program that is entered in the corresponding individual parameter

of MFunc[1..8]/MFuncSubProg. This allows to include own "M-functions" in the NC syntax. "Non-modal" means in this context that the respective subroutine is called once and only once.



It is not possible to assign an M-function to a subroutine if it has a spindle function assigned to it:

- Spindle syntax:
Parameters of the folder SP/SpFunc/SpSyn/Syn[1..8]/...
- Spindle group syntax:
Parameters of the folder SP/SpGrFunc/SpGrSyn[1..4]/...

Path	NCP/SubProg/NonModalMFunc/MFunc[1..8]/MFuncCode	
Value range	0.. 10000 0.. 9999 for MACODA compatibility	
Pre-assignment	0	.../MFunc[1..8]/MFuncCode
Unit	-	
Description of parameter values	0	No M-function defined
	>0	M-code number of M-function
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3090 00003	
Data type	Int0_10T_t (int)	
	4 byte integer, unsigned for MACODA compatibility	

Example:

Non-modal M-function

Specification:

.../MFunc[1]/MFuncCode: 6

.../MFunc[1]/MFuncSubProg: UpVonM6

Result:

Programming "M6" starts the subroutine "UpVonM6".

Subroutine name

Subroutine name		MFuncSubProg
Defines up to 8 names MFunc[1..8]/MFuncSubProg of subroutines, which are called up in dependence of MFunc[1..8]/MFuncCode.		
Path	NCP/SubProg/NonModalMFunc/MFunc[1..8]/MFuncSubProg	
Value range	{Text}	
Pre-assignment	UpVonM6	.../MFunc[1]/MFuncCode
	<Space>	.../MFunc[2..8]/MFuncCode
Unit	-	

NC programming (NCP) - Basic parameters

Description of parameter values	Subroutine name of the M-function	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3090 00004	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

15.2.4 Modal subroutines SwiOn[1..63] / SwiOff

Switch-off function SwiOff

Switch-off syntax

Switch-off syntax		SwiOffSynSubProg
Defines the syntax of an NC function with which the "modal action" of the modal group of NC functions can be switched off from the parameters		
../SwiOn[1..63]/SwiOnSynSubProg		
.		
Path	NCP/SubProg/ModalSubProg/SwiOff/SwiOffSynSubProg	
Value range	{NC syntax}	
Pre-assignment	G80	
Unit	-	
Description of parameter values	Switches off the modal action of the NC functions under ../SwiOn[1..63]/SwiOnSynSubProg (Switch-off function).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3090 00005	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

Switch-on function SwiOn[1..63]

Switch-on syntax

Switch-on syntax	SwiOnSynSubProg
Defines the NC syntax of up to 63 functions	
SwiOn[1..63]/SwiOnSynSubProg,	
used to call 63 modal subroutines.	
If it locates a function defined by individual parameter SwiOn[1]/SwiOnSynSubProg to SwiOn[63]/SwiOnSynSubProg in the NC block currently active, the control starts the program that is entered in the corresponding individual parameter of SwiOn[1..63]/NameSubProg.	

Using the standard NC syntax, data can also be transferred to the program called. These data remain available until the respective function or the modal effect is deselected. The maximum number of parameters to be transferred can be specified in the corresponding individual parameter of SwiOn[1..63]/NofParSubProg.

This allows to include own, modally acting functions in the NC syntax.

"Modal" means that the respective subroutine is called up after each path information programmed. This effect is active until the control finds another defined NC syntax in an active NC block than specified in the following parameter:

SwiOn[1..63]/SwiOnSynSubProg

Path	NCP/SubProg/ModalSubProg/SwiOn[1..63]/SwiOnSynSubProg	
Value range	{NC syntax}	
Pre-assignment	The entries are preassigned with the NC syntax of the MTX standard cycles:	
	G81	.../SwiOn[1]/SwiOnSynSubProg
	G82	.../SwiOn[2]/SwiOnSynSubProg
	G83	.../SwiOn[3]/SwiOnSynSubProg
	G84	.../SwiOn[4]/SwiOnSynSubProg
	G85	.../SwiOn[5]/SwiOnSynSubProg
	G86	.../SwiOn[6]/SwiOnSynSubProg
	G87	.../SwiOn[7]/SwiOnSynSubProg
	G88	.../SwiOn[8]/SwiOnSynSubProg
	G89	.../SwiOn[9]/SwiOnSynSubProg
	G165	.../SwiOn[10]/SwiOnSynSubProg
	G261	.../SwiOn[11]/SwiOnSynSubProg
	G262	.../SwiOn[12]/SwiOnSynSubProg
	G263	.../SwiOn[13]/SwiOnSynSubProg
	G266	.../SwiOn[14]/SwiOnSynSubProg
	G267	.../SwiOn[15]/SwiOnSynSubProg
	G271	.../SwiOn[16]/SwiOnSynSubProg
	G272	.../SwiOn[17]/SwiOnSynSubProg
	G274	.../SwiOn[18]/SwiOnSynSubProg
	G275	.../SwiOn[19]/SwiOnSynSubProg
	G276	.../SwiOn[20]/SwiOnSynSubProg
	G277	.../SwiOn[21]/SwiOnSynSubProg
	G279	.../SwiOn[22]/SwiOnSynSubProg
	G280	.../SwiOn[23]/SwiOnSynSubProg
	<Blank>	.../SwiOn[24..63]/SwiOnSynSubProg
Unit	-	
Description of parameter values	Defines the NC function for calling up a subroutine. All 63 functions SwiOn[1..63]/SwiOnSynSubProg form a "modal group", i.e. only one of these functions can be active at a time. The functions deselect one another mutually.	
Access rights	MTB:	Write
	Setter:	Read

NC programming (NCP) - Basic parameters

	Operator:	Read
Activation		System restart (CS)
Dependency		-
Macoda ID		3090 00005
Data type		Str8_t (isoLatin1String)
		8 byte string for MACODA compatibility

Subroutine name

Subroutine name		SwiOn[1..63]/NameSubProg
Defines up to 63 names SwiOn[1..63]/NameSubProg of modal subroutines, which are called up in dependence of SwiOn[1..63]/SwiOnSynSubProg. If it locates a function defined by individual parameter SwiOn[1]/SwiOnSynSubProg to SwiOn[63]/SwiOnSynSubProg in the NC block currently active, the control starts the program that is entered in the corresponding individual parameter of SwiOn[1..63]/NameSubProg. Using the standard NC syntax, data can also be transferred to the program called. These data remain available until the respective function or the modal effect is deselected. The maximum number of parameters to be transferred can be specified in the corresponding individual parameter of SwiOn[1..63]/NofParSubProg. This allows to include own, modally acting functions in the NC syntax. "Modal" means that the respective subroutine is called up after each path information programmed. This effect remains effective until the control locates another defined NC syntax in SwiOn[1..63]/SwiOnSynSubProg in the NC block.		
Path	NCP/SubProg/ModalSubProg/SwiOn[1..63]/NameSubProg	
Value range	{Text}	
Pre-assignment	The entries are preassigned with names of the MTX standard cycles:	
	STC_G81	.../SwiOn[1]/NameSubProg Drilling
	STC_G82	.../SwiOn[2]/NameSubProg Countersinking
	STC_G83	.../SwiOn[3]/NameSubProg Deep hole drilling
	STC_G84	.../SwiOn[4]/NameSubProg Tapping without compensating chuck (left, right)
	STC_G85	.../SwiOn[5]/NameSubProg Reaming
	STC_G86	.../SwiOn[6]/NameSubProg Fine boring
	STC_G87	.../SwiOn[7]/NameSubProg Back boring
	STC_G88	.../SwiOn[8]/NameSubProg Gun drill

	STC_G89	.../SwiOn[9]/NameSubProg Rigid tapping with chip breaking
	STC_G165	.../SwiOn[10]/NameSubProg Thread single cut
	STC_G261	.../SwiOn[11]/NameSubProg Thread milling
	STC_G262	.../SwiOn[12]/NameSubProg Drill thread milling
	STC_G263	.../SwiOn[13]/NameSubProg Circulating
	STC_G266	.../SwiOn[14]/NameSubProg Face milling, rectangle
	STC_G267	.../SwiOn[15]/NameSubProg Face milling, circle
	STC_G271	.../SwiOn[16]/NameSubProg Rectangular pocket
	STC_G272	.../SwiOn[17]/NameSubProg Circular Pocket
	STC_G274	.../SwiOn[18]/NameSubProg Slot straight
	STC_G275	.../SwiOn[19]/NameSubProg Slot curved
	STC_G276	.../SwiOn[20]/NameSubProg Rectangular pin
	STC_G277	.../SwiOn[21]/NameSubProg Circle pin
	STC_G279	.../SwiOn[22]/NameSubProg Ground link straight
	STC_G280	.../SwiOn[23]/NameSubProg Ground link curved
	<Blank>	.../SwiOn[24..63]/NameSubProg
Unit	-	
Description of parameter values	Name of a modal subroutine	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3090 00006	
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility	

Number of parameters

	Number of parameters	SwiOn[1..63]/NofParSubProg
	Defines the maximum number of parameter values SwiOn[1..63]/NofParSubProg that can be transferred for every subroutine specified in SwiOn[1..63]/NameSubProg If it locates a function defined by individual parameter SwiOn[1]/SwiOnSynSubProg to SwiOn[63]/SwiOnSynSubProg in the NC block currently active, the control starts the program that is entered in the corresponding individual parameter of SwiOn[1..63]/NameSubProg. Using the standard NC syntax, data can also be transferred to the program called. These data remain available until the respective function or the modal effect is deselected. The maximum number of parameters to be transferred can be specified in the corresponding individual parameter of SwiOn[1..63]/NofParSubProg. This allows to include own, modally acting functions in the NC syntax. "Modal" means that the respective subroutine is called up after each path information programmed. This effect remains until the control finds the NC syntax (switch-off syntax) defined in SwiOn[0]/SwiOnSynSubProg in the active block.	
Path	NCP/SubProg/ModalSubProg/SwiOn[1..63]/NofParSubProg	
Value range	0..99	
Pre-assignment	5	.../SwiOn[1]/NofParSubProg (G81)
	10	.../SwiOn[2]/NofParSubProg (G82)
	13	.../SwiOn[3]/NofParSubProg (G83)
	8	.../SwiOn[4]/NofParSubProg (G84)
	6	.../SwiOn[5]/NofParSubProg (G85)
	8	.../SwiOn[6]/NofParSubProg (G86)
	8	.../SwiOn[7]/NofParSubProg (G87)
	13	.../SwiOn[8]/NofParSubProg (G88)
	10	.../SwiOn[9]/NofParSubProg (G89)
	5	.../SwiOn[10]/NofParSubProg (G165)
	15	.../SwiOn[11]/NofParSubProg (G261)
	16	.../SwiOn[12]/NofParSubProg (G262)
	12	.../SwiOn[13]/NofParSubProg (G263)
	17	.../SwiOn[14]/NofParSubProg (G266)
	14	.../SwiOn[15]/NofParSubProg (G267)
	20	.../SwiOn[16]/NofParSubProg (G271)
	16	.../SwiOn[17]/NofParSubProg (G272)
	16	.../SwiOn[18]/NofParSubProg (G274)
	18	.../SwiOn[19]/NofParSubProg (G275)
	20	.../SwiOn[20]/NofParSubProg (G276)
	16	.../SwiOn[21]/NofParSubProg (G277)
	18	.../SwiOn[22]/NofParSubProg (G279)
	20	.../SwiOn[23]/NofParSubProg (G280)
	0	.../SwiOn[24..63]/NofParSubProg

Unit	-
Description of parameter values	Maximum number of transferable parameter values for the modal subroutine defined with SwiOn[1..63]/NofParSubProg .
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	3090 00007
Data type	Byte0_99_t (byte) 1 byte integer for MACODA compatibility

15.3 Polar coordinate programming (PolarCoord)

15.3.1 Syntax of angle 1

Syntax of angle 1		PolarCoordAngle1
Defines the NC syntax of angle 1 for polar coordinate programming. Since a parameter for the name of the angle can be set here, conflicts between existing axis names and the name of the angle can be avoided.		
Path	NCP/PolarCoord/PolarCoordAngle1	
Value range	{NC syntax}	
Pre-assignment	A	
Unit	-	
Description of parameter values	Syntax of angle 1 for polar coordinate programming	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8005 00001	
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility	

15.3.2 Syntax of angle 2

Syntax of angle 2		PolarCoordAngle2
Defines the NC syntax of angle 2 for polar coordinate programming. Since a parameter for the name of the angle can be set here, conflicts between existing axis names and the name of the angle can be avoided.		
Path	NCP/PolarCoord/PolarCoordAngle2	
Value range	{NC syntax}	
Pre-assignment	B	
Unit	-	

NC programming (NCP) - Basic parameters

Description of parameter values	Syntax of angle 2 for polar coordinate programming	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8005 00002	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	

15.4 Corner rounding (CornRound)

15.4.1 Maximum allowed deviation from the contour

Maximum allowed deviation from contour		MaxDev
Specifies for each channel Ch[k] the maximum permissible deviation between the ground contour and the originally programmed contour.		
The parameter MaxDev in the folder NCP/CornRound/Ch[k]/ applies to both "circular corner rounding" and "corner rounding with splines". These functions finish the contour transition between two following blocks because the control automatically inserts a transition circle or a spline.		
Path	NCP/CornRound/Ch[k]/MaxDev	
Value range	0.000 .. 1000.000	
	0.001 .. 999.999 for MACODA compatibility	
Pre-assignment	0.1	
Unit	mm	
Description of parameter values	Maximum permissible deviation from originally programmed contour.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00110	
Data type	Double0_1T_t (double, 3 decimal places)	
	4 byte float, 3 decimal places for MACODA compatibility	

15.4.2 Minimum block length for corner rounding

Minimum block length for corner rounding	MinLen
Specifies the minimum block length "MinLen" from which a circular transition or spline is generated for each channel Ch[k].	
The parameter MinDev in the folder NCP/CornRound/Ch[k]/ applies to both "circular corner rounding" and "corner rounding with splines".	
If machining paths are generated with CAD systems, path segments of very short travel distance frequently occur, which are due to inadequate numbers. For these blocks, the "corner rounding" function would not really make sense.	
Path	NCP/CornRound/Ch[k]/MinLen

Value range	2 .. 90
Pre-assignment	2
Unit	µm
Description of parameter values	Minimum block length from which a circular transition block or spline is generated.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7050 00120
Data type	MinLenCornRound_t (float, 0 decimal places) 4 byte float, 0 decimal places for MACODA compatibility

15.4.3 Angle up to which the block transitions are seen as continuous

Angle up to which block transitions are considered continuously		MaxAng
Specifies for each channel Ch[k] up to what angle "MaxAng" block transitions are considered as continuous. No circular transition blocks or splines are computed for block transitions involving angles that are smaller than the angle entered here. The parameter "MaxAng" in the folder NCP/CornRound/Ch[k] applies to both "circular corner rounding" and "corner rounding with splines".		
Path	NCP/CornRound/Ch[k]/MaxAng	
Value range	0.000 .. 45.000	
Pre-assignment	1.000	
Unit	° (degrees)	
Description of parameter values	Maximum angle up to which block transitions are considered continuous.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00130	
Data type	MaxAngCornRound_t (double, 3 decimal places) 4 byte float, 3 decimal places for MACODA compatibility	

15.5 Diameter programming (DiaProg)

15.5.1 Effect of incremental specifications when programming the diameter

Effect of incremental specifications when programming the diameter	IncrSetPointsDiaProg
Determines how incrementally programmed specifications are interpreted when programming the diameter.	
Path	NCP/DiaProg/IncrSetPointsDiaProg

NC programming (NCP) - Basic parameters

	Value range	0	Diameter difference
		1	Radius difference
Pre-assignment		0	Diameter difference
Unit		-	
Description of parameter values		0	Diameter difference
			Incremental specifications are interpreted as diameter values.
		1	Radius difference
			Incremental specifications are interpreted as radius values.
Access rights	MTB:		Write
	Setter:		Write
	Operator:		Read
Activation			System reset (SR)
Dependency		-	
Macoda ID		8010 00001	
Data type		Byte0_1_t (byte)	
			1 byte integer, unsigned for MACODA compatibility

15.5.2 Channel axes/coordinates[1..8] that can be activated

Channel axes/coordinates that can be activated[1..8]		Discord[1..8]
Defines 1..8 channel-specific axes or coordinates as diameter coordinates DiaCoord[1..8], that can be programmed with DIA(...).		
Path	NCP/DiaProg/Ch[k]/DiaCoord[1..8]	
Value range	0.. 8	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Axis of axis classification 1 is diameter axis.
	>0	Number of coordinates in the channel Ch[k] that becomes the diameter axis.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7010 00210	
Data type	ChAx_t (unsignedByte)	
	1 byte integer, unsigned for MACODA compatibility	
<i>Example:</i>		
Diameter programming		
A channel has the following axis configuration:		
1: x		
2: y		
3: z		

4: VA

5: C

With the settings DiaCoord[1] = 1 and DiaCoord[2] = 4, the coordinate x and the axis VA become diameter axes xØ and VAØ as soon as DIA(...) activates diameter programming.

15.6 Tool path correction (CutComp)

15.6.1 Starting and exiting correction build-up strategy

Starting and exiting correction build-up strategy for tool path correction		CutCompActDeact-Mode
Specifies the strategy used to start and exit the tool path correction build-up (tool edge radius compensation).		
Path	NCP/CutComp/CutCompActDeactMode	
Value range	0	Immediately after programming
	1	With the next programmed traversing motion
Pre-assignment	0	Immediately after programming
Unit	-	
Description of parameter values	0	Immediately after programming The tool path correction (tool edge radius compensation) is exactly started and exited at the position at which the respective NC command "G41", "G42" or "G40" is programmed.
	1	With the next programmed traversing motion The tool path correction (tool edge radius compensation) is only started or exited at the next programmed traversing motion.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8009 00001	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

16 NC optimization (NCO) - Basic parameters

16.1 General information

The following settings can be made in the data group NC Optimization (NCO):

- [Look ahead, page 279](#)
- [Memory management, page 288](#)
- [File organization, page 291](#)
- [Processor utilization, page 295](#)
- [Path velocity, page 296](#)
- [In-Position evaluation, page 297](#)
- [Position window, page 297](#)
- [Date / time setting, page 298](#)
- [Accuracy specification for circular programming, page 306](#)
- [Velocity scaling, page 307](#)
- [Accuracy programming, page 312](#)
- [Block transfer inhibit, page 313](#)
- [System units, page 313](#)
- [Database tables, page 315](#)
- [Communication server, page 316](#)
- [Web server, page 316](#)

16.2 Look-ahead

16.2.1 General information

The "Look-ahead" function includes the following channel-specific settings:

- Total number of blocks for block preparation and interpolation
- Max. number of function-related blocks
- Percentage distribution of the entire number of blocks

Parameter <AdditionalNCBlockMem> in the startup.xml file

The CNC control requires memory especially for the NC block data. The memory requirement is estimated using the configuration and is reserved during control startup for this purpose. In some use cases, this evaluation cannot be sufficient so that the waiting state of the channel 20 (waiting for resources) or the channel error 1927 is generated in case of lack of memory.

If these effects occurs, the memory area for the NC block data has to be increased. This should be realized via parameter AdditionalNCBlockMem in the "startup.xml" file. The parameter is specified in bytes and added to the internally determined memory size.

Due to the variety of applications and configurations of the CNC control, it is not possible to specify a value valid for all applications. As a rule of thumb, 10000 bytes per configured NC block in the channel (Macoda parameter 7060 00110) can be assumed. For example, if 300 blocks have been configured for channel 1 and 500 have been configured for channel 2, a good value would be $(300+500)*10000 = 8000000$ - provided that NC programs are processed simultaneously in both channels and a very high number of blocks is actually required.

Example:

Entry for the “startup.xml” file:

```
<AdditionalNCBlockMem>8000000</AdditionalNCBlockMem>
```

The template of the “startup.xml” file is contained in the “/feprom” directory. The changed file has to be stored in the directories “/usrfep” or “/” (root directory) of the control.

The option to increase the memory size by creating another unused channel with a high number of blocks is not recommended. The memory area is increased by the additional channel but the memory requirement and the administrative expense is also increased by the additional channel. The parameter contained in “startup.xml” is much more memory-efficient.

16.2.2 Total number of blocks for block preparation and interpolation

Total number of blocks for block preparation and interpolation	NofBlkPrep
Specifies for each channel (Ch[k]) the number of NC blocks in total to be available for look-ahead and interpolation.	
The interpolator requires at most as many blocks as have been set or programmed as SHAPE order. This maximum occurs when all NC blocks are so short that they are completely executed within one interpolation cycle. Advance and subsequent interpolation are then separated from each other by precisely the number of SHAPE order blocks. If the interpolation NC blocks are longer, the block demand of the interpolator is reduced accordingly. All NC blocks that are not needed by the interpolation are available to the block preparation NC functions for look-ahead.	
 The set number of blocks does not only include real blocks from the NC program, but also internal blocks (insert blocks) that depend on different active functions!	

Path	NCO/LookAh/Ch[k]/NofBlkPrep	
Value range	3 .. 999	
Pre-assignment	30	NCO/LookAh/Ch[1]/NofBlkPrep
	3	NCO/LookAh/Ch[2..60]/NofBlkPrep
Unit	-	
Description of parameter values	Maximum number of look-ahead blocks	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7060 00110	
Data type	Blk_t (int)	
	4 byte integer, unsigned for MACODA compatibility	

16.2.3 Maximum number of function-related blocks

Maximum number of jerk-limited velocity control

Maximum number of jerk-limited velocity control	MaxBlkVelCalc

It defines the maximum number of NC blocks to be used for look-ahead for "jerk-limited velocity control G8" channel-specifically (Ch[k]).	
Path	NCO/LookAh/Ch[k]/MaxBlk/MaxBlkVelCalc
Value range	0 .. 1,000
Pre-assignment	10
Unit	-
Description of parameter values	0 No look-ahead
	>0 Maximum number of NC blocks for look-ahead
 The set maximum values are only adhered to if the total number has been configured sufficiently large in the parameter "NofBlk-Prep" and is also assigned a sufficiently large portion of the function in the parameter "PercVelCalc" (in the folder NCO/LookAh/Ch[k]/PercSplit/).	
Access rights	MTB: Write
	Setter: Read
	Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7060 00120 [1]
Data type	Int0_1T_t (unsignedInt)
4 byte integer, unsigned for MACODA compatibility	

Maximum block number for block buffer

Maximum block number for block buffer		MaxBlkBuffered
Defines for each channel (Ch[k]) the maximum number of blocks whose average path length limits the maximum path velocity.		
Path	NCO/LookAh/Ch[k]/MaxBlk/MaxBlkBuffered	
Value range	0 .. 1,000	
Pre-assignment	0	
Unit	-	
Description of parameter values	n = 0	The maximum path velocity is limited to the path velocity of each block, divided by the interpolation cycle. This has the effect that no block can be skipped.
	n > 0	The maximum path velocity is limited to the average path velocity of (n+1) subsequent blocks, divided by the interpolation cycle. This has the effect that, under favorable conditions, i.e. one of the n+1 blocks is long enough, up to n blocks can be skipped during interpolation.
 The set maximum values are only adhered to if the total number has been configured sufficiently large in the parameter "NofBlk-Prep" and is also assigned a sufficiently large portion of the function in the parameter "PercBuffered" (in the folder NCO/LookAh/Ch[k]/PercSplit/).		
Access rights	MTB:	Write
	Setter:	Read

NC optimization (NCO) - Basic parameters

	Operator:	Read
Activation		System reset (SR)
Dependency		-
Macoda ID		7060 00120 [2]
Data type		Int0_1T_t (unsignedInt)
		4 byte integer, unsigned for MACODA compatibility

Maximum block number for spline functions

	Max. number of blocks for Spline functions	MaxBlkSpline
	Specifies for each channel (Ch[k]) the number of NC blocks in total to be used for look-ahead for the "spline functions" (e.g. online calculation of the spline coefficients).	
Path	NCO/LookAh/Ch[k]/MaxBlk/MaxBlkSpline	
Value range	0 .. 1,000	
Pre-assignment	10	NCO/LookAh/Ch[1]/MaxBlk/MaxBlkSpline
	10	NCO/LookAh/Ch[2]/MaxBlk/MaxBlkSpline
	0	NCO/LookAh/Ch[3..60]/MaxBlk/MaxBlkSpline
Unit	-	
Description of parameter values	0	No look-ahead
	>0	Maximum number of NC blocks for look-ahead
	 The set maximum values are only adhered to if the total number has been configured sufficiently large in the parameter "NofBlk-Prep" and is also assigned a sufficiently large portion of the function in the parameter "PercSpline" (in the folder NCO/LookAh/Ch[k]/PercSplit/).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation		System reset (SR)
Dependency		-
Macoda ID		7060 00120 [3]
Data type		Int0_1T_t (unsignedInt)
		4 byte integer, unsigned for MACODA compatibility

Maximum block number for 3D milling cutter path correction

	Max. number of blocks for 3D milling cutter path correction	MaxBlkTCorr3D
	Specifies for each channel (Ch[k]) the number of NC blocks in total to be used for inner corner calculation for "3D-cutter radius correction".	
Path	NCO/LookAh/Ch[k]/MaxBlk/MaxBlkTCorr3D	
Value range	0 .. 1,000	
Pre-assignment	5	NCO/LookAh/Ch[1]/MaxBlk/MaxBlkTCorr3D
	5	NCO/LookAh/Ch[2]/MaxBlk/MaxBlkTCorr3D
	0	NCO/LookAh/Ch[3..60]/MaxBlk/MaxBlkTCorr3D
Unit	-	

Description of parameter values	0	No look-ahead
	>0	Maximum number of NC blocks for inner corner calculation
		The set maximum values are only adhered to if the total number has been configured sufficiently large in the parameter "NofBlk-Prep" and is also assigned a sufficiently large portion of the function in the parameter "PercTCorr3D" (in the folder NCO/LookAh/Ch[k]/PercSplit/).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7060 00120 [5]	
Data type	Int0_1T_t (unsignedInt)	
	4 byte integer, unsigned for MACODA compatibility	

Maximum block number for other functions

Max. number of blocks for other functions		MaxBlkMiscFunc
Defines the maximum number of NC block for "other functions" channel-specifically (Ch[k]).		
This parameter is configured for all functions with small and medium look ahead range such as chamfers+radiuses, tangential tool guidance, 2.5D path correction, distance-dependent signal (ADS).		
MaxBlkMiscFunc also contains the number of blocks which can be in preparation simultaneously within the interpolation. For an active path SHAPE, up to 100 blocks might lie between the advance and subsequent interpolation if the NC blocks are very short.		
		When using the "Distance-dependent signal (ADS)" function, a value of at least 45 has to be set
Path	NCO/LookAh/Ch[k]/MaxBlk/MaxBlkMiscFunc	
Value range	0 .. 1,000	
Pre-assignment	5	NCO/LookAh/Ch[1]/MaxBlk/MaxBlkMiscFunc
	5	NCO/LookAh/Ch[2]/MaxBlk/MaxBlkMiscFunc
	3	NCO/LookAh/Ch[3..60]/MaxBlk/MaxBlkMiscFunc
Unit	-	
Description of parameter values	0	No look-ahead
	>0	Maximum number of NC blocks for inner corner calculation
		The set maximum values are only adhered to if the total number has been configured sufficiently large in the parameter "NofBlk-Prep" and is also assigned a sufficiently large portion of the function in the parameter "PercTCorr3D" (in the folder NCO/LookAh/Ch[k]/PercSplit/).
Access rights	MTB:	Write
	Setter:	Read

	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7060 00120 [4]	
Data type	Int0_1T_t (unsignedInt)	
	4 byte integer, unsigned for MACODA compatibility	

16.2.4 Percentage distribution of the total number of blocks

General information

For all functions (jerk-limited velocity profile generation G8, buffered blocks, spline functions, 3D milling cutter path correction, other functions), the following must be taken into account when assigning distribution in percent of the entire number of blocks for look-ahead:

The maximum number of blocks for each individual parameter is:

$$N_{\max} = \min([MaxBlkVelCalc|MaxBlkBuffered|MaxBlkSpline|MaxBlkTCorr3D]|MaxBlkMiscFunc, NofBlkPrep * [PercVelCalc|PercBuffered|PercSpline|PercTCorr3D|PercMiscFunc]).$$

The actually considered maximum block number $N_{\max}$ might be limited by the respective maximum value from
[MaxBlkVelCalc|MaxBlkBuffered|MaxBlkSpline|MaxBlkTCorr3D|MaxBlkMisc-
Func]
(see figure below).

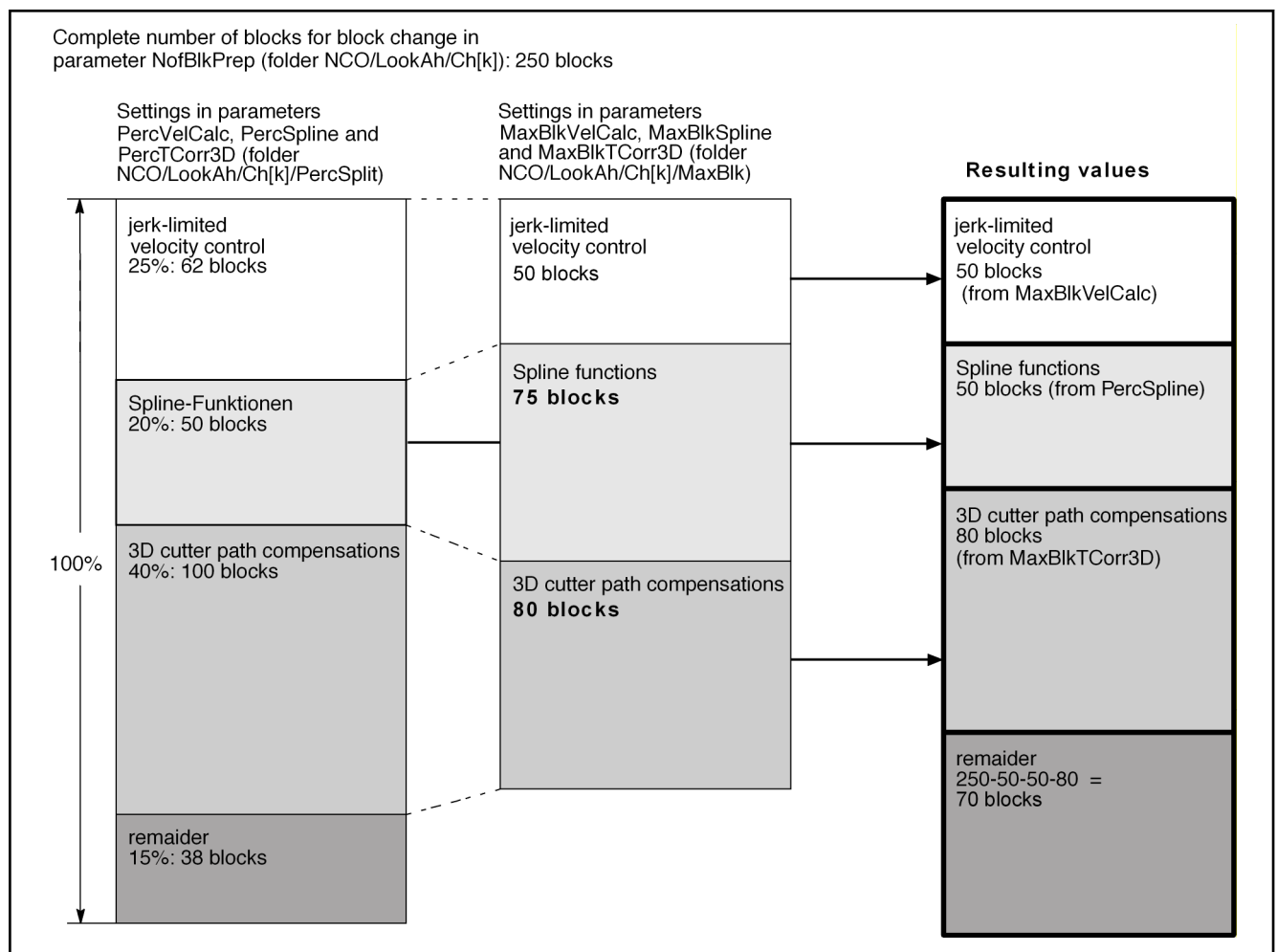


Fig. 16-1: Determination of the number of blocks for look-ahead

Percentage number of blocks of jerk-limited velocity control

Percentage number of blocks of jerk-limited velocity control		PercVelCalc
Specifies for each channel (Ch[k]) for "jerk-limited velocity profile generation G8" the percentage of the total number of blocks to be scaled for look-ahead (from NofBlkPrep).		
Path	NCO/LookAh/Ch[k]/PercSplit/PercVelCalc	
Value range	0 .. 100	
Pre-assignment	30	NCO/LookAh/Ch[1]/PercSplit/PercVelCalc
	30	NCO/LookAh/Ch[2]/PercSplit/PercVelCalc
	0	NCO/LookAh/Ch[3..60]/PercSplit/PercVelCalc
Unit	%	
Description of parameter values	Percentage of total number of blocks for look-ahead.	
 If the total of the values of PercVelCalc, PercBuffered, PercSpline PercTCorr3D and PercMiscFun is greater than 100%, an internal standardization in the total to 100% is carried out.		
Access rights	MTB:	Write
	Setter:	Read

NC optimization (NCO) - Basic parameters

	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7060 00130 [1]	
Data type	Byte0_100_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Percentage number of blocks for block buffer

Percentage of number of blocks for block buffer		PercBuffered
Defines for each channel (Ch[k]) the percentage of number of blocks (related to NofBlkPrep) whose average path length limits the maximum path velocity.		
Path	NCO/LookAh/Ch[k]/PercSplit/PercBuffered	
Value range	0 .. 100	
Pre-assignment	5	NCO/LookAh/Ch[1]/PercSplit/PercBuffered
	5	NCO/LookAh/Ch[2]/PercSplit/PercBuffered
	0	NCO/LookAh/Ch[3..60]/PercSplit/PercBuffered
Unit	%	
Description of parameter values	Percentage of total number of blocks for look-ahead.	
 If the total of the values of PercVelCalc, PercBuffered, PercSpline, PercTCorr3D and PercMiscFun is greater than 100%, an internal standardization in the total to 100% is carried out.		
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7060 00130 [2]	
Data type	Byte0_100_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Percentage of the block number for spline functions

Percentage of the block number for spline functions		PercSpline
Specifies for the "spline functions" (e.g. online calculation of the spline coefficients) for each channel (Ch[k]) what percentage of the total number of blocks is to be used for the preview .		
Path	NCO/LookAh/Ch[k]/PercSplit/PercSpline	
Value range	0 .. 100	
Pre-assignment	35	NCO/LookAh/Ch[1]/PercSplit/PercVelCalc
	35	NCO/LookAh/Ch[2]/PercSplit/PercSpline
	0	NCO/LookAh/Ch[3..60]/PercSplit/PercSpline
Unit	%	
Description of parameter values	Percentage of total number of blocks for look-ahead.	



If the total of the values of PercVelCalc, PercBuffered, PercSpline PercTCorr3D and PercMiscFunc is greater than 100%, an internal standardization in the total to 100% is carried out.

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7060 00130 [3]	
Data type	Byte0_100_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Percentage number of blocks for 3D milling cutter path correction

Percentage of the block number for 3D milling cutter path correction	PercTCorr3D

Specifies for each channel (Ch[k]) for "3D milling cutter radius correction" the percentage of the total number of blocks to be scaled for look-ahead (from NofBlkPrep).

Path	NCO/LookAh/Ch[k]/PercSplit/PercTCorr3D	
Value range	0 .. 100	
Pre-assignment	15	NCO/LookAh/Ch[1]/PercSplit/PercTCorr3D
	15	NCO/LookAh/Ch[2]/PercSplit/PercTCorr3D
	0	NCO/LookAh/Ch[3..60]/PercSplit/PercTCorr3D
Unit	%	
Description of parameter values	Percentage of total number of blocks for look-ahead.	



If the total of the values of PercVelCalc, PercBuffered, PercSpline PercTCorr3D and PercMiscFunc is greater than 100%, an internal standardization in the total to 100% is carried out.

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7060 00130 [5]	
Data type	Byte0_100_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Percentage of number of blocks for other functions

Percentage of number of blocks for other functions	PercMiscFunc

Specifies for each channel (Ch[k]) for "other functions" the percentage of the total number of blocks to be scaled for look-ahead (from NofBlkPrep).

Path	NCO/LookAh/Ch[k]/PercSplit/PercMiscFunc	
Value range	0 .. 100	
Pre-assignment	15	NCO/LookAh/Ch[1]/PercSplit/PercMiscFunc

NC optimization (NCO) - Basic parameters

	15	NCO/LookAh/Ch[2]/PercSplit/PercMiscFunc
	0	NCO/LookAh/Ch[3..60]/PercSplit/PercMiscFunc
Unit	%	
Description of parameter values	Percentage of total number of blocks for look-ahead.	
	 If the total of the values of PercVelCalc, PercBuffered, PercSpline, PercTCorr3D and PercMiscFun is greater than 100%, an internal standardization in the total to 100% is carried out.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7060 00130 [4]	
Data type	Byte0_100_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

16.3 Memory management (Mem)

16.3.1 General information

Memory management distinguishes between

- System memory
- Channel-specific memory

16.3.2 System memory

Buffer size to process external programs via NFS

	Buffer for processing external programs via NFS	BuffSizeNfs
	Specifies the size of a buffer where data received from active NFS servers is stored temporarily. With its integrated NFS client, the MTX allows 2 directories which are stored externally to be "mounted" to its internal file system. This is possible only if the Rexroth NFS server provides access to these directories on the respective source computers in the network. For instance, part programs for the entire machine stock are very often stored centrally in these external directories. If such programs are called up for execution, the control receives the data via the network, also during the machining process. However, if extremely large data volumes (for instance, part programs for freeform machining) are involved and large network loads occur, this may slow down the flow of data required by the controller. Consequence: Machining falters. This can be prevented by a buffer of adequate size.	
Path	NCO/Mem/SysMem/BuffSizeNfs	
Value range	2048 .. 4194304	
Pre-assignment	8192	
Unit	Byte	

Description of parameter values	Buffer Size for External Execution	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3080 00002	
Data type	BuffNfs_t (int)	
	4 byte integer, unsigned for MACODA compatibility	

16.3.3 Channel-specific memory

Buffer size for external block selection

Buffer size for NC block selection		BuffSizeExternBlk
Specifies for each channel Ch[k] the buffer size to be reserved for buffered NC block selection if NC blocks are stored temporarily in a buffer until they are processed.		
Path	NCO/Mem/ChMem/Ch[k]/BuffSizeExternBlk	
Value range	0 .. 1048576	
Pre-assignment	1024	
Unit	Byte	
Description of parameter values	Memory size of buffer	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7060 00510	
Data type	BuffSizeExBlk_t (int)	
	4 byte integer for MACODA compatibility	

Memory for CPL programs (subroutines, variable, ...)

Memory for CPL (UPs, variable, ...)		CplStack
Reserves memory for stack for each channel Ch[k].		
During their execution, CPL programs need the "stack" for subroutine calls, local variables, etc.		
Path	NCO/Mem/ChMem/Ch[k]/CplStack	
Value range	1024 .. 3145728	
Pre-assignment	4096	
Unit	Byte	
Description of parameter values	Memory size for stack	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

NC optimization (NCO) - Basic parameters

Activation	System restart (CS)
Dependency	-
Macoda ID	7070 00010 [1]
Data type	CplStack_t (int) 4 byte integer, unsigned for MACODA compatibility

Memory for CPL link tables (UP-Tab, Var.-Tab., ...)

Memory for CPL link tables (SR table, Var table, ...)		CplLinkBuff
Reserves memory for link tables for each channel Ch[k]. Editing a CPL program requires a "link table", which is automatically created when the program is called. The link table contains e.g. a subroutine table, tables of variables, and jump tables.		
Path	NCO/Mem/ChMem/Ch[k]/CplLinkBuff	
Value range	1024 .. 3145728	
Pre-assignment	20480	
Unit	Byte	
Description of parameter values	Memory size for link tables	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7070 00010 [2]	
Data type	CplStack_t (int) 4 byte integer, unsigned for MACODA compatibility	

Number of blocks for retrace mode

Number of blocks retrace mode		NofBlkRetrace
Specifies the number of blocks in the retrace memory for the reverse on the contour for each channel (Ch[k]). The blocks on which the reversing should take place are buffered in the retrace memory. Using this parameter you can specify by how many NC blocks driving back on the path is possible. Note that blocks not triggering traversing motions are counted as well. e.g. subroutine calls: N100 P SR.		
Path	NCO/Mem/ChMem/Ch[k]/NofBlkRetrace	
Value range	0 .. 100000	
Pre-assignment	0	
Unit	-	
Description of parameter values	Number of	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7060 00610	

Data type Int0_100000_t
4 byte integer, unsigned for MACODA compatibility

16.4 File organization (FileOrg)

16.4.1 General information

In the file organization, find settings for

- Search paths for programs and link tables
- Max. number of simultaneously open files
- Linking type

16.4.2 Search paths

Search paths for subroutines

Search	SrchPathSubProg

Defines the search path (directory tree) in which MTX should search for called (sub)routines.

The default directory structure of the MTX is shown in the figure below:

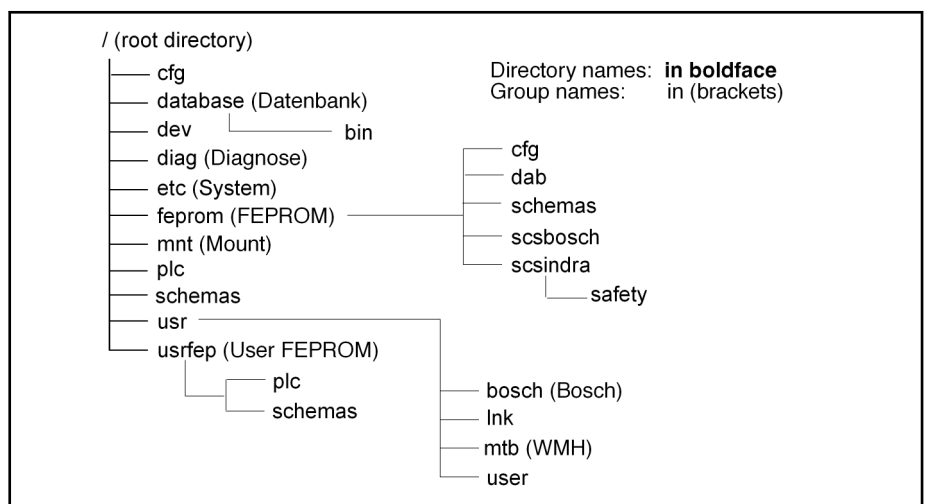


Fig. 16-2: Standard MTX directory structure

Path	NCO/FileOrg/SrchPathSubProg
Value range	{Text}
Pre-assignment	. /usr /usr/ user mtb bosch /usrfepr /feprom/ cycles /feprom
Unit	-
Description of parameter values	Given the text entries specified below and starting from the root directory, various different "search paths" can be put together with which the directories to be searched are defined.

The following applies:

- The text entries of the parameter "SrchPathSubProg" are written to a **comprehensive string** and are separated by a **blank**. With regard to the restrictions of the string length that can actually be used, read the notes given after "Data type".
- The character "." marks the beginning of a new path starting from the directory of the program calling. A **blank** has to come in front of further characters.

- A preceding "/" marks the beginning of a new path starting from the root directory.
- A "/" following the text entry turns the path specified so far into a prefix for all following entries (up to an entry beginning with "/" or ".").
- The continue character ">" (previously used for directory names longer than 8 characters) must be **omitted**.
- Only directory names of the MTX directory structure are permitted for directory texts.

Example:

Search path for subroutines

With the default settings, the search is conducted in the following order:

1. . (directory of the calling program)
2. /usr
3. /usr/user
4. /usr/mtb
5. /usr/bosch
6. /usrfep
7. /feprom/cycles
8. /feprom

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3080 00001	
Data type	Str240_t (isoLatin1String) 24 * 8 byte string for MACODA compatibility	



This parameter elementarily differs in the entering in the NC configurator and the control-internal data storage. **Due to this difference, the processable length of the string entered in the NC configurator is significantly smaller than it would be possible with regard to the data type.**

*The following only applies to the **control-internal data storage under the Macoda ID 3080 00001**:*

- This parameter is organized control-internally in 24 x 8 byte string segments. This results in a length of 192 characters max. Due to the specialties in the data storage, not all the 192 characters can be used.
- Therefore, the character "." which identifies the beginning of a new path, has to stand alone in a 8 byte string segment. No other characters may be entered in the same segment.
- Additionally, the following entries are expected each in one individual segment. Due to this, many additional characters can remain unused.
- If a directory name is longer than 8 characters, the eighth character written is ">". The remaining directory name is continued in the next segment.

Search path for link tables

Search path for link tables		SrchPathLinkTab
	Defines the directory of the MTX (search path) where all generated link tables are stored.	
Path	NCO/FileOrg/SrchPathLinkTab	
Value range	{Text}	
Pre-assignment	/usr/lnk/	
Unit	-	
Description of parameter values	The search path of the directory (link tables) is made up from the following directory names. <i>The following applies:</i> • The search path must begin and end with the character "/". • A "/" has to be written as well between the directory names.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3080 00004	
Data type	Str80_t (isoLatin1String)	
	10 * 8 byte string for MACODA compatibility	

16.4.3 Link process optimization for NC programs with the same name

Enable link process optimization (same project names)

Enable link process optimization (same project names)		EnablEqProgName-sLinkOpti
Enables a link process optimization for NC programs with the same name. This allows an optimized handling when using NC programs with the same name, but stored in different directories.		
Path	NCO/FileOrg/EnablEqProgNamesLinkOpti	
Value range	0 (No) 1 (Yes)	
Pre-assignment	0 (No)	
Unit	-	
Description of parameter values	0 (No) The link process optimization is disabled for identical NC program names. However, if NC programs with the same name are used, but stored in different directories, disturbing post-linking of these programs can occur frequently at runtime. 1 (Yes) The link process optimization is enabled for identical NC program names. An optimum handling of NC programs with the same name in different directories is thus ensured.	
Access rights	MTB: Write Setter: Write Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	3080 00010	
Data type	Byte0_1_t 1 byte integer, unsigned for MACODA compatibility	

16.4.4 Channel-specific setting of the file organization

Number of open files per channel

Number of open files per channel	MaxOpenFile
Specifies for each channel (Ch[k]) the number of files to be opened simultaneously.	
<i>This includes all</i>	
• CPL files opened with OPEN and • NC programs that are open at the same time (incl. all nested subroutines).	
 This parameter should be modified only for working with programs of substantial nesting depth or with a large number of CPL files open at the same time because a higher parameter value means that more system resources are tied up.	
Path	NCO/FileOrg/ChFileOrg/Ch[k]/MaxOpenFile
Value range	5 .. 200
Pre-assignment	10

Unit	-
Description of parameter values	Max. number of simultaneously open files.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	7060 00210
Data type	OpFiles_t (int) 2 byte integer for MACODA compatibility

Linking type

Linking type		KindOfLinking
Specifies the linking type for each channel Ch[k].		
Path	NCO/FileOrg/ChFileOrg/Ch[k]/KindOfLinking	
Value range	0	Linking at program runtime
	1	Linking at program selection
Pre-assignment	0	Linking at program runtime
Unit	-	
Description of parameter values	0	Linking at program runtime A selected NC program can be started immediately. Linking runs in the background at program runtime.
	1	Linking at program selection An NC program is immediately linked after the program selection. It can only be started after the linking.
Access rights	MTB: Write Setter: Write Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7070 00030	
Data type	Byte0_1_t (byte) 1 byte integer, unsigned for MACODA compatibility	

16.5 Processor load (Perf)

16.5.1 CPU time for block processing

CPU time for NC block processing	CpuTimeBlkPrep
Limits the portion of processor capacity made available by the MTX for block processing (SAV).	
Thus, the system can be scaled so as to permit the execution of applications of less than top priority (e.g. user interface) while part programs are being executed, which tie up a lot of CPU capacity.	
Path	NCO/Perf/CpuTimeBlkPrep

NC optimization (NCO) - Basic parameters

Value range	50 .. 100
Pre-assignment	98
Unit	%
Description of parameter values	Maximum percentage of total CPU time for NC block processing.
Access rights	MTB: Read Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	9040 00010
Data type	CpuTimeBlkPrep_t (byte) 4 byte integer for MACODA compatibility

16.6 Path velocity (PathVel)

16.6.1 Keep velocity in motion-neutral NC blocks

Keep velocity in motion-neutral NC blocks		KeepPathVelNoIpoAct
For each channel, it defines for each channel whether NC blocks with motion-neutral NC functions may cause a deceleration of the path velocity to v=0 (= compatibility mode) or whether the velocity is to be maintained. The functions "Exact stop" and "Path slope" are not affected.		
Path	NCO/PathVel/Ch[i]/KeepPathVelNoIpoAct	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		NC blocks with motion-neutral NC functions may cause a deceleration of the path velocity to v=0.
	1	Yes
		NC blocks with motion-neutral NC functions do not cause a deceleration of the path velocity to v=0 if not required.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7030 00500	
Data type	Byte0_1_t (byte)	

16.7 In-Position evaluation (InPosEval)

16.7.1 Mode of the In-Position evaluation

Mode of the In-Position evaluation		InPosEvalMode
For each channel, it defines how the In-Position information of the channel axes is evaluated in the interpolator.		
Path	NCO/InPosEval/Ch[i]/InPosEvalMode	
Value range	0	Evaluate all channel axes for all NC blocks
	1	Evaluate all channel axes only for traversing blocks
	2	Evaluate only channel axes with traversing command
Pre-assignment	0	Evaluate all channel axes for all NC blocks
Unit	-	
Description of parameter values	0	Evaluate all channel axes for all NC blocks This is the compatibility mode without restrictions.
	1	Evaluate all channel axes only for traversing blocks All channel axes are evaluated in this mode, but only for NC blocks with a traversing motion.
	2	Evaluate only channel axes with traversing command All channel axes are not evaluated anymore in this mode. Only the axes with a traversing command are still evaluated.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 00500	
Data type	Byte0_2_t (byte)	

16.8 Position window (InPosWin)

16.8.1 Inpos window, rough

Inpos window, rough	InPosRough

Specifies "In-position window, rough" for G61(IPS2) for each axis Dr[i].



This parameter is only of relevance to the following drive types:

- DiAx, EcoDrive (drive type in MAIN/Dr[i]/Scs/DrType = 1)
- HNC100 (drive type in MAIN/Dr[i]/Scs/DrType = 4)

The control automatically generates the signal "Axis in position" for the respective axis when "G61 (IPS2)" is active. Instead of the usually required Sercos parameter S-0-0261 in the drive, it uses the NC-internal command and actual values and the settings of these parameters.

As soon as the axis has finished a traversing motion and is within the set in-position window, the interface signal "Axis in position" is output.

NC optimization (NCO) - Basic parameters



The MTX provides the option of swapping between two different in-position windows in connection with the "Exact stop" function while G61(IPS2) and G61(IPS1) are active.

If G61(IPS1) is to be used for an "in-position window, fine", the Sercos parameter S-0-0057 must always exist in the drive.

Path	NCO/InPosWin/Dr[i]/InPosRough
Value range	-100.000 .. 100.000 -20.000 .. 20.000 for MACODA compatibility
Pre-assignment	0.01
Unit	mm or ° (degrees)
Description of parameter values	Size of the "In-position window, rough"
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1015 00100
Data type	Float-100_100_t (float, 3 decimal places) 4 byte float, 3 decimal places for MACODA compatibility

16.9 Date/time setting (DateTime)

16.9.1 General information

In the selection

- Winter
- Summer
- Clock change to winter time
- Clock change to summer

Date, time and rules for the clock change to summer/winter time can be set for an MTX.

In addition, details on the time zone are also specified:

- Name of Time Zone
- MTX location
- Changing between "Coordinated Universal Time (UTC)" (previously GMT) and "local time" at the MTX location.

Internally, the controller always uses UTC. The conversion to local time is made for display purposes only (system time, time stamps of files).



If no input is made in the parameters Summer, Winter and Clock change to winter/summer, then UTC time without changes for winter/summer applies.

If only the parameters for summer and winter are specified, then only the local time without changes for winter/summer will be displayed.

For information on global time zones and time differences "UTC", see the website "http://www.ptb.de/index_en.html" of the German national metrology institute "Physikalisch-Technische Bundesanstalt" on the Internet (English version available).

16.9.2 Winter

Name of time zone

Name of time zone (Winter time)		WinterZoneName
	Defines the name of the time zone for the MTX location during the winter period.	
Path	NCO/DateTime/Winter/WinterZoneName	
Value range	<3 .. 8 characters>	
	{blank}	
Pre-assignment	{blank}	
Unit	-	
Description of parameter values	{blank}	no switching between winter and summer time.
	<3 .. 8 characters>	Freely selectable name of the time zone at the location of the control (no digits, no "+", no "-"). Names with only one or two characters are not applied when entering.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	If either the parameter "WinterLoc" or "WinterDiffUtc" contain an entry, the "WinterZoneName" must also have a valid entry. If the parameter "WinterDiffUtc" is empty, the parameter "WinterLoc" is automatically deleted when deleting "WinterZoneName".	
Macoda ID	9050 00001 [1]	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	
	<i>Example:</i>	
	Time zone at the location of the controller:	
	CET (Central European Time)	

Location

Location (Winter time)	WinterLoc

Specifies the location of the controller in relation to longitude 0 (Greenwich) during the winter period.

NC optimization (NCO) - Basic parameters

Path	NCO/DateTime/Winter/WinterLoc	
Value range	{blank}	not specified
	+	west of Greenwich
	-	east of Greenwich
Pre-assignment	Status upon delivery: {empty} (no specification) Setting on default value in the NC configurator: - (east of Greenwich)	
Unit	-	
Description of parameter values	+	west of Greenwich Location of the controller is to the west of longitude 0 (Greenwich).
	-	east of Greenwich Location of the controller is to the east of longitude 0 (Greenwich).
	{blank}	not specified The entry remains empty if there is no switching between winter and summer time.
	Access rights	MTB: Write Setter: Read Operator: Read
	Activation	System restart (CS)
Dependency	The "WinterLoc" entry also requires entries in the parameters "WinterZoneName" and "WinterDiffUtc". "WinterLoc" is automatically deleted if the parameters "WinterZoneName" and "WinterDiffUtc" are deleted.	
Macoda ID	9050 00001 [2]	
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility	

Time difference UTC > Local time

Time difference UTC > local time (Winter time)		WinterDiffUtc
Specifies by how much the time differs between UTC time and local time during the winter period at the MTX location.		
Path	NCO/DateTime/Winter/WinterDiffUtc	
Value range	<8 characters>	
Pre-assignment	{blank}	
Unit	-	
Description of parameter values	Input of the time difference in the format "hh:mm:ss" (hours:minutes:seconds).	
	Leading zeroes, minutes and seconds are optional and can be left out.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	

Dependency	The "WinterDiffUtc" entry also requires entries in the parameters "WinterZoneName" and "WinterLoc". If the parameter "WinterZoneName" is empty, the parameter "WinterDiffUtc" is automatically deleted when deleting the parameter "WinterLoc".
Macoda ID	9050 00001 [3]
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility

16.9.3 Summer

Name of time zone

Name of time zone (summer time)		SummerZoneName
Defines the time zone for the MTX location during the summer period.		
Path	NCO/DateTime/Summer/SummerZoneName	
Value range	{3 .. 8 characters}	
	{blank}	
Pre-assignment	{blank}	
Unit	-	
Description of parameter values	{blank}	no switching between winter and summer time.
	<3 .. 8 characters>	Freely selectable name of the time zone at the location of the control (no digits, no "+", no "-"). Names with only one or two characters are not applied when entering.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	If either the parameter "SummerLoc" or "SummerDiffUtc" contain an entry, the "SummerZoneName" must also have a valid entry. If the parameter "SummerDiffUtc" is empty, the parameter "SummerLoc" is automatically deleted when deleting "SummerZoneName".	
Macoda ID	9050 00001 [4]	
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility	
	<i>Example:</i>	
	Time zone at the location of the controller:	
	PST (Pacific Standard Time)	

Location

Location (summer time)		SummerLoc
Specifies the location of the controller in relation to longitude "0" (Greenwich) during the summer period.		
Path	NCO/DateTime/Summer/SummerLoc	
Value range	{blank}	not specified

NC optimization (NCO) - Basic parameters

	+	west of Greenwich
	-	east of Greenwich
Pre-assignment		
	Status upon delivery:	{empty} (no specification)
	Setting on default value in the NC configurator:	- (east of Greenwich)
Unit	-	
Description of parameter values	+	west of Greenwich Location of the controller is to the west of longitude 0 (Greenwich).
	-	east of Greenwich Location of the controller is to the east of longitude 0 (Greenwich).
	{blank}	not specified The entry remains empty if there is no switching between winter and summer time.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	The "SummerLoc" entry also requires entries in the parameters "SummerZoneName" and "SummerDiffUtc". "SummerLoc" is automatically deleted if the parameters "SummerZoneName" and "SummerDiffUtc" are deleted.	
Macoda ID	9050 00001 [5]	
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility	

Time difference UTC > Local time

Time difference UTC > local time (summer time)		SummerDiffUtc
	Specifies by how much the time differs between UTC time and local time during the summer period at the MTX location.	
Path	NCO/DateTime/Summer/SummerDiffUtc	
Value range	<8 characters>	
Pre-assignment	{blank}	
Unit	-	
Description of parameter values	Input of the time difference in the format "hh:mm:ss" (hours:minutes:seconds). Leading zeroes, minutes and seconds are optional and can be left out.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	The "SummerDiffUtc" entry also requires entries in the parameters "SummerZoneName" and "SummerLoc".	

If the parameter "SummerZoneName" is empty, the parameter "SummerLoc" is automatically deleted when deleting "SummerDiffUtc".

Macoda ID 9050 00001 [6]
Data type Str8_t (isoLatin1String)
 8 byte string for MACODA compatibility

16.9.4 Clock change to winter time

Date

Date (change to winter time)		SwiWinterDate
Specification of the date to change to winter time.		
Path	NCO/DateTime/SwiWinter/SwiWinterDate	
Value range	{blank} <nnn> J<nnn> M<mm>.<w>.<d>	
Pre-assignment	{blank}	
Unit	-	
Description of parameter values	{blank} Winter time starts on the last Sunday in October. At 03:00:00 local summer time, the clock will be set back by 1 hour to 02:00:00 local time. <nnn> Date of conversion (excluding Feb. 29!). The following applies: 0 <= <nnn> <= 364 Example: 303 equates to conversion on October 30 J<nnn> Date of conversion (excluding Feb. 29!). The following applies: 0 <= <nnn> <= 365 Example: Y304 equates to conversion on October 30 M<mm>.<w>.<d> Date of conversion The following applies: <mm> is the month (1-12) <w> is the week (1-5) in month (5 = last week in the month) <d> is the day of the week (0-6) (0 = Sunday). Example: M10.5.0 equates to conversion on the last Sunday of October	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	9050 00001 [9]	
Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility	

NC optimization (NCO) - Basic parameters

Time

Time (change to winter time)		SwiWinterTime
Local summer time when the change from summer to winter time is to take place.		
Path	NCO/DateTime/SwiWinter/SwiWinterTime	
Value range	<8 characters>	
Pre-assignment	{blank}	
Unit	-	
Description of parameter values	{blank} Equals 03:00:00. This means, at 03:00:00 local summer time, the clock will be set back by 1 hour to 02:00:00 local time. <8 characters> Input of time of clock change in the format "hh:mm:ss" (hours:minutes:seconds). Leading zeroes, minutes and seconds are optional and can be left out.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	9050 00001 [10]	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	
	<i>Example:</i>	
	PST (Pacific Standard Time) with summer to winter change (e.g. Los Angeles).	
	PST	Name of time zone (WinterName)
	west of Greenwich	IndraMotion MTX location (WinterLoc)
	7	Time difference UTC > local time (WinterDiffUtc)
	M4.1.0	Clock change to winter time (SwiWinterDate)
	3	Time of clock change (SwiWinterTime)

16.9.5 Clock Change to Summer Time

Date

Date (change to summer time)		SwiSummerDate
Specification of the date to change to summer time.		
Path	NCO/DateTime/SwiSummer/SwiSummerDate	
Value range	{blank} <nnn> J<nnn> M<mm>.<w>.<d>	
Pre-assignment	{blank}	

	Unit	-
Description of parameter values	{blank}	Summer time begins on the last Sunday in March. The clock is advanced by 1 hour at 02:00:00 local time to 03:00:00.
	<nnn>	Date of conversion (excluding Feb. 29!). The following applies: 0 <= <nnn> <= 364 Example: 59 equates to conversion on March 1
	J<nnn>	Date of conversion (excluding Feb. 29!). The following applies: 0 <= <nnn> <= 365 Example: J60 equates to conversion on March 1
	M<mm>.<w>.<d>	Date of conversion The following applies: <mm> is the month (1-12) <w> is the week (1-5) of the month. (5 = last week of the month) <d> is the day of the week (0-6). (0 = Sunday). Example: M3.5.0 equates to conversion on the last Sunday of March
	Access rights	MTB: Write Setter: Read Operator: Read
	Activation	System restart (CS)
	Dependency	-
	Macoda ID	9050 00001 [7]
	Data type	Str8_t (isoLatin1String) 8 byte string for MACODA compatibility

Time

		<table><tr><td>Time (change to summer time)</td><td>SwiSummerTime</td></tr></table>	Time (change to summer time)	SwiSummerTime
Time (change to summer time)	SwiSummerTime			
		Local winter time when the change from winter to summer time is to take place.		
	Path	NCO/DateTime/SwiSummer/SwiSummerTime		
	Value range	<8 characters> {blank}		
	Pre-assignment	{blank}		
	Unit	-		
Description of parameter values	{blank}	Equals 02:00:00. Corresponds to 02:00:00. That means, at 02:00:00 local time, the clock is set 1 hour ahead to 03:00:00 local time.		
	<8 characters>	Input of time of clock change in the format "hh:mm:ss" (hours:minutes:seconds).		

NC optimization (NCO) - Basic parameters

		Leading zeroes, minutes and seconds are optional and can be left out.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	9050 00001 [8]	
Data type	Str8_t (isoLatin1String)	
	8 byte string for MACODA compatibility	
	<i>Example:</i>	
	Settings for CET with clock change for winter/summer (e.g. Germany):	
	CET	Name of time zone (SummerZoneName)
	east of Greenwich	MTX location (SummerLoc)
	1	Time difference UTC > local time (SummerDiffUtc)
	M3.5.0	Clock change to summer (SwiSummerDate)
	2	Time of clock change (SwiSummerTime)

16.10 Accuracy specification for circular programming (CircleAccur)

16.10.1 General information

Accuracy specification for circular programming concerns

- Center point programming
- Radius programming

16.10.2 Center point programming

Tolerance from which a runtime error is generated

Tolerance from which a runtime error is generated	RadTolCentProg

Specifies for each channel Ch[k] from when the internal controller center point correction is disabled.

Maximum permitted difference between the center point–starting point radius and the center point–end point radius. If radius differences occur that are below the specified threshold, the control corrects the center point coordinates automatically. Radius differences greater than the threshold value generate a runtime error.

Path	NCO/CircleAccur/Ch[k]/CentProg/RadTolCentProg
Value range	0 .. 100000 0 .. 99999 for MACODA compatibility
Pre-assignment	10000
Unit	0.1 µm
Description of parameter values	Limit from which no internal center point correction will take effect
Access rights	MTB: Write Setter: Read

	Operator:	Read
Activation		System reset (SR)
Dependency		-
Macoda ID		7050 00020
Data type		Double0_100T_t (double)
		4 byte float for MACODA compatibility

16.10.3 Radius programming

Tolerance from which a runtime error is generated

	Tolerance from which a runtime error is generated	RadTolRadProg
	Defines for each channel Ch[k] up to what deviation inaccurate radius entries will be corrected with circular interpolation without generating a runtime error. This allows, for instance, compensation for inaccurate computations in data preprocessing.	
Path	NCO/CircleAccur/Ch[k]/RadProg/RadTolRadProg	
Value range	0 .. 100000 0 .. 99999 for MACODA compatibility	
Pre-assignment	100	
Unit	0.1 µm	
Description of parameter values	Amount of radius correction with circular interpolation.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00030	
Data type	Double0_100T_t (double) 4 byte float for MACODA compatibility	

16.11 Scale factors (VelScaleFact)

16.11.1 Channel-specific scaling factors for feed velocities and feed/revolution

Scaling Factor for Feed Programming

	Scaling factor for feed programming	FactVel
	Specifies for each channel Ch[k] the units in which the "feed rate" is programmed.	
Path	NCO/VelScaleFact/Ch[k]/FactVel	
Value range	1 (mm/min (inch/min)) 60 (60 mm/s (60 inch/s)) 1,000 (1000 m/min (1000 inch/min)) 60000 60000 m/s (60000 inch/s)	
Pre-assignment	(mm/min (inch/min))	
Unit	-	

NC optimization (NCO) - Basic parameters

Description of parameter values	Units of feed rate:	
	1	Units in 1 mm/min or 1 inch/min
	60	Units in 60 mm/s or 60 inch/s
	1,000	Units in 1000 m/min or 1000 inch/min
	60000	Units in 60000 m/s or 60000 inch/s
	 Inch units are used when working with inch programming (G70).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7040 00010	
Data type	double	
	4 byte float for MACODA compatibility	

Scaling factor for feed/revolution

Scaling factor for feed per revolution		FactVelPerRev
Specifies for each channel Ch[k] the units in which the "feed per revolution" should be programmed.		
Path	NCO/VelScaleFact/Ch[k]/FactVelPerRev	
Value range	0.001	0.001 mm (0.001 inch)
	0.01	(0.01 mm (0.01 Inch))
	1.000	(1 mm (1 Inch))
	1000.000	1000 mm (1000 inch)
Preassignment	1.000	(1 mm (1 Inch))
Unit	-	
Description of parameter values	Unit for feed per revolution:	
	0.001	Unit in 0.001 mm or 0.001 inch
	0.01	Unit in 0.01 mm or 0.01 inch
	1.000	Unit in 1 mm or 1 inch
	1000.000	Unit in 1000 mm or 1000 inch
	 Inch units are used when working with inch programming (G70).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7040 00020	
Data type	double, 3 decimal places	
	8 byte float, 3 decimal places for MACODA compatibility	

16.11.2 Channel-specific scaling factors for rotary axis velocities

General information

Various scaling factors for metric and non-metric rotary axis velocities are set here.

Rotary axis velocities for rotary and endless axes are based on whether the behavior of the controller is that of a typical **European** or **American** controller. In this case, it is set separately for "inch programming" (G70) and for "metric programming" (G71) whether the velocity calculation should be made with the assignment "1 degree = 1 mm" or "1 degree = 1 inch".

The settings are made in the parameters "FactRotInch" and "FactRotMetr" in the folder NCO/VelScaleFact/Ch[k]/.

European setting:

Specification:

Scaling factor for non-metric rotary axis velocities

(FactRotMetr: 1 degree = 1 mm)

Scaling factor for metric rotary axis velocities

(FactRotMetr: 1 degree = 1 mm)

In this case, note the following:

- Both with metric and with inch programming, one degree always corresponds to 1 mm.
- If the rotary axis is alone programmed for G71, the feed is interpreted in degree/min.
- In the case of G70 and only rotary axis programming, the rotary axis will travel too rapidly by a factor of 25.4 (feed rate value interpreted in degrees/min).

Example:

$F100 = 100 \text{ inch/min} = 2540 \text{ mm/min} = 2540 \text{ degrees/min}$

- If the rotary axis and linear axis move together, both are scaled equally at G71.

Example:

The linear axis moves by 100 mm, the rotary axis by 100 degrees; both these axes move at a velocity of 0.7071 times the programmed feed rate in [mm/min] or [degrees/min] respectively, because the distances to be traveled by both of these axes are interpreted to be of equal length.

- If the linear axis travels by 100 inch and the rotary axis by 100 degree for G70, the linear axis travels at almost 100% of the programmed velocity, since the distance traveled by the linear axis is assumed to be 25.4 times longer than the distance to be traveled by the rotary axis.

American setting:

Specification:

Scaling factor for non-metric rotary axis velocities

(FactRotInch: 1 degree = 1 inch)

Scaling factor for metric rotary axis velocities

(FactRotMetr: 1 degree = 1 inch)

In this case, note the following:

- For metric as well as inch programming, one degree always corresponds to one inch.
- If only the rotary axis is programmed in G70, the feed is interpreted in degree/min.

NC optimization (NCO) - Basic parameters

- With G71 and exclusive rotary axis programming, the rotary axis travels to slowly by a factor of 25.4
(feed value not interpreted in degrees/min.)
Example:
 $F100 = 100 \text{ mm/min} = 3.937 \text{ inch/min} = 3.397 \text{ degree/min}$.
- If the rotary axis and linear axis move together, both are scaled equally at G70.
Example:
The linear axis moves by 100 inch, the rotary axis by 100 degree. Both these axes move with 0.7071* programmed feed in [inch/min] or [degree/min], since both distances to be traveled are assumed to be of equal length.
- If the linear axis travels by 100 mm and the rotary axis by 100 degree in G71, the rotary axis travels at almost 100% of the programmed velocity, since the distance traveled by the rotary axis is assumed to be 25.4 times longer than the distance to be traveled by the linear axis.

General setting:

Specification:

Scaling factor for non-metric rotary axis velocities

(FactRotInch: 1 degree = 1 inch)

Scaling factor for metric rotary axis velocities

(FactRotMetr: 1 degree = 1 mm)

In this case, note the following:

- For G70 and for G71, the feed is interpreted in degree/min if only the rotary axis is programmed.
- With G70 and G71, the rotary and the linear axes are scaled equally if they travel jointly,.

Example:

G70:

The linear axis travels by 100 inches, the rotary axis travels by 100 degrees; both these axis travel at 0.7071* programmed feed in [inch/min] or [degrees/min].

Example:

G71:

The linear axis travels by 100 mm, the rotary axis travels by 100 degrees; both these axis travel at 0.7071* programmed feed in [mm/min] or [degrees/min].



*The following setting does **not** make any sense physically:*

- Scaling factor for non-metric rotary axis velocities
(FactRotInch: 1 degree = 1 mm)
- Scaling factor for metric rotary axis velocities
(FactRotMetr: 1 degree = 1 inch)

Scaling factor for non-metric rotary axis velocities

Scaling factor for non-metric Rotary axis velocities	FactRotInch

	Defines for each channel Ch[k] whether a rotary axis corresponds to "1 degree = 1 mm", or "1 degree = 1 Inch" in the velocity calculation under G70 (inch programming).	
Path	NCO/VelScaleFact/Ch[k]/FactRotInch	
Value range	1	(1 degree = 1 mm)
	2	(1 degree = 1 inch)
Pre-assignment	1	(1 degree = 1 mm)
Unit	-	
Description of parameter values	1	(1 degree = 1 mm) The feed rate for a non-metric rotary axis equates to 1 degree/min $\Rightarrow$ 1 mm/min. Example: F100 $\rightarrow$ 100 mm/min = 100 degrees/min (feed of rotary axis in degree/min).
	2	(1 degree = 1 inch) The feed rate for a non-metric rotary axis corresponds to 1 degree/min $\Rightarrow$ 1 inch/min. Example: F100 $\rightarrow$ 100 inch/min = 100 degrees/min (feed of rotary axis in degree/min).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7040 00110 [1]	
Data type	Byte1_2_t (byte) 1 byte integer, unsigned for MACODA compatibility	

Scaling factor for metric rotary axis factors

Scaling factor for metric rotary axis velocities		FactRotMetr
	Defines for each channel Ch[k] whether a rotary axis corresponds to "1 degree = 1 mm", or "1 degree = 1 Inch" in the velocity calculation under G71 (metric programming).	
Path	NCO/VelScaleFact/Ch[k]/FactRotMetr	
Value range	1	(1 degree = 1 mm)
	2	(1 degree = 1 inch)
Pre-assignment	1	(1 degree = 1 mm)
Unit	-	
Description of parameter values	1	(1 degree = 1 mm) The feed rate for a metric rotary axis equates to 1 degree/min $\Rightarrow$ 1 mm/min. Example: F100 $\rightarrow$ 100 mm/min = 100 degree/min (feed of rotary axis in degree/min)

NC optimization (NCO) - Basic parameters

	2	(1 degree = 1 inch) The feed rate for a metric rotary axis corresponds to 1 degree/min $\Rightarrow$ 1 inch/min. Example: F100 $\rightarrow$ 100 inch/min = 100 degrees/min (feed of rotary axis in degree/min).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7040 00110 [2]	
Data type	Byte1_2_t (byte) 1 byte integer, unsigned for MACODA compatibility	

16.12 Precision programming (PrecProg)

16.12.1 Contour deviation tolerance for "PrecProg" (PRP)

Contour deviation tolerance for "PrecProg" (PRP)		ContDevTol
Specifies for precision programming "PrecProg" (PRP) the maximum permissible path error at the contour transition and radius error at the circular arc.		
 The value that is currently active can be modified temporarily by programming.		
Path	NCO/PrecProg/ContDevTol	
Value range	0.000 .. 100.000 0.0001 .. 99.9999 for MACODA compatibility	
Pre-assignment	0.02	
Unit	mm	
Description of parameter values	Maximum deviation at the contour transition or maximum radius error at the circular arc.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	8003 00001	
Data type	Double0_100_t (double, 3 decimal places) 8 byte float, 4 decimal places for MACODA compatibility	

16.13 Block transfer inhibit (BlkTransferLock)

16.13.1 Standstill type at block end for block transfer inhibit

Standstill type at block end for block transfer inhibit	StandstModeBlk-TransferLock

Specifies the behavior of the block transfer inhibit at block end.

If the block end is reached when the block transfer inhibit is active and if the path velocity is not 0, this parameter determines how the standstill is performed.



When decelerating to standstill, there has to be sufficient distance for deceleration after block end!

Path	NCO/BlkTransferLock/StandstModeBlkTransferLock	
Value range	0	Velocity Jump
	1	Decelerating
Pre-assignment	0	Velocity Jump
Unit	-	
Description of parameter values	0	Velocity Jump
		Setting to standstill with velocity jump
	1	Decelerating
		Setting to standstill by deceleration
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	8011 00001	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

16.14 System units (CorrUnit)

16.14.1 Units for tables and corrections

Units for tables and corrections	SysCorrUnit

Defines

- the units used by the PLC for specification of external correction values. These values must be entered to five decimal places in the unit of measurement specified here.

Example:

The selected unit is mm. If an external correction value is to be set to 5 mm, specify the value 50000 in the PLC.

- the units for storing table values internally. This parameter is analyzed when a table is created and is then stored in the table.

Irrespective of the units set, correction values may be entered into the dialog in mm or inch.

NC optimization (NCO) - Basic parameters

Path	NCO/CorrUnit/SysCorrUnit	
Value range	0	(mm)
	2	(inch)
Pre-assignment	0	(mm)
Unit	-	
Description of parameter values	0	(mm)
	Units for external correction values from the PLC: 0.0001 mm	
	Internal units for tables: mm	
	2	(inch)
	Units for external correction values from the PLC: 0.0001 inches	
	Internal units for tables: inch	
	MTB:	Write
	Setter:	Read
Access rights	Operator:	Read
	Activation	System restart (CS)
Dependency	-	
Macoda ID	9020 00010	
Data type	Byte0_4_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

16.14.2 Default measuring unit for asynchronous linear axes

Default measuring unit for asynchronous linear axes	AsynchrAxUnit

Indicates in which unit the position and velocity specification for asynchronous linear axes are programmed and interpreted.



The velocity for asynchronous axes is programmed via address FA.

The following points have to be observed:

- This parameter has also a meaning for NC block specifications which are specified from PLC for asynchronous axes.
- If an asynchronous axis is applied from a channel (GAX, WAX), the unit according to the current channel status (G70/G71) is applied from now on for this axis.
- If a channel axis is transferred from a channel (RAX), the unit according to the setting is applied for this parameter.
- The unit for asynchronous axes can be programmed explicitly via programming attributes **INCH** and **MM** independently from this parameter settings. This also applies to the velocity, e.g. VA=INCH(120) FA=INCH(100) .
- The unit of linear axes is selected axis-wise via interface signal **qAx_Jo-gInch** in the setup mode.

Path	NCO/CorrUnit/AsynchrAxUnit	
Value range	0	mm (mm/min)
	1	inch (inch/min)

Pre-assignment	0	mm (mm/min)
Unit	-	
Description of parameter values	0	mm (mm/min) Unit for positions: mm Unit for velocities: mm/min
	1	inch (inch/min) Unit for positions: inch Unit for velocities: inch/min
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	9020 00020	
Data type	Byte0_1_t (byte) 1 byte integer, unsigned for MACODA compatibility	

16.15 Database tables (DbTables)

16.15.1 Buffer size for change event of the DB tables

Buffer size for change event of the DB tables		DbTabEventBuffSize
This parameter determines the number of events that can be buffered in each change buffer.		
MTX user interfaces (e.g. IndraWorks Operation) are informed about changes in the database tables in order to be able to visualize the current state. Due to performance reasons, every single change is identified by the user interface. In case of a fast sequence of writing operations by the PLC or CPL, the user interface can only be informed completely if the change events are buffered in the MTX.		
Path	NCO/DbTables/DbTabEventBuffSize	
Value range	1 .. 64	
Pre-assignment	8	
Unit	-	
Description of parameter values	Number of buffered events of each change event	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	9050 00010	
Data type	Byte1_64_t (byte) 1 byte integer, unsigned for MACODA compatibility	

16.16 Communication server (CommServ)

16.16.1 Enabling communication server

Enable communication server		EnablCommServ
Determines if the communication server is to be enabled at system start.		
Path	NCO/CommServ/EnablCommServ	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No
		Communication server is not enabled
	1	Yes
		Communication server enabled
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4090 00100	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

16.17 Web server (WebServ)

16.17.1 Enable Web server

Enable web server		EnablWebServ
Determines if the web server is to be enabled at system start.		
Path	NCO/WebServ/EnablWebServ	
Value range	0	No
	1	Yes
Pre-assignment	1	Yes
Unit	-	
Description of parameter values	0	No
		Web server is not enabled
	1	Yes
		Web server is enabled
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	4090 00001	

Data type Byte0_1_t (byte)
1 byte integer, unsigned for MACODA compatibility

17 Operating functions (OPF) - Basic parameters

17.1 General information

The Operating Functions (OPF) data group allows the following settings:

- [Returning to contour, page 319](#)
- [Re-entry, page 321](#)
- [Handwheel functionality, page 322](#)
- [Jog axes in workpiece coordinates, page 323](#)

17.2 Return to path (ReturnPath)

17.2.1 Return strategy

Return strategy		ReturnMode
Specifies for each channel Ch[k] the return strategy for "return to path" mode.		
Path	OPF/ReturnPath/Ch[k]/ReturnMode	
Value range	1	Automatic
	2	Single block
	3	Manual mode
Pre-assignment	1	Automatic
Unit	-	
Description of parameter values	1	Automatic
		If the channel interface of the input signal "Ready to return to contour" (iCh_ReadyReEnter) is set, the return movement is started upon "NC start" and the output signal "Returning to contour active" (iCh_ReEnterAct) is output. The movement then stops at the placement point. The placement point is the position reached upon completion of the first return movement. The NC state then sets to "Ready" and the output signal "Return" (qCh_RetCont) is set. Upon the next "Cycle Start", machining is resumed. With "Cycle Start" and the reset signal "Ready to return to contour" (iCh_ReadyReEnter), machining can be continued even without automatic returning.
	2	Single block
		If the channel interface of the input signal "Ready to return to contour" (iCh_ReadyReEnter) is set, the return movement is started upon "NC start" and the output signal "Returning to contour active" (iCh_ReEnterAct) is output. At the end of each movement, the NC state changes to "Ready" and the "Ready to return to path" (qCh_RetCont) interface output signal is set. Press "NC start" multiple times and the placement point can be freely selected. With "Cycle Start" without the input signal "Ready to return to contour" (iCh_ReadyReEnter), machining is continued.
	3	Manual mode
		The placement point is approached in jog steps. At the end of each movement, the NC state changes to "Ready" and the

Operating functions (OPF) - Basic parameters

		"Return" (qCh_RetCont) interface output signal is set. Machining is resumed upon the next "Cycle Start".
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7060 00310	
Data type	Byte1_3_t (byte)	
	4 byte integer for MACODA compatibility	

17.2.2 Return point

Approach point		ReturnPoint
Specifies for each channel Ch[k] the approaching point for "return to path" mode.		
If the tool is on the placement point, the controller moves to the approaching point the next time that the "NC is started". Any modified corrections will be included.		
Path	OPF/ReturnPath/Ch[k]/ReturnPoint	
Value range	1	Starting point
	2	End point
	3	Point of interruption
Pre-assignment	1	Starting point
Unit	-	
Description of parameter values	1	Approaching point = Starting point
	2	Approaching point = End point
	3	Approaching point = Interruption point
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7060 00320	
Data type	Byte1_3_t (byte)	
	4 byte integer for MACODA compatibility	

17.2.3 Maximum number of jog movements to be recorded

Maximum number of jog movements to be recorded		NofJogSteps
Specifies for each channel Ch[k] how many jog steps will be recorded when leaving the contour in "return to path" mode.		
A softkey is available for terminating the recording process prematurely.		
Path	OPF/ReturnPath/Ch[k]/NofJogSteps	
Value range	0 .. 999	
Pre-assignment	20	

Unit	-
Description of parameter values	Maximum number of jog movements to be recorded
Access rights	MTB: Write
	Setter: Read
	Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7060 00330
Data type	Int0_999_t (int)
	4 byte integer for MACODA compatibility

17.3 Re-entry (ReEntry)

17.3.1 Function mode for re-entry

Function mode for re-entry data		ReEntryFuncMode
	Specifies the mode of the "Re-entry" functionality.	
Path	OPF/ReEntry/ReEntryFuncMode	
Value range	0	Simple re-entry
	1	Extended re-entry
Pre-assignment	0	Simple re-entry
Unit	-	
Description of parameter values	0	Simple re-entry
		In this mode, re-entry marks are only stored channel-specifically in the directory specified using the "ReEntryDataPath" parameter.
	1	Extended re-entry
		In this mode, re-entry marks are stored channel-specifically as well as program-specifically in the directory specified using the "ReEntryDataPath" parameter.
		The RCBE command generated additional information for re-entry marks (program stack, breakpoint, file information) and stores this information.
Access rights	MTB: Write	
	Setter: Read	
	Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	3080 00300	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

17.3.2 Directory for re-entry data

Directory for re-entry data	ReEntryDataPath
Defines the directory of the MTX where the data of the "Re-entry" functionality is stored.	

Operating functions (OPF) - Basic parameters

Path	OPF/ReEntry/ReEntryDataPath
Value range	{Text}
Pre-assignment	/usr/bosch/SRun/
Unit	-
Description of parameter values	The search path of the directory for the re-entry data is made up from the following directory names. <i>The following applies:</i> • The search path must begin and end with the character "/". • A "/" has to be written as well between the directory names.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	3080 00301
Data type	Str80_t (isoLatin1String) 10 * 8 byte string for MACODA compatibility

17.4 Handwheel functionality (Handw)

17.4.1 General information

You can make handwheel functionality settings for 2 handwheels Wh[1..2].

17.4.2 Number of increments/revolution

Number of increments/revolution		IncrPerRev
Specifies the number of increments per revolution of a connected handwheel Wh[1..2].		
Path	OPF/Handw/Wh[1..2]/IncrPerRev	
Value range	Integer range	
Pre-assignment	100	
Unit	-	
Description of parameter values	Number of increments per revolution	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	9060 00001	
Data type	int 4 byte integer for MACODA compatibility	

Example:

Handwheel evaluation

The machine control panel "VAM.." with Profibus DP quadruples the number of increments of the connected handwheel. If a handwheel with 100 increments per revolution is used, the parameter has to be set to 400.

17.5 Jogging axes in workpiece coordinates (WcsJog)

17.5.1 General information

Under "Jogging axes in workpiece coordinates", various different parameters (e.g. scaling, jog distance etc) and velocity data (WCS jog interface) can be specified for jogging.

17.5.2 Channel-specific data

Scaling of jog increment

Scaling of jog increment		IncrWcsJog
	Specifies for each channel Ch[k] the size of an increment.	
Path	OPF/WcsJog/Ch[k]/IncrWcsJog	
Value range	0.000 .. 100.000 0.001 .. 100.000 for MACODA compatibility	
Pre-assignment	1.000	
Unit	µm or 0.001°	
Description of parameter values	Increment size	
	 In the channel-specific "inch" state, the set parameter is multiplied by the factor 2.54 for a linear coordinate.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 01020	
Data type	Float0_100_t (float, 3 decimal places) 4 byte float, 3 decimal places for MACODA compatibility	

Variable jog distance

Variable jog distance		DistWcsJog
	Specifies for each channel Ch[k] how many increments shall be moved after pressing the jog button in travel step "variable steps".	
Path	OPF/WcsJog/Ch[k]/DistWcsJog	
Value range	0.000 .. 1000000.000 1 .. 100000 for MACODA compatibility	
Pre-assignment	2000.000	
Unit	incr	

Operating functions (OPF) - Basic parameters

Description of parameter values	Jog distance (increments)
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7050 01030
Data type	Dist_t (double, 3 decimal places) 4 byte integer, unsigned for MACODA compatibility

Program name for the default configuration

Program name for default configuration		TrafoProgName
	Specifies for each channel Ch[k] the name of the part program to be executed once when jogging in workpiece coordinates.	
Path	OPF/WcsJog/Ch[k]/TrafoProgName	
Value range	{Text}	
Pre-assignment	-	
Unit	-	
Description of parameter values	Name of a part program	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 01100	
Data type	Str32_t (isoLatin1String) 8 byte string for MACODA compatibility	
	<i>Example:</i>	
	Program name for jogging in workpiece coordinates	
	The part program named "InitstringForJogWCS" activates an axis transformation and/or a coordinate transformation.	
	 The program "InitstringForJogWCS" must be in the search path in order to be executed upon the first jog step.	

Program name for the coordinate selection

Program name for coordinate selection		CoordProgName
	Specifies for each channel Ch[k] the name of the part program to be executed when jogging in workpiece coordinates.	
	The program is executed upon each jog step.	
Path	OPF/WcsJog/Ch[k]/CoordProgName	
Value range	{Text}	
Pre-assignment	-	

Unit	-
Description of parameter values	Name of a part program
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7050 01110
Data type	Str32_t (isoLatin1String) 8 byte string for MACODA compatibility <i>Example:</i> program name for coordinate selection The part program named "CoordSelectForJogWCS" activates the coordinate selection. Content of the part program "CoordSelectForJogWCS": 10 COORDNO%=PLC(<Type>,<DB number>,<Address>,<Size>) N10 JogWCSSelect JWSCOORD[COORDNO%]  The program "CoordSelectForJogWCS", which must be in the search path, is executed upon each jog step.

17.5.3 Velocity data for the WCS jog interface

Slow

Slow	SlowVelWcs
	Specifies jog velocity "Slow" in workpiece coordinates (WCS).
Path	OPF/WcsJog/Vel/WcsJog/SlowVelWcs
Value range	0.000 .. 1000000000.000 0.000 .. 36000000.000 for MACODA compatibility
Pre-assignment	100.000
Unit	mm/min or °/min
Description of parameter values	Jog velocity "Slow"
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	7050 01000 [1]
Data type	Vel_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

Medium

Medium	MedVelWcs
	Specifies jog velocity "Medium" in workpiece coordinates (WCS).

Operating functions (OPF) - Basic parameters

Path	OPF/WcsJog/Vel/WcsJog/MedVelWcs
Value range	0.000 .. 1000000000.000 0.000 .. 36000000.000 for MACODA compatibility
Pre-assignment	100.000
Unit	mm/min or °/min
Description of parameter values	Jog velocity "Medium"
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	7050 01000 [2]
Data type	Vel_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

Fast

Fast		FastVelWcs
	Specifies jog velocity "Fast" in workpiece coordinates (WCS).	
Path	OPF/WcsJog/Vel/WcsJog/FastVelWcs	
Value range	0.000 .. 1000000000.000 0.000 .. 36000000.000 for MACODA compatibility	
Pre-assignment	1000.000	
Unit	mm/min or °/min	
Description of parameter values	Jog velocity "Fast"	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 01000 [3]	
Data type	Vel_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

Rapid traverse

Rapid traverse		RapidVelWcs
	Specifies jog velocity "Rapid traverse" in workpiece coordinates (WCS).	
Path	OPF/WcsJog/Vel/WcsJog/RapidVelWcs	
Value range	0.000 .. 1000000000.000 0.000 .. 36000000.000 for MACODA compatibility	
Pre-assignment	2000.000	
Unit	mm/min or °/min	
Description of parameter values	Jog velocity "Rapid traverse"	
Access rights	MTB: Write	

	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 01000 [4]	
Data type	Vel_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

Defined step width

Defined step width		DefStepVelWcs
	Specifies jog velocity "Defined step width" (incremental jogging) in workpiece coordinates (WCS).	
Path	OPF/WcsJog/Vel/WcsJog/DefStepVelWcs	
Value range	0.000 .. 1000000000.000	
	0.000 .. 36000000.000 for MACODA compatibility	
Pre-assignment	100.000	
Unit	mm/min or °/min	
Description of parameter values	Jog velocity "Defined step width"	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 01000 [5]	
Data type	Vel_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

Variable step width

Variable step width		VarStepVelWcs
	Specifies jog velocity "Variable step width" (incremental jogging) in workpiece coordinates (WCS).	
Path	OPF/WcsJog/Vel/WcsJog/VarStepVelWcs	
Value range	0.000 .. 1000000000.000	
	0.000 .. 36000000.000 for MACODA compatibility	
Pre-assignment	500.000	
Unit	mm/min or °/min	
Description of parameter values	Jog velocity "Variable step width"	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 01000 [6]	

Data type

Vel_t (double, 3 decimal places)
8 byte float, 3 decimal places for MACODA compatibility

System axis number of the interface for jogging in the WCS

	System axis no. of interface for jogging in WCS	WcsJogIf
	Specifies a system axis with the axis interface "Manual feed" (qAx_ManFeed_00 to qAx_ManFeed_03), used to derive the selection of the feed (slow, medium, fast, rapid traverse) or the number of increments (1, 10, 100, 1000, x) (see OPF/WcsJog/Vel/WcsJog/...VelWcs).	
Path	OPF/WcsJog/Vel/WcsJogIf	
Value range	0 .. 250	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Feed or number of increments is not derived from the axis interface. In this case, a feed value has to be specified at the same time a "coordinate" is selected, and a step width when incremental jogging is performed.
	>0	Number i of the system axis Dr[i].
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 01010	
Data type	Dr_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

18 Safety functions (SPF) - Basic parameters

18.1 Overview

You can select the following settings in the data group *Safety functions (SPF)*:

- [Waypoints, page 329](#)
- [Range monitoring, page 332](#)
- [Distance monitoring to a programmed endpoint, page 335](#)
- [One-dimensional collision monitoring, page 336](#)

18.2 Waypoints (WayPoint)

18.2.1 General information

The MTX can monitor up to 64 positions (waypoints) throughout the system, with a **maximum of 8** waypoints that can be assigned to each axis. As long as the current actual position of an axis (corrected with compensation values, e.g. LSEC) is greater than or equal to the monitored position, an appropriate bit is set within the signal range iAx_PosSwitch1 to iAx_PosSwitch8 (waypoint 1-8) on the axis interface.



Monitor by linking the respective bits of an axis via the PLC.

For rotary axes, you can also specify whether a waypoint is to be monitored in respect of the whole traversing range of the axis or in respect to its range between 0° and 360°.

For each monitoring point Point[1..64], you can set the following parameters.

18.2.2 Waypoint point[1..64]

System axis number

	System axis number	DrIndWayPoint
	Assigns a system axis index i of the system axis Dr[i] to the respective waypoint Point[1..64].	
Path	SFU/WayPoint/Point[1..64]/DrIndWayPoint	
Value range	0 .. 250	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Monitoring position is not used.
	>0	System axis index of the axis that shall be assigned to the waypoint Point[1..64].



A single axis may have a maximum of 8 waypoints assigned to it.

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	

Safety functions (SPF) - Basic parameters

Macoda ID 2010 00100

Data type Dr_t (byte)

2 byte integer, with sign for MACODA compatibility (value = axis index, value < 0: always monitor axis, values > 0: only monitor axis if reference)

Position

	Position	PosWayPoint
	Defines the position value of one of the 64 possible waypoints Point[1..64].	
Path	SFU/WayPoint/Point[p]/PosWayPoint	
Value range	-1000000.000 .. 1000000.000 -999999.9999 .. 999999.9999 for MACODA compatibility	
Pre-assignment	0.000 0.0000 for MACODA compatibility	
Unit	mm or ° (degrees)	
Description of parameter values	Position value of a waypoint	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2010 00110	
Data type	Pos_t (double, 3 decimal places) 8 byte float, 4 decimal places for MACODA compatibility	

Monitoring type

	Type of monitoring	MonTypeWayPoint
Path	SFU/WayPoint/Point[1..64]/MonTypeWayPoint	
Value range	1	always monitor axis
	2	only monitor axis if reference
Pre-assignment	1	always monitor axis
Unit	-	
Description of parameter values	1	The Position Point[1..64] of the assigned axis (in DrIndWayPoint) is always monitored
	2	The Position Point[1..64] of the assigned axis (in DrIndWayPoint) is only monitored when the reference point of the axis has already been reached.
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2010 00100	
Data type	Byte0_2_t (byte)	

2 byte integer, with sign for MACODA compatibility (value = axis index, value < 0: always monitor axis, values > 0: only monitor axis if reference)

Monitoring for rotary axes

Monitoring for rotary axes	MonRotWayPoint

Defines for each of the 64 possible waypoints Point[1..64] whether the waypoint for a rotary axis is to be monitored in respect of the whole traversing range of the axis or in respect of its range between 0° and 360°.



The parameter is only important for axes of axis movement type "rotary axis"

(see "SysAxType" in folder MAIN/Dr[i]/AxFunc/).

Path	SFU/WayPoint/Point[1..64]/MonRotWayPoint	
Value range	0	entire traversing range
	1	within the modulo range
Pre-assignment	0	entire traversing range
Unit	-	
Description of parameter values	0	Monitoring is "absolute". The configured position to be monitored may be anywhere in the entire traversing range of the rotary axis.
	1	Monitoring is "cyclic". The configured position to be monitored may be anywhere in the range 0° through 360°.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2010 00130	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Interface signal of the point

Interface signal of the point	IfSigWayPoint

Defines for each of the 64 possible waypoints Point[1..64] which interface signal of the respective axis interface serves as interface signal. The PLC input signal to the respective NC output signal is given.

Path	SFU/WayPoint/Point[p]/IfSigWayPoint	
Value range	0	iAx_PosSwitch1
	1	iAx_PosSwitch2
	2	iAx_PosSwitch3
	3	iAx_PosSwitch4
	4	iAx_PosSwitch5
	5	iAx_PosSwitch6
	6	iAx_PosSwitch7

Safety functions (SPF) - Basic parameters

	7	iAx_PosSwitch8
		Note that the designations iAx_PosSwitch1 .. iAx_PosSwitch8 in the NC configuration data correspond to the actual values 0 .. 7.
Pre-assignment	0	iAx_PosSwitch1
Unit	-	
Description of parameter values	The PLC input signal for the respective axis interface for the monitoring signal.	
		Ensure that the single parameters (.../Point[1..64]/IfSigWayPoint) that are set to the same value are assigned to different axes in the parameter "DrIndWayPoint" in folder SFU/WayPoint/Point[1..64]/.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	2010 00120	
Data type	Byte0_7_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

18.3 Area control (AreaCtrl)

18.3.1 General information

The MTX can define up to 10 two-dimensional areas (Area[1-10]) as "dead ranges or workspaces". Inputs of positions running into a dead area or out of a workspace are not accepted provided the respective control area is active.

For each area control Area[1..10], you can set the following parameters:

18.3.2 Area monitoring Area[1..10]

Type of monitoring area

Monitoring area type		AreaTypeAreaCtrl
Specifies the type of control area for each of maximum 10 possible area controls (Area[1..10]).		
Path	SFU/AreaCtrl/Area[1..10]/AreaTypeAreaCtrl	
Value range	0	Monitoring cannot be enabled
	1	dead area
	2	working area
Pre-assignment	0	Monitoring cannot be enabled
Unit	-	
Description of parameter values	0	The control area cannot be enabled.
	1	The control area is a dead area.
	2	The control area is a workspace.
Access rights	MTB:	Write

	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8002 00031	
Data type	Byte0_2_t (byte)	
	4 byte integer for MACODA compatibility	

Programmability of the monitoring area

Programmability of the monitoring area		ProgModeAreaCtrl
	Specifies whether a program can modify the size and position of a control area Area[1..10].	
Path	SFU/AreaCtrl/Area[1..10]/ProgModeAreaCtrl	
Value range	0	program cannot modify range
	1	program can modify range
Pre-assignment	1	program can modify range
Unit	-	
Description of parameter values	0	The control area cannot be modified
	1	The control area can be modified by a program.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8002 00032	
Data type	Byte0_1_t (byte)	
	4 byte integer for MACODA compatibility	

18.3.3 Dimension Dim[1..2]

Axis number of the physical axis

Axis number of the physical axis		DrIndAreaCtrl
	Defines for each of maximum 10 possible area controls (Area[1..10]) the number of the first system axis (for Dim[1]) or the second system axis (for Dim[2]) that span the respective control area.	
Path	SFU/AreaCtrl/Area[1..10]/Dim[1..2]/DrIndAreaCtrl	
Value range	1 .. 250	
Pre-assignment	1	SFU/AreaCtrl/Area[1..10]/Dim[1]/DrIndAreaCtrl
	2	SFU/AreaCtrl/Area[1..10]/Dim[2]/DrIndAreaCtrl
Unit	-	
Description of parameter values	Number of the first system axis (for Dim[1]) or the second system axis (for Dim[2])	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Safety functions (SPF) - Basic parameters

Activation	System restart (CS)
Dependency	-
Macoda ID	8002 00001, 8002 00002
Data type	Dr1_t (byte) 1 byte integer, unsigned for MACODA compatibility

Center point of the monitoring area (in MCS)

Center point of monitoring area (in MCS)		CentPointAreaCtrl
Defines for each of maximum 10 possible area controls (Area[1..10]) the desired center point coordinates of the control area on the first participating axis (1st dimension Dim[1]) or on the second participating axis (2nd dimension Dim[2]).		
Path	SFU/AreaCtrl/Area[1..10]/Dim[1..2]/CentPointAreaCtrl	
Value range	-1000000.000 .. 1000000.000	
Pre-assignment	0.000	
Unit	mm	
Description of parameter values	Center point coordinates of the control area of the first or second participating axis	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8002 00011, 8002 00012	
Data type	Pos_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

Extension of the monitoring area (Side Length)

Extension of the monitoring area (Side Length)		ExpandAreaCtrl
Defines for each of maximum 10 possible area controls (Area[1..10]) the relative length in the positive or negative axis direction of the control area on the first participating axis (1st dimension Dim[1]) or on the second participating axis (2nd dimension Dim[2]).		
Path	SFU/AreaCtrl/Area[1..10]/Dim[1..2]/ExpandAreaCtrl	
Value range	-1000000.000 .. 1000000.000	
Pre-assignment	0.000	
Unit	mm	
Description of parameter values	Relative length in the positive or negative axis direction of the control area (Area[1..10]) on the first participating axis (1st dimension Dim[1]) or on the second participating axis (2nd dimension Dim[2])	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	

Macoda ID 8002 00021, 8002 00022
Data type Pos_t (double, 3 decimal places)
 8 byte float, 3 decimal places for MACODA compatibility

18.4 Distance control to programmed end point (AxDistCtrl)

18.4.1 General information

This control tests the distance to the programmed endpoint. If axes are to travel at high velocities and with interpolation across blocks, distances that are set too short may have the effect that, from an NC perspective, reaching the distance to the endpoint coincides with the beginning of the next block. In this case, a corresponding interface signal might not output.



Distance monitoring has no relevance to

- slave axes when axes coupling is enabled
- Handwheel mode
- "Approach reference point " mode

18.4.2 Enabling the distance control

Enable distance control		EnablAxDistCtrl
Specifies for each axis Dr[i] whether distance control to the programmed end-point shall be enabled.		
Path	SFU/AxDistCtrl/Dr[i]/EnablAxDistCtrl	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Distance control not enabled.
	1	Yes
		Distance control enabled.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1015 00010	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

18.4.3 Distance to the programmed endpoint

Distance to programmed endpoint	DistEndPoint
Specifies for each axis Dr[i] from what distance to the programmed endpoint the axis interface signal "Axis near endpoint" (iAx_DistCtrl) shall be output.	
Path	SFU/AxDistCtrl/Dr[i]/DistEndPoint

Safety functions (SPF) - Basic parameters

Value range	0.000 .. 1000000.000 0.000 .. 999.000 for MACODA compatibility
Pre-assignment	10.000
Unit	mm or ° (degrees)
Description of parameter values	Desired distance to programmed endpoint (on the condition that distance control is enabled). As soon as the axis moves within this distance, the control unit sets the output signal "Axis near endpoint" (iAx_DistCtrl) on the respective axis interface. The signal is reset with every new position data input, provided the programmed path is longer than the distance to the endpoint as set in the parameter.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	1015 00010
Data type	Dist_t (double, 3 decimal places) 4 byte float, 3 decimal places for MACODA compatibility

18.5 One-dimensional collision monitoring (CollMon)

18.5.1 General information

The function "One-dimensional collision monitoring" makes it possible to enable up to 10 monitoring groups Gr[1..10] (Group(1..10)). Every monitoring group checks 2 system axes for collision.

18.5.2 Enabling the one-dimensional collision monitoring

Enabling one-dimensional collision monitoring		EnablCollMon
Defines for each of the 10 possible monitoring groups Gr[1..10] whether it is enabled.		
Path	SFU/CollMon/Gr[1..10]/EnablCollMon	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No Monitoring group Gr[1..10] is not enabled.
	1	Yes Monitoring group Gr[1..10] is enabled.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8008 00001	

Data type Byte0_1_t (byte)
1 byte integer, unsigned for MACODA compatibility

18.5.3 Monitoring axis 1

Monitoring axis 1	DrIndMonAx1

Defines for each of the 10 possible monitoring groups Gr[1..10] the system axis number of the "first monitoring axis".



Monitoring axes 1 and 2 can belong to different channels.

Path SFU/CollMon/Gr[1..10]/DrIndMonAx1
Value range 0 .. 250
Pre-assignment 0
Unit -
Description of parameter values 0 "1st monitoring axis" is not defined.
>0 System axis number of the "1st monitoring axis"
Access rights **MTB:** Write
Setter: Read
Operator: Read
Activation System restart (CS)
Dependency -
Macoda ID 8008 00002
Data type Dr_t (byte)
1 byte integer, unsigned for MACODA compatibility

18.5.4 Monitoring Axis 2

Monitoring axis 2	DrIndMonAx2

Defines for each of the 10 possible monitoring groups Gr[1..10] the system axis number of the "second monitoring axis".



Monitoring axes 1 and 2 can belong to different channels.

Path SFU/CollMon/Gr[1..10]/DrIndMonAx2
Value range 0 .. 250
Pre-assignment 0
Unit -
Description of parameter values 0 "2nd monitoring axis" is not defined.
>0 System axis number of the "second monitoring axis"
Access rights **MTB:** Write
Setter: Read
Operator: Read
Activation System restart (CS)
Dependency -
Macoda ID 8008 00003

Safety functions (SPF) - Basic parameters

Data type Dr_t (byte)
1 byte integer, unsigned for MACODA compatibility

18.5.5 Traveling direction

Traveling direction		DirCollMon
Defines for each of the 10 possible monitoring groups Gr[1..10] the traveling direction between the "first" and "second" monitoring axes.		
Path	SFU/CollMon/Gr[1..10]/DirCollMon	
Value range	0	synchronous
	1	asynchronous
Pre-assignment	0	synchronous
Unit	-	
Description of parameter values	0	Synchronous traveling direction between the monitoring axes. When velocity is positive, both axes move in the same direction.
	1	Asynchronous traveling direction between the monitoring axes.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8008 00004	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

18.5.6 Monitor upper limit

Monitor upper limit		ActUpLim
Defines for each of the 10 possible monitoring groups Gr[1..10] whether the upper limit shall be monitored.		
Path	SFU/CollMon/Gr[1..10]/ActUpLim	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		The upper limit is not monitored
	1	Yes
		The upper limit is monitored
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	

Macoda ID 8008 00005
Data type Byte0_1_t (byte)
 1 byte integer, unsigned for MACODA compatibility

18.5.7 Monitor lower limit

Monitor lower limit		ActLowLim
Defines for each of the 10 possible monitoring groups Gr[1..10] whether the lower limit shall be monitored.		
Path	SFU/CollMon/Gr[1..10]/ActLowLim	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		The lower limit is not monitored
	1	Yes
		The lower limit is monitored
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8008 00006	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

18.5.8 Upper limit value

Upper limit value		PosUpLim
Defines the upper limit for each of the 10 possible monitoring groups Gr[1..10].		
To permit determination of the upper limit value, the two monitored axes must be traveled to the collision limit and the actual positions of either axis must be read out. Depending on the particular application, the upper limit value results from the sum total or from the difference of the axis positions.		
Path	SFU/CollMon/Gr[1..10]/PosUpLim	
Value range	-1000000.000 .. 1000000.000	
Pre-assignment	0.000	
Unit	mm or ° (degrees)	
Description of parameter values	Upper limit value	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	

Safety functions (SPF) - Basic parameters

Macoda ID 8008 00007
Data type Pos_t (double, 3 decimal places)
 1 byte integer, unsigned for MACODA compatibility

18.5.9 Lower limit value

Lower limit value		PosLowLim
Defines the lower limit for each of the 10 possible monitoring groups Gr[1..10]. To permit determination of the lower limit value, the two monitored axes must be traveled to the collision limit and the actual positions of either axis must be read out. Depending on the particular application, the lower limit value results from the sum total or from the difference of the axis positions.		
Path	SFU/CollMon/Gr[1..10]/PosLowLim	
Value range	-1000000.000 .. 1000000.000	
Pre-assignment	0.000	
Unit	mm or ° (degrees)	
Description of parameter values	Lower limit value	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8008 00008	
Data type	Pos_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

19 Safety Engineering (SAFE) - Optional parameters -

19.1 General information

Use the data group "Safety Engineering" (SAFE) to parameterize the drive function "Safety Engineering" for each drive Dr[i].

19.2 Drive-specific Safety parameters

19.2.1 Enabling intelligent Safety Engineering

Enable intelligent Safety Engineering	EnablSafeTech

Defines for each drive Dr[i] whether the drive function "Safety Engineering" can be enabled (if present).



This function can only be used with "IndraDrive"
(see "DrType" in the folder MAIN/Dr[i]/Scs/).

Path	SAFE/Dr[i]/EnablSafeTech	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		The "Safety Engineering" drive function cannot be enabled.
	1	Yes
Access rights		The "Safety Engineering" drive function can be enabled.
	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1001 00002	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

19.2.2 Hiding Safety Engineering in the NC

Hide safety engineering in NC	SupprSafeTechNc

Defines for each drive Dr[i] with enabled Safety Engineering (see parameter "EnablSafeTech") whether the Safety Engineering can be analyzed within the NC or whether only the safety signals are to be transferred.



This function can only be used with "IndraDrive"
(see "DrType" in the folder MAIN/Dr[i]/Scs/).

This parameter is **only** intended for testing the Safety Engineering. It can be tested whether the drive generates an error message when the safety limits are exceeded without analyzing the signals.

Safety Engineering (SAFE) - Optional parameters -

Path	SAFE/Dr[i]/SupprSafeTechNc	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
	The Safety Engineering is analyzed: The Safety Engineering specifications are considered when generating the command values.	
	1	Yes
	The Safety Engineering is not analyzed: Test mode; i.e. the signals of the Safety Engineering are only transferred.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1001 00003	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

19.2.3 Scaling factor for save velocities

Scaling factor for safe velocities		VelWeightFact
Defines for each drive Dr[i] the scaling factor for safe velocities used to scale the command value generation in the NC.		
Especially when using encoders with a rough resolution, a "measured" velocity fluctuation might not cause an drive-internal exceedance of the safe velocity.		
Path	SAFE/Dr[i]/VelWeightFact	
Value range	0 .. 100	
Pre-assignment	95	
Unit	%	
Description of parameter values	Scaling factor for safe velocities	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1001 00004	
Data type	Byte0_100_t (byte)	
	2 byte integer for MACODA compatibility	

20 Electronic axis coupling (COUP)Parameter-

20.1 General information

In the data group Electronic axis coupling (COUP), gantry groups and coupled axes can be parameterized.

20.2 Axis coupling (AxCouple)

20.2.1 Maximum following error

Maximum lag distance		MaxFollErr
Defines the maximum following error allowed for each Gantry slave axis Dr[i]. The following error results from the difference of the command value/actual value deviations between master and slave: Following error = $\text{Abs}((\text{actual}_{\text{Master}} - \text{Command}_{\text{Master}}) - (\text{Actual}_{\text{Slave}} - \text{Command}_{\text{Slave}}))$		
Path	COUP/AxCouple/Dr[i]/MaxFollErr	
Value range	0.000 .. 1000.000 0.000 .. 999.999 for MACODA compatibility	
Pre-assignment	0.000	
Unit	mm or ° (degrees)	
Description of parameter values	0.000	No following error monitoring. Gantry axes cannot be traveled with this setting.
	>0.000	Maximum permissible following error (monitoring enabled). If this following error is exceeded, the controller generates a coupling error.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00060	
Data type	Double0_1T_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

20.2.2 Axis group index

Axis group index		MastInd
<i>Specifies for each axis Dr[i]</i>		
- whether it belongs to a gantry group and - whether it is a master axis of a gantry group.		
Path	COUP/AxCouple/Dr[i]/Gantry/MastInd	
Value range	0 .. 250	
Pre-assignment	0	
Unit	-	

Electronic axis coupling (COUP)Parameter-

Description of parameter values	0	Axis Dr[i] does not belong to any gantry group.
	>0	Axis belongs to a gantry group. For each participating axis Dr[i] of the gantry group (including the master axis), the axis index i of the master axis Dr[i=master axis] must be entered into the parameter MastInd.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00061	
Data type	Dr_t (byte)	
	1 byte integer for MACODA compatibility	
<i>Example:</i>		
Specification of the axis groups index:		
COUP/AxCouple/Dr[2]/Gantry/MastInd = 2		
COUP/AxCouple/Dr[5]/Gantry/MastInd = 2		
COUP/AxCouple/Dr[6]/Gantry/MastInd = 2		
The gantry group consists of axes 2, 5 and 6. The master axis is axis 2.		

20.2.3 Maximum standstill torque total of the group

Max. standstill torque of group		MaxGrTorq
<i>Defines for every axis Dr[i=master axis]:</i>		
- whether the standstill torque monitoring of the group is to be enabled - its torque limit.		
Path	COUP/AxCouple/Dr[i]/Gantry/MaxGrTorq	
Value range	0.00 .. 10000.00	
Pre-assignment	0.00	
Unit	%	
Description of parameter values	0.00	Monitoring is disabled.
	>0.00	Monitoring is enabled: Maximum total torque of all axes in the group at a standstill. The total static torque results from the static torque values of all individual axes of the group. The total value is entered only for master axes.
 The involved drives have to be parameterized in a way that the value in S-0-0092 is interpreted as % of the rated torque!		
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1003 00062	

Data type Double0_1T_t (double, 2 decimal places)
8 byte float, 2 decimal places for MACODA compatibility

20.2.4 Compensate Gantry offset command value

Compensate for gantry command value offset	CmdOffslpoMode

The parameter specifies for each drive Dr[i=slave]

- whether the "command value offset" between the gantry master and the gantry slave may be compensated by setting the appropriate interface/signal;
- whether the "command value offset" shall equate to the upper limit given in parameter "Upper limit for gantry command value offset" ("MaxRedCmdOffs" in the folder COUP/AxCouple/Dr[i]/Gantry/) or to the upper limit given in parameter "Maximum following error"=compatibility ("MaxFollErr" in the folder COUP/AxCouple/Dr[i]/).

This parameterization is only to be done for the slave drives Dr[i=slave] in the group.

Path	COUP/AxCouple/Dr[i]/Gantry/CmdOffslpoMode	
Value range	0	No
	1	ax. value from "Upper limit for gantry command value offset"
	2	max. value from "Maximum lag distance"
Pre-assignment	2	max. value from "Maximum lag distance"
Unit	-	
Description of parameter values	0	No No compensation of command value offset permitted.
	1	Compensation of the command value offset permitted up to "Upper limit for gantry command value offset" in the parameter MaxRedCmdOffs.
	2	Compensation of the command value offset permitted up to "Maximum following error" in the parameter "MaxFollErr".
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00066	
Data type	Byte0_2_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

20.2.5 Upper limit for Gantry command value offset

Upper limit for Gantry command value offset	MaxRedCmdOffs

This parameter defines the maximum distance up to which the command value offset between the gantry master and the gantry slave may be compensated by setting the appropriate interface/signal.



The parameter can only be used after the setting in
COUP/AxCouple/Dr[i]/Gantry/CmdOffslpoMode = 1 (max. value from "Upper limit for gantry command value offset").

Electronic axis coupling (COUP)Parameter-

	This parameterization is only to be done for the slave drives Dr[i=slave] in the group.	
Path	COUP/AxCouple/Dr[i]/Gantry/MaxRedCmdOffs	
Value range	0.00 .. 10000.00 0.00 .. 9999.99 for MACODA compatibility	
Pre-assignment	0.00	
Unit	mm or ° (degrees)	
Description of parameter values	Maximum distance of the command value offset	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	CmdOffsIpoMode	
Macoda ID	1003 00067	
Data type	Double0_1T_t (double, 2 decimal places) 8 byte float, 2 decimal places for MACODA compatibility	

20.2.6 Synchronous run monitoring on

Enable synchronous run monitoring		EnablSynchrMon
Specifies when gantry axes Dr[i] whether all axes have reached their reference point within a monitoring window.		
 The parameter should be disabled again after startup.		
Path	COUP/AxCouple/Dr[i]/Gantry/EnablSynchrMon	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No No monitoring enabled.
	1	Yes Monitoring enabled, monitoring window according to parameter "SynchrMonWin"..
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00063	
Data type	Byte0_1_t (byte) 1 byte integer, unsigned for MACODA compatibility	

20.2.7 Synchronous run monitoring window

Synchronous run monitoring window		SynchrMonWin
Specifies when referencing gantry axes Dr[i] the monitoring window in which all axes in the gantry group must have reached their reference point. This parameterization is only to be done for the master axis Dr[i=master axis] in the group.		
Path	COUP/AxCouple/Dr[i]/Gantry/SynchrMonWin	
Value range	0.000 .. 100000.000	
Pre-assignment	0.000	
Unit	ms	
Description of parameter values	0.000	All axes must have completed referencing in the same interpolation cycle.
	>0.000	Maximum permissible time window in milliseconds (is always rounded up to full interpolation cycles). Since it is not ensured that the data for all axes are received in the same Sercos/interpolation cycle, it is recommended to select a time window of at least +/-1 interpolation cycles.
 The parameter should be disabled again after startup because the axes can no longer be aligned when referencing if a skew position exists.		
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00064	
Data type	Double0_100T_t (double, 3 decimal places)	
	4 byte integer, unsigned for MACODA compatibility	

20.2.8 Reference switch monitoring

Reference switch monitoring		EnablRefSwiMon
Enables monitoring of the reference switch while referencing gantry axes Dr[i]. If an axis is already positioned on the reference switch when referencing is started, referencing is stopped to avoid a reversal in direction of movement. This parameterization is only to be done for the master axis Dr[i=master axis] in the group.		
Path	COUP/AxCouple/Dr[i]/Gantry/EnablRefSwiMon	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Monitoring not enabled.

Electronic axis coupling (COUP)Parameter-

	1	Yes
		Monitoring enabled.
		Monitoring is possible only if a reference switch is provided and evaluated and if the parameter S-0-0400 is transferred in the cyclic drive telegram.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00065	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

20.2.9 Additional axes: System axis number of the master

Additional axes: System axis number of the master		MastIndSuplAx
Defines for each axis Dr[i] whether it is allowed to behave as a supplementary axis master or supplementary axis slave or which axis Dr[i] could be its supplementary axis master.		
Path	COUP/AxCouple/Dr[i]/SuplAx/MastIndSuplAx	
Value range	0 .. 250	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	No index: Axis Dr[i] may be neither a supplementary axis master nor a supplementary axis slave.
	>0 (case 1)	Value = own axis index i of Dr[i]. Axis Dr[i] is allowed to be a supplementary axis master. It can be made a master by programming/enabling a slave.
	>0 (case 2)	Value <> own axis index i of Dr[i]. Axis Dr[i] is allowed to be a supplementary axis slave of the master axis defined by the parameter value.
	A system axis with a parameter value other than "0" can be made a slave by programming the appropriate NC syntax. The potential master axis defined in the parameter value then automatically becomes the master axis. Making the required entry for the master axis ensures that no invalid groups are configured	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00080	
Data type	Dr_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Example:

Supplementary axis master / supplementary axis slave:

COUP/AxCouple/Dr[1]/Gantry/MastIndSupplAx 1	= The 1st axis is a possible supplementary axis master.
COUP/AxCouple/Dr[2]/Gantry/MastIndSupplAx 0	= The 2nd axis may be neither a supplementary axis master nor a supplementary axis slave.
COUP/AxCouple/Dr[4]/Gantry/MastIndSupplAx 1	= The 4th axis is a possible supplementary axis slave of the 1st axis.

21 Digitalizing (DIG) - Optional parameters

21.1 General information

The memory size for the data to be digitized can be adjusted in the data group "Digitalisieren (DIG)".

For detailed information on the "Digitize" function, refer to the manual "MTX 15VRS Functional Description - Extension" (DOK-MTX***-NC*F*EX*V15-RERS-EN-P).

21.2 Digitized data

21.2.1 Number of digitizing buffers

Number of digitizing buffers		NofDigBuff
Specifies the number of digitizing buffers to buffer "real-time data". The buffer size is 512 bytes.		
Path	DIG/NofDigBuff	
Value range	20 .. 2000	
Pre-assignment	50	
Unit	-	
Description of parameter values	Number of digitizing buffers	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8006 00001	
Data type	DigBuff_t (unsignedInt)	
	2 byte integer, unsigned for MACODA compatibility	

21.2.2 Number of reserve buffers

Number of reserve buffers		NofReservBuff
Specifies the minimum number of free digitizing buffers. The buffer size is 512 bytes.		
If the number of free buffers falls below the number of reserve buffers, the control responds according to the parameterization of the digitizing function by the part program (also refer to the command "RecordSet").		
Path	DIG/NofReservBuff	
Value range	5 .. 100	
Pre-assignment	10	
Unit	-	
Description of parameter values	Number of reserve buffers	
Access rights	MTB: Write Setter: Read Operator: Read	

Digitalizing (DIG) - Optional parameters

Activation	System restart (CS)
Dependency	-
Macoda ID	8006 00002
Data type	ResDigBuff_t (unsignedInt) 2 byte integer, unsigned for MACODA compatibility

22 Measuring (MEAS) - Optional parameters

22.1 General information

The data group Measuring (MEAS) is where axis-dependent measurement touch probes can be set. These settings are allowed for all drives that have the MAIN functionality assigned in the data group.

22.2 Touch probes

22.2.1 Touch probe function available

Touch probe function available	EnablProbeFunc

The system axis Dr[i] is defined as measurement axis when setting the machine parameters.

- The NC functions require measurement axes with touch probe 1:
 - G75.1 or G75 (travel to first touch probe)
 - FME or FlyMeas (Measurement On-the-fly)
- The NC function requires measurement axes with touch probe 2:
 - G75.2 (travel to second touch probe)

There are dependencies on the following machine parameters (MP):

- As soon as a measurement axis is defined for touch probe 1, it is to be specified at which edge it has to trigger. This is executed via MP **Probe1Edge**.
- As soon as a measurement axis is defined for touch probe 2, it is also to be specified at which edge it has to trigger. This is executed via MP **Probe2Edge**.
- MP **Probe1Edge** and MP **Probe2Edge** are irrelevant for all the system axes Dr[i] that were not defined as measurement axes.

The NC checks during Sercos startup only whether every measurement axis defined as drive supports the touch probe function. If this is not the case, the respective drive cancels the Sercos startup and indicates the missing Sercos IDN and therefore, the missing touch probe function via a respective error message.

Additional settings and checks on the drive might be required. These are to be carried out by the user before the touch probe function of the NC starts operating. For detailed information, refer to the documentation of the drive manufacturer.

The user is also responsible for a proper signal generation for the touch probes 1 and 2 and that it is connected correctly to the hardware input of the drive.

Path	MEAS/Dr[i]/EnablProbeFunc (< MTX09: MEAS/Dr[i]/EnablMeas)	
Value range	0	No
	1	Only touch probe 1
	2	Both touch probes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No

Measuring (MEAS) - Optional parameters

		The system axis Dr[i] is no measurement axis
	1	Only touch probe 1
		The system axis Dr[i] is no measurement axis for touch probe 1
	2	The system axis Dr[i] is no measurement axis for touch probe 1 and 2
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00012	
Data type	Byte0_2_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

22.2.2 Edge touch probe 1

Edge of touch probe 1		Probe1Edge
	If the system axis Dr[i] was defined as measurement axis via the machine parameter "EnablProbeFunc", it is defined here at which edge the 1st touch probe triggers.	
Path	MEAS/Dr[i]/Probe1Edge (< MTX09: MEAS/Dr[i]/MeasEdge)	
Value range	0	negative edge
	1	positive edge
Pre-assignment	1	positive edge
Unit	-	
Description of parameter values	0	negative edge
		The 1st touch probe triggers in case of a negative edge.
	1	positive edge
		The 1st touch probe triggers in case of a positive edge.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	1003 00011	
Data type	Byte0_1_t (byte)	
	4 byte integer, unsigned for MACODA compatibility	

22.2.3 Edge Touch Probe 2

Edge of touch probe 2	Probe2Edge
If the system axis Dr[i] was defined as measurement axis via the machine parameter "EnablProbeFunc", it is defined here at which edge the 2nd touch probe triggers.	

Path	MEAS/Dr[i]/Probe2Edge	
Value range	0	negative edge
	1	positive edge
Pre-assignment	1	positive edge
Unit	-	
Description of parameter values	0	negative edge
		The 2nd touch probe triggers in case of a negative edge.
	1	positive edge
Access rights		The 2nd touch probe triggers in case of a positive edge.
	MTB:	Write
	Setter:	Read
Activation	Operator:	Read
	System restart (CS)	
Dependency	-	
Macoda ID	1003 00013	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

23 Punching and nibbling (PUN) - Optional parameters

23.1 General information

The Punching and Nibbling (PUN) data group is where functions are set for stroke monitoring and stroke release times.

23.2 Stroke monitoring

23.2.1 Enabling punching and nibbling

	Enable punching and nibbling	EnablPunch
	Enables the functions "Punching" and "Nibbling". <i>In the part program, the functions "Punching" and "Nibbling" can be controlled individually:</i>	
	- Punch (PUN) programs punching - Nibble (NIB) programs nibbling (see also "TimeRef" in folder PUN/StrokRelTime/Dr[i]/)	
Path	PUN/EnablPunch	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No Functions "Punching" and "Nibbling" are disabled.
	1	Yes Functions "Punching" and "Nibbling" are enabled.
	 An additional digital input and an additional digital output must be reserved if the "Punching/Nibbling" function is enabled (see folder PLC/DigIO/...).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8001 00010	
Data type	Byte0_1_t (byte)	
	1 byte integer for MACODA compatibility	

23.2.2 Stroke monitoring time

Stroke check time	StrokeMonTime
Specifies whether or not, and if so, after what period of time (T_{stroke}) an error message is displayed if a stroke cannot be released or completed.	
An MTX-P digital input and output (see folder PLC/DigIO/...) is used to communicate with the punching control.	

Punching and nibbling (PUN) - Optional parameters

Errors that may occur when stroke monitoring time is enabled:

After the T_{stroke} interval has elapsed, the following warning will be displayed:

- "Stroke cannot be released":
This means that the control cannot request the stroke because it is running (low level is applied to the signal at the digital input of the MTX-P). In this case, T_{stroke} starts to run at the stroke release time.
- "Stroke has not been released":
This means that the control requested the stroke via the digital output, however, the punching control did not release the stroke (digital input does not change to LOW level). In this case, T_{stroke} starts to run when the digital output is set.
- "Stroke has not been completed":
This means that the punching control released the stroke but the stroke was not completed within the T_{stroke} interval (LOW level is still applied to the digital input).

Path	PUN/StrokeMonTime	
Value range	0 .. 4294967295	
Pre-assignment	1,000	
Unit	ms	
Description of parameter values	0	Stroke monitoring off.
	>0	Stroke monitoring on. The specified time is rounded off internally to the raster of the sercos cycle time (see also parameter "ScsCycTime" in folder SCS/Scslf[1]/).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	8001 00030	
Data type	unsignedInt	
	4 byte integer, unsigned for MACODA compatibility	

23.3 Axis-specific stroke release times

23.3.1 Reference of stroke release times

Referencing stroke release times		TimeRef
Specifies for each axis Dr[i] the zero point in time for stroke release. The time when the stroke is finally released depends on the setting in "TimeOffs" in the folder PUN/StrokeRelTime/Dr[i]/.		
Path	PUN/StrokeRelTime/Dr[i]/TimeRef	
Value range	0	interpolation end point
	1	In-Pos window of the axis
Pre-assignment	1	In-Pos window of the axis
Unit	-	

Description of parameter values	0	The zero point in time is defined as the end point of interpolation of the stroke release block. The stroke release can be offset by programming "PtBlkEnd" (PTE).
	1	The zero point in time is reached when the in-position window of the axis is reached. If several axes are involved in the path movement: stroke release when the last participating axis has reached its in-position window. The stroke release can be offset by programming "PtInpos" (PTI).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	TimeOffs in the folder PUN/StrokeRelTime/Dr[i]/	
Macoda ID	8001 00021	
Data type	Byte0_1_t (byte)	
	4 byte integer, unsigned for MACODA compatibility	

23.3.2 Stroke release times

Stroke release time		TimeOffs
Offsets the stroke release for each axis Dr[i], relating to the zero point in time defined in the parameter "TimeRef" (PUN/StrokeRelTime/Dr[i]).		
Path	PUN/StrokeRelTime/Dr[i]/TimeOffs	
Value range	Integer range	
Pre-assignment	0	
Unit	ms	
Description of parameter values	<0	Early stroke release: Only possible if TimeRef in (the folder PUN/StrokeRelTime/Dr[i]/) is assigned the value "0". Maximum possible stroke release: at the beginning of the stroke release block.
	0	No time offset of stroke release.
	>0	Subsequent stroke release.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	8001 00020	
Data type	int	
	4 byte integer for MACODA compatibility	

24 Online corrections (ONL) - Optional parameters -

24.1 General information

The online correction (ONL) can be set channel-specifically for the

- [Machine coordinate system \(MCS\), page 361](#) and
- [Workpiece coordinate system \(WCS\), page 369](#)

24.2 Online correction in MCS (distance control)

24.2.1 General information

Online correction in MCS requires the following settings:

- [Enabling online correction in MCS, page 361](#)
- [Basic data, page 362](#)
- [Sensor selection, page 363](#)
- [Increment evaluation, page 365](#)
- [Limitations, page 366](#)
- [Monitoring, page 368](#)

24.2.2 Enabling the online correction in MCS

Enabling the MCS online correction

Enabling MCS online correction		EnablMcsCorr
	Specifies channel-specifically whether online correction (distance control) is available in channel Ch[k].	
Path	ONL/McsOnlCorr/Ch[k]/EnablMcsCorr	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Distance control not available.
	1	Yes
		Distance control available.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 00700	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

24.2.3 Basic data

Channel axis

Channel axis		ChAxMcsCorr
Specifies channel-specifically Ch[k] which channel axis is corrected by the distance control (also see CHAN/Ch[k]/Coord/Acs/ChAx[1..8]/ChAxName (axis name).		
 This parameter can be overridden in the part program by "DistCtrl(1) DcAxis(..)".		
Path	ONL/McsOnlCorr/Ch[k]/Base/ChAxMcsCorr	
Value range	0 .. 8	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	No channel axis selected
	>0	Number of the channel axis on which the distance control has an effect on.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00702	
Data type	ChAx_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Correction direction

Direction of correction		DirMcsCorr
Specifies for each channel Ch[k] in which direction the channel axis correction shall act (ONL/McsOnlCorr/Ch[k]/Base/DirMcsCorr).		
Path	ONL/McsOnlCorr/Ch[k]/Base/DirMcsCorr	
Value range	0	Positive
	1	Negative
Pre-assignment	0	Positive
Unit	-	
Description of parameter values	0	positive correction direction
	1	negative correction direction
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	ChAxMcsCorr in folder ONL/McsOnlCorr/Ch[k]/Base/	
Macoda ID	7050 00702	

Data type Byte0_1_t (byte)
1 byte integer, unsigned for MACODA compatibility

Interface signal

Interface signal		ChIfSigMcsCorr
Defines for each channel Ch[k] with which channel interface signal an enabled online correction (distance control) shall be reported.		
Path	ONL/McsOnlCorr/Ch[k]/Base/ChIfSigMcsCorr	
Value range	-1	no feedback
	0	Customer input 0 (22.0)
	1	Customer input 1 (22.1)
	2	Customer input 2 (22.2)
	3	Customer input 3 (22.3)
	4	Customer input 4 (22.4)
	5	Customer input 5 (22.5)
	6	Customer input 6 (22.6)
	7	Customer input 7 (22.7)
Pre-assignment	-1	no feedback
Unit	-	
Description of parameter values	-1	no feedback active
	0..7	Customer input 0..7 (22.0 .. 22.7) selected channel interface signal: 22.0: iCh_Custom1 22.1: iCh_Custom2 ... 22.7: iCh_Custom8
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00708	
Data type	Byte-1_7_t (byte) 1 byte integer, unsigned for MACODA compatibility	

24.2.4 Sensor selection (Device)

Sensor selection (device)	Device
<i>The sensor selection (device) specifies which device group delivers the on-line correction data in the MCS:</i> - from the drive (Dr) - via permanent CPL variables (CplVar) For this, select the ONL/McsOnlCorr/Ch[k]/Device folder and, with the right mouse button, select the appropriate entry with the function "Convert to..." from the context menu . Dr or CplVar.	

Depending on device type, different subfolders will be presented in the folder ONL/McsOnlCorr/Ch[k]/Device/. By default, the device group "Drive" (Dr) is active.

Macoda ID 7050 00710 (1 = drive, 2 = CPL variable)

24.2.5 Device = Dr (Drive)

Drive number

	Drive number	DrIndMcsCorr
	Specifies for each channel Ch[k] the number of the drive if the device group "Dr" is selected.	
Path	ONL/McsOnlCorr/Ch[k]/Device/Dr/DrIndMcsCorr	
Value range	0 .. 250	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	No drive selected
	>0	Drive Number
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 00712	
Data type	Dr_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Sercos Parameters

	Sercos parameters	ScsParMcsCorr
	Specifies with which Sercos parameter (S-0-xxxx or P-0-xxxx) of the drive telegram the sensor value for the distance control is provided if the device group "Dr" is selected (drive according to parameter "ONL/McsOnlCorr/Ch[k]/Device/Dr/DrIndMcsCorr").	
Path	ONL/McsOnlCorr/Ch[k]/Device/Dr/ScsParMcsCorr	
Value range	S-0-xxxx or P-0-xxxx	
Pre-assignment	<EmptyString>	
	 If the sensor value for the distance control is provided by the drive, the parameter "ScsParMcsCorr" has to be adjusted before (e.g. S-0-0053, Actual position value 2 incremental).	
Unit	-	
Description of parameter values	Sercos parameters from the drive telegram	
Access rights	MTB: Write	
	Setter: Read	
	Operator: Read	
Activation	System restart (CS)	
Dependency	Implicitly overwrites the parameter CplVarMcsCorr	

Macoda ID 7050 00714
Data type Str16_t (isoLatin1String)
 2 x 8-byte string for MACODA compatibility

24.2.6 Device = CplVar (permanent CPL variable)

Variable name

	Variable name	CplVarMcsCorr
	Specifies the permanent CPL variable (@xxxxxx) used to provide the sensor value for the distance control if the device group "CplVar" has been selected.	
Path	ONL/McsOnlCorr/Ch[k]/Device/CplVar/CplVarMcsCorr	
Value range	@xxxxxx	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Name of the permanent CPL variable	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	Implicitly overwrites the parameter "ScsParMcsCorr"	
Macoda ID	7050 00714	
Data type	Str16_t (isoLatin1String) 2 x 8-byte string for MACODA compatibility	

24.2.7 Increment calculation

Multiplier

	Multiplier	FactMcsCorr
	Defines, in conjunction with parameter "DivMcsCorr" (in folder ONL/McsOnlCorr/Ch[k]/IncrScale/), a conversion factor for the sensor value: Sensor value * FactMcsCorr / DivMcsCorr	
Path	ONL/McsOnlCorr/Ch[k]/IncrScale/FactMcsCorr	
Value range	-10000000.000 .. 10000000.000 -9999999.999 .. 9999999.999 for MACODA compatibility	
Pre-assignment	1.000	
Unit	-	
Description of parameter values	Multiplier of the correction factor.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 00720	
Data type	Double-10M_10M_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

Divisor

	Divisor	DivMcsCorr
	Defines, in conjunction with parameter "FactMcsCorr" (in folder ONL/McsOnlCorr/Ch[k]/IncrScale/), a conversion factor for the sensor value: Sensor value * FactMcsCorr / DivMcsCorr	
Path	ONL/McsOnlCorr/Ch[k]/IncrScale/DivMcsCorr	
Value range	1.000 .. 10000000.000 1.000 .. 9999999.999 for MACODA compatibility	
Pre-assignment	10000.000	
Unit	-	
Description of parameter values	Divisor of the correction factor.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 00721	
Data type	Double0_10M_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

24.2.8 Limits

Maximum velocity

	Maximum velocity	MaxVelMcsCorr
	Defines a maximum rate of change of the online correction value.	
	 This parameter can be overridden in the part program by "DistCtrl(DcLimit(...))".	
Path	ONL/McsOnlCorr/Ch[k]/Lim/MaxVelMcsCorr	
Value range	0.000 .. 1000000000.000 0.001 .. 999999.999 for MACODA compatibility	
Pre-assignment	999999.999	
Unit	mm/min	
Description of parameter values	Maximum rate of change	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00740	
Data type	Vel_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

Maximum acceleration

Maximum acceleration	MaxAccMcsCorr

Defines a maximum steepness (acceleration) of the online correction value.



This parameter can be overridden in the part program by "DistCtrl(DcLimit(...))".

Path	ONL/McsOnlCorr/Ch[k]/Lim/MaxAccMcsCorr
Value range	0.000 .. 1000.000 0.001 .. 99.999 for MACODA compatibility
Pre-assignment	99.999
Unit	m/s ²
Description of parameter values	Maximum acceleration
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7050 00741
Data type	Acc_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

Smoothing/filter

Smoothing/Filter	FilterMcsCorr

Defines a filter for smoothing the sensor values.



This parameter can be overridden in the part program by "DistCtrl(1) DCFilter(<Time>)".

Path	ONL/McsOnlCorr/Ch[k]/Lim/FilterMcsCorr
Value range	0.0 .. 100.0 0.0 .. 40.0 for MACODA compatibility
Pre-assignment	0.0
Unit	ms
Description of parameter values	0.0 No smoothing enabled >0.0 Smoothing time
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7050 00730
Data type	Float0_100_t (float, 1 decimal place) 4 byte float, 1 decimal places for MACODA compatibility

24.2.9 Monitoring

Collision detection

Collision monitoring		CollDet
Defines a tolerance band for collision detection for online correction.		
 This parameter can be overridden in the part program by "DistCtrl(DcMon(...))".		
If the tolerance band is exceeded, a warning will be generated and the part program will be stopped using Feed hold.		
Path	ONL/McsOnlCorr/Ch[k]/Det/CollDet	
Value range	0.000 .. 1000.000 0.000 .. 999.999 for MACODA compatibility	
Pre-assignment	0.000	
Unit	mm	
Description of parameter values	0.000 No collision detection possible >0.000 Size of the tolerance band for collision detection	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00750	
Data type	Double0_1T_t (double, 3 decimal places) 4 byte float, 3 decimal places for MACODA compatibility	

Hole detection

Hole detection		HoleDet
 This parameter can be overridden in the part program by "DistCtrl(DcMon(...))".		
Defines a tolerance band for hole detection for online correction.		
If the tolerance band is exceeded, it is assumed that there is a hole; distance control is interrupted for the time needed to travel over it.		
Path	ONL/McsOnlCorr/Ch[k]/Det/HoleDet	
Value range	0.000 .. 1000.000 0.000 .. 999.999 for MACODA compatibility	
Pre-assignment	0.000	
Unit	mm	
Description of parameter values	0.000 No hole detection possible >0.000 Size of the tolerance band for hole detection	
Access rights	MTB: Write Setter: Read	

	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00752	
Data type	Double0_1T_t (double, 3 decimal places) 4 byte float, 3 decimal places for MACODA compatibility	

24.3 Online correction in WCS

24.3.1 General information

Online correction in WCS requires the following settings:

- [Enabling online correction in WCS, page 369](#)
- [Sensor selection, page 369](#)
- [Increment evaluation, page 372](#)
- [Limitations, page 373](#)
- [Axis interface for increment specification, page 374](#)

24.3.2 Enabling the online correction in WCS

Enabling the WCS online correction

Enable WCS online correction		EnablWcsCorr
Specifies for each channel Ch[k] whether online correction shall be applied to workpiece coordinates.		
Path	ONL/WcsOnlCorr/Ch[k]/EnablWcsCorr	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No No online correction available.
	1	Yes Online correction available.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 00900	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

24.3.3 Sensor selection (Device)

Sensor selection (device)	Device

The sensor selection (device) specifies which device group delivers the on-line correction data in the WCS:

- from the drive (Dr)
- via permanent CPL variables (CplVar)
- from the connected handwheel (Handw)

For this, select the **ONL/WcsOnlCorr/Ch[k]/Device** folder and, with the right mouse button, select the appropriate entry with the function "Convert to..." from the context menu: Dr, CplVar or Handw.

Depending on device type, different subfolders are provided in the folder ONL/WcsOnlCorr/Ch[k]/Device/. By default, the device group "Handwheel" (Hands) is enabled.

Macoda ID 7050 00910 (1 = drive, 2 = CPL variable, 3 = handwheel)



In special cases, the online correction data can be supplied via a drive (e.g. P-0-0553) or via a permanent CPL variable.

In this case, the parameters "DrIndWcsCorr" and "ScsParWcsCorr" must be accounted for.

24.3.4 Device = Dr (Drive)

Drive number

Drive number	DrIndWcsCorr

Specifies for each channel Ch[k] the number of the drive the data of the on-line correction come from if the device group "Dr" is selected.

Path	ONL/WcsOnlCorr/Ch[k]/Device/Dr/DrIndWcsCorr	
Value range	0 .. 250	
Pre-assignment	1	
Unit	-	
Description of parameter values	0	No drive selected
	>0	Drive Number
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 00912	
Data type	Byte0_2_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

Sercos Parameters

Sercos parameters	ScsParWcsCorr

Specifies with which Sercos parameter (S-0-xxxx or P-0-xxxx) of the drive telegram (drive according to parameter "ONL/WcsOnlCorr/Ch[k]/Device/Dr/DrIndWcsCorr") the correction value is returned if the device group "Dr" is selected.

Path ONL/WcsOnlCorr/Ch[k]/Device/Dr/ScsParWcsCorr

Value range	S-0-xxxx or P-0-xxxx
Pre-assignment	-
Unit	-
Description of parameter values	Sercos parameters from the drive telegram
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	Implicitly overwrites the parameter "CplVarWcsCorr"
Macoda ID	7050 00914
Data type	Str16_t (isoLatin1String) 2 x 8-byte string for MACODA compatibility

24.3.5 Device = CplVar (permanent CPL variable)

Variable name

Variable name	CplVarWcsCorr
	Specifies the permanent CPL variable (@xxxxxx) used to provide the correction value if the device group "CPLVar" has been selected.
Path	ONL/WcsOnlCorr/Ch[k]/Device/CplVar/CplVarWcsCorr
Value range	@xxxxxx
Pre-assignment	-
Unit	-
Description of parameter values	Name of the permanent CPL variable
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	Implicitly overwrites the parameter "ScsParWcsCorr"
Macoda ID	7050 00914
Data type	Str16_t (isoLatin1String) 2 x 8-byte string for MACODA compatibility

24.3.6 Device = Handw (Handwheel

Handwheel number

Handwheel number	HandwWcsCorr
	Specifies for each channel Ch[k] the number of the handwheel the data of the online correction come from if the device group "Handw" is selected.
Path	ONL/WcsOnlCorr/Ch[k]/Device/Handw/HandwWcsCorr
Value range	1 Handwheel 1 2 Handwheel 2
Pre-assignment	1 Handwheel 1
Unit	-

Online corrections (ONL) - Optional parameters -

Description of parameter values	1	Handwheel 1 Online correction data come from handwheel 1
	2	Handwheel 2 Online correction data come from handwheel 2
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 00912	
Data type	Byte0_2_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

24.3.7 Increment calculation

Multiplier

Multiplier		FactWcsCorr
Defines, in conjunction with parameter DivWcsCorr (in folder ONL/WcsOnlCorr/Ch[k]/IncrScale/), a conversion factor for the online correction value:		
Sensor value * FactWcsCorr / DivWcsCorr		
Path	ONL/WcsOnlCorr/Ch[k]/IncrScale/FactWcsCorr	
Value range	-10000000.000 .. 10000000.000	
	-9999999.999 .. 9999999.999 for MACODA compatibility	
Pre-assignment	1.000	
Unit	-	
Description of parameter values	Multiplier of the correction factor	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 00920	
Data type	Double-10M_10M_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

Divisor

Divisor		DivWcsCorr
Defines, in conjunction with parameter "FactWcsCorr" (in folder ONL/WcsOnlCorr/Ch[k]/IncrScale/), a conversion factor for the online correction value:		
Sensor value * FactWcsCorr / DivWcsCorr		
Path	ONL/WcsOnlCorr/Ch[k]/IncrScale/DivWcsCorr	
Value range	1.000 .. 10000000.000	
	1.000 .. 9999999.999 for MACODA compatibility	

Pre-assignment	10000.000
Unit	-
Description of parameter values	Divisor of the correction factor
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System restart (CS)
Dependency	-
Macoda ID	7050 00921
Data type	Double-10M_10M_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

24.3.8 Limits

Maximum velocity

Maximum velocity	MaxVelWcsCorr

Limits the speed for online corrections.



This parameter can be overridden in the part program by "DistCtrl(DcLimit(...))".

Path	ONL/WcsOnlCorr/Ch[k]/Lim/MaxVelWcsCorr
Value range	0.0 .. 1000000000.0 0.1 .. 999999.9 for MACODA compatibility
Pre-assignment	99999.9
Unit	mm/min
Description of parameter values	Maximum Velocity
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7050 00940
Data type	Vel_t (double, 1 decimal place) 8-byte float, 1 decimal place for MACODA compatibility

Maximum acceleration

Maximum acceleration	MaxAccWcsCorr

Defines a maximum steepness (acceleration) of the online correction value.



This parameter can be overridden in the part program by "DistCtrl(DcLimit(...))".

Path	ONL/WcsOnlCorr/Ch[k]/Lim/MaxAccWcsCorr
Value range	0.000 .. 1000.000 0.001 .. 99.999 for MACODA compatibility

Pre-assignment	99.999
Unit	m/s ²
Description of parameter values	Maximum acceleration
Access rights	MTB: Write
	Setter: Read
	Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7050 00941
Data type	Acc_t (double, 3 decimal places)
	8 byte float, 3 decimal places for MACODA compatibility

24.3.9 Axis interface for increment input

System axis number

	System axis number	DrIndIncrSource
	Specifies a system axis with the axis interface "qAx_ManFeed_00" to "qAx_ManFeed_03" (Manual feed bit 0-3) used to derive the size of an increment.	
	If the tolerance band is exceeded, a warning will be generated and the part program will be stopped using Feed hold.	
Path	ONL/McsOnlCorr/Ch[k]/IncrSource/DrIndIncrSource	
Value range	0 .. 250	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Increment size is not derived from the axis interface
	>0	Number of the system axis
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System restart (CS)	
Dependency	-	
Macoda ID	7050 00926	
Data type	Dr_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

25 Traveling to positive stop (MPS) - Optional parameters

25.1 General information

The data group "Move to positive stop (MPS)" includes settings for enabling, axis speeds, axis accelerations and torque limits.

25.2 Drive-dependent move to positive stop

25.2.1 Traverse to positive stop can be enabled

Move to positive stop enable		EnablMovePoStop
<i>Specifies for each machining axis Dr[i] whether</i>		
- measurements at the positive stop "FsProbe" (FSP) can be enabled, and - Move to positive stop "FsMove" (FSM) can be enabled		
Path	MPS/Dr[i]/EnablMovePoStop	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		FSP and FSM cannot be enabled.
	1	Yes
		<i>The axes identified with "1" of the corresponding channel can be</i>
		- selected for measuring under FSP. - moved to positive stop under FSM.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1003 00030	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

25.2.2 Maximum axis velocity

Maximum axis velocity	MaxVelPoStop
<i>Specifies for each machining axis Dr[i]</i>	
- what maximum axis velocity can be used to move to positive stop ("FsMove" (FSM)). - what maximum axis velocity can be used to carry out "Measurement at positive stop" ("FsProbe" (FSP)). The path velocity is calculated from the axis velocity of the axes involved.	

Traveling to positive stop (MPS) - Optional parameters

		FSP and FSM have to be enabled with "EnablMovePoStop".
Path	MPS/Dr[i]/MaxVelPoStop	
Value range	0.000 .. 10000000000.000 0.000 .. 360000.000 for MACODA compatibility	
Pre-assignment	1.000	
Unit	mm/min or °/min	
Description of parameter values	Maximum axis velocity (FSM and FSP)	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	EnablMovePoStop	
Macoda ID	1005 00030	
Data type	Vel_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

25.2.3 Maximum axis acceleration

Maximum axis acceleration		MaxAccPoStop
	Specifies for each machining axis Dr[i] what maximum axis acceleration can be used to move to positive stop ("FsMove" (FSM)).	
Path	MPS/Dr[i]/MaxAccPoStop	
Value range	0.000 .. 1000.000 0.001 .. 999.999 for MACODA compatibility	
Pre-assignment	0.1	
Unit	m/s ² or 1000 °/s ²	
Description of parameter values	Maximum axis acceleration (FSM)	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	EnablMovePoStop	
Macoda ID	1010 00030	
Data type	Acc_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

25.2.4 Torque limit value

Torque limit value	MaxTorqPoStop
<i>Specifies for each machining axis Dr[i]</i>	
- which maximum torque t S-0-0092 can be used to move to positive stop ("FsMove" (FSM)). - which maximum torque t S-0-0092 can be used to move the measuring axis to positive stop ("FsProbe" (FSP)).	



The drive has to be parameterized in a way that the value in S-0-0092 is interpreted as % of the nominal torque!

Path	MPS/Dr[i]/MaxTorqPoStop
Value range	0.00 .. 1000.00 0.00 .. 500.00 for MACODA compatibility
Pre-assignment	20.00
Unit	%
Description of parameter values	Torque limit value in % of M_N (temporarily in S-0-0092).
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	1003 00031
Data type	Torq_t (float, 2 decimal places) 4-byte float, 2 decimal places for MACODA compatibility

25.2.5 Maximum Permitted Position Deviation at Positive Stop

Maximum permitted position deviation at positive stop		MaxDevPoStop
Specifies for each machining axis Dr[i] the maximum position deviation at positive stop ("FsMove" (FSM)).		
Path	MPS/Dr[i]/MaxDevPoStop	
Value range	-1000000.000 .. 1000000.000 0.000 .. 9999.999 for MACODA compatibility	
Pre-assignment	0.100	
Unit	mm or ° (degrees)	
Description of parameter values	Maximum position deviation at positive stop	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1003 00032	
Data type	Pos_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

25.2.6 Position offset at positive stop

Position offset at positive stop		PosOffsPoStop
Specifies for each machining axis Dr[i] the maximum position offset at positive stop ("FsMove" (FSM)).		
Path	MPS/Dr[i]/PosOffsPoStop	
Value range	0.000 .. 1000.000 0.000 .. 1000.000 for MACODA compatibility	

Traveling to positive stop (MPS) - Optional parameters

Pre-assignment	0.100
Unit	mm or ° (degrees)
Description of parameter values	Position offset at positive stop. For hydraulic drives, the behavior at the positive stop can be improved by increasing the offset value.  - It is recommended to retain the default setting 0.100 for electric drives. - The offset value is limited to at a minimum of 0.010 by the MTX.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	1003 00033
Data type	Double0_1T_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

26 Splines (SPL) - Optional parameters

26.1 Overview

As opposed to linear interpolation, the function "Splines" (SPL) does not create straight lines between the data points, but curves instead.

The most essential settings of this data group are:

- start and end conditions for application-oriented spline machining in the NC program,
- deviations from path speed
- basic settings for c^2 -constant global cubic splines.

26.2 Spline parameters

26.2.1 Maximum permissible starting point/end point deviation

Maximum Allowed Starting Point/Endpoint Deviation	MaxPosDevSpline

Defines the maximum allowable deviation between the starting point of a spline block and the end point of the preceding block for spline coefficient programming.



The parameter is only relevant for spline coefficient programming with the initialization "SplineDef(1...5)".

Path	SPL/MaxPosDevSpline
Value range	0.010 .. 100.000 0.01 .. 99.99 for MACODA compatibility
Pre-assignment	0.1
Unit	µm
Description of parameter values	Allowable deviation between the programmed spline coefficient 0 of a coordinate and the end point of the preceding block. If the deviation is greater than this value, a runtime error is generated.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	8007 00010
Data type	Double0.01_100_t (double, 3 decimal places) 8 byte float, 2 decimal places for MACODA compatibility

26.2.2 Maximum permissible relative path velocity error

Maximum Allowable Deviation for the Path Velocity	MaxVelDevSpline

Specifies a dimensional tolerance for the deviation of the path velocity on the spline curve.

Splines (SPL) - Optional parameters

		With increasing precision, the calculation time needed in NC block preparation (block cycle time) rises.
Path	SPL/MaxVelDevSpline	
Value range	0.01 .. 100.00	
Pre-assignment	1.00	
Unit	%	
Description of parameter values	Permissible dimensional tolerance, relating to path velocity. The smaller the parameter value, the higher the approximation quality of the inverse path length function $w(s)$. If the parameter has the value 100%, $w(s)$ is approximated by a straight line, i.e. the spline interpolation is performed directly via the spline parameter w .	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	8007 00020	
Data type	Double0.01_100_t (double, 2 decimal places) 8 byte float for MACODA compatibility	

26.2.3 Degree of connection spline

Degree of connection spline		SplineType
Insertion of a connecting spline between two neighboring partial sequences based on a restricted look-ahead range.		
These splines are c^2 -continuous global cubic splines (type 2 splines).		
Path	SPL/SplineType	
Value range	3	3rd degree connection spline
	4	4th degree connection spline
Pre-assignment	4	4th degree connection spline
Unit	-	
Description of parameter values	3	3rd-degree connecting spline will be inserted, c^1 -constant to preceding spline
	4	4th-degree connecting spline will be inserted, c^2 -constant to preceding spline
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	8007 00099	
Data type	int 4 byte integer for MACODA compatibility	

26.2.4 Overlapping of two partial sequences

Overlapping of two partial sequences		NofOverlapBlk
Specifies the number of NC blocks that overlap between two spline partial sequences.		
These splines are c^2 -continuous global cubic splines (type 2 splines).		
Path	SPL/NofOverlapBlk	
Value range	Entire integer range	
Pre-assignment	4	
Unit	-	
Description of parameter values	Number of NC blocks that overlap.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	8007 00099	
Data type	int	
	4 byte integer for MACODA compatibility	

26.2.5 Spline variant for vector orientation

Spline Variant for Vector Orientation		VectOriSpline
Specifies spline variant for vector orientation.		
These splines are c^2 -continuous global cubic splines (type 2 splines).		
Path	SPL/VectOriSpline	
Value range	1	common spline for path and orientation movement
	2	independent splines for path and orientation movement
Pre-assignment	1	common spline for path and orientation movement
Unit	-	
Description of parameter values	1	Common spline for path and orientation movement
	2	Independent splines for path and orientation movement
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	8007 00099	
Data type	int	
	4 byte integer for MACODA compatibility	

26.2.6 Dimensional tolerance for B-spline approximation (default setting)

Dimensional tolerance for B-spline approximation (preset)	BSplineApproxTol

Splines (SPL) - Optional parameters

This parameter determines the presetting for the dimensional tolerance in b-spline approximation (SplineType = 4 (4th degree connection spline)). The value cannot be modified in the NC program.

A very large value results in corresponding deviations of b-spline curve from the approximating points.

Path	SPL/BSplineApproxTol
Value range	0.000 .. 10.000 0.000 .. 9.999 for MACODA compatibility
Pre-assignment	0.100
Unit	mm
Description of parameter values	Dimensional tolerance for distance from b-spline curve
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	8007 00030
Data type	Double0_10_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

27 Tangential tool guidance (TTL) -Optional parameters

27.1 General information

The data group *Tangential tool control (TTL)* contains channel-dependent parameters for

- [Channel-dependent tool axis definition, page 383](#)
- [Number of tool edges, page 383](#)
- [Angle settings for tool guidance, page 384](#)
- [Communication settings for PLC, page 387](#)

27.2 Definition of tool axes

27.2.1 System axis number of the rotary axis

System axis number of the rotary axis	RotAxTangT

Defines channel-specifically the system axis number of the rotary or endless axis that is to be used as rotary axis for the "Tangential tool guidance" function after startup.



The axis number can be modified in the part program by programming "TangTool" (TTL).

"Tangential tool guidance" guides a tool axis into a defined offset angle (see parameter "OffAngTangT" in the folder TTL/Ch[k]/) to the programmed path in the active plane.

Path	TTL/Ch[k]/RotAxTangT	
Value range	1 .. 250	
Pre-assignment	4	
Unit	-	
Description of parameter values	0	No system axis selected
	>0	Number of the system axis (Rotary axis = rotary axis or endless axis).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00210	
Data type	Dr1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

27.3 Number of tool edges

27.3.1 Number of edges

Number of edges	NofEdgeTangT

Defines the number of equally spaced tool edges that are provided on the tool used.

Tangential tool guidance (TTL) -Optional parameters

	This allows the control to optimize the necessary setting rotation of the tool.	
Path	TTL/Ch[k]/NofEdgeTangT	
Value range	0 .. 1000	
Pre-assignment	1	
Unit	-	
Description of parameter values	1	The tool is asymmetric and has only one tool edge. This edge is run along the contour with the offset angle taken into account.
	>1	The tool is symmetric, having the stated number of equally spaced edges. If there is a knee in the contour, the tool is rotated just enough for the nearest tool edge to be positioned at the necessary offset angle to contour. If, with rotary axes, this rotation would lead to a violation of the limit range, the control will try to use another tool edge.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00220	
Data type	Int0_1T_t (unsignedInt)	
	4 byte integer for MACODA compatibility	

27.4 Angle settings for tool guidance

27.4.1 Rotate the tool by 180 degrees for change in direction

Rotate tool 180 degrees for a reversal of direction		DirTangT
Has the effect that, irrespective of the offset angle, the tool is not rotated if a reversal of direction of movement (180-degree knee) occurs. Otherwise, the tool behaves as described for parameter "NofEdgeTangT".		
Path	TTL/Ch[k]/DirTangT	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No Tool rotation is possible.
	1	Yes The tool keeps its position (no rotation).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	

Macoda ID 7050 00220

Data type Byte0_1_t (byte)

4 byte integer for MACODA compatibility

Example:

Effect of the offset angle/reversal of direction in this example: 0 degrees

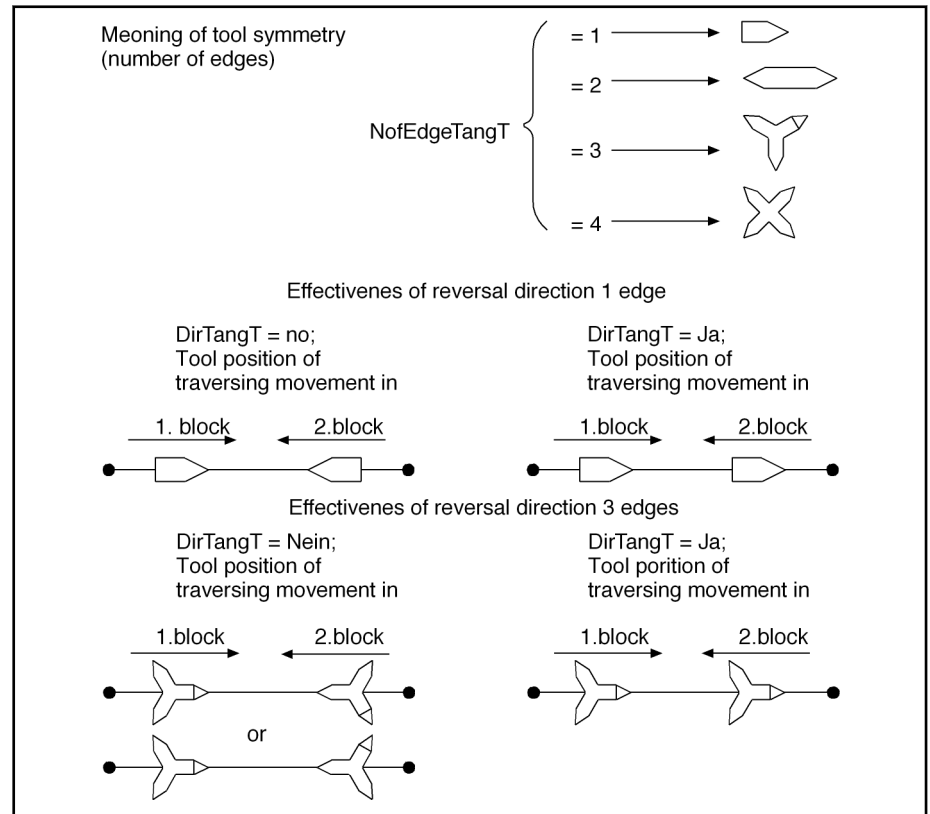


Fig. 27-1: Meaning of tool symmetry and reversal of direction

27.4.2 Intermediate block angle (after which an intermediate block is inserted)

Intermediate block angle (after which an intermediate block is inserted)	BlkAngTangT

Specifies from which knee angle between 2 contour elements (NC blocks) an intermediate block shall be inserted (see also parameter "NcPlcComTangT").



The intermediate block angle can be modified in the part program by programming "TangTool" (TTL).

In the execution of this intermediate block, the control rotates the tool axis until the offset angle required by the next block is reached. If no intermediate block is inserted, the tool axis jumps directly to its new position at the beginning of the next block.

Tangential tool guidance (TTL) -Optional parameters

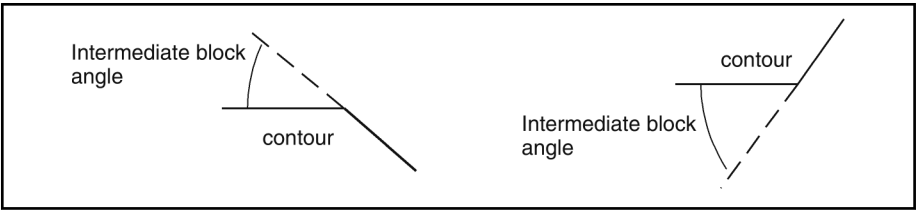


Fig. 27-2: Intermediate block angle

Path	TTL/Ch[k]/BlkAngTangT
Value range	0.000 .. 180.000
Pre-assignment	0.000
Unit	° (degrees)
Description of parameter values	Size of the intermediate block angle
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	7050 00230
Data type	Double0_180_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility

27.4.3 Offset angle (between the symmetry axis of the tool and the path tangent)

Offset angle (between the tool symmetry axis and the path tangent)	OffAngTangT

Specifies the offset angle between the symmetry axis of the tool and the path tangent.

 The offset angle can be modified in the part program by programming "TangTool" (TTL).

The figure below shows how the offset angle is determined:

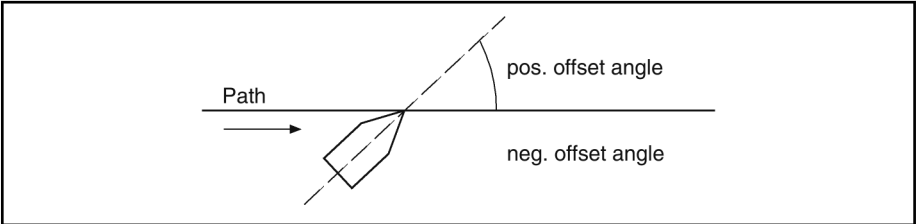


Fig. 27-3: Setting angle

Path	TTL/Ch[k]/OffAngTangT
Value range	-180.000 .. 180.000
Pre-assignment	0.000
Unit	° (degrees)
Description of parameter values	Size of the offset angle
Access rights	MTB: Write Setter: Read

	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00240	
Data type	Double-180_180_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

27.4.4 Starting angle for the offset angle

Starting angle for the offset angle		StartAngTangT
Specifies the value to which the offset angle is to be set upon activation of the "Tangential tool guidance" function.		
Path	TTL/Ch[k]/StartAngTangT	
Value range	0	Offset angle between the symmetry axis of the tool and the path tangent
	1	Current position
Pre-assignment	0	Offset angle between the symmetry axis of the tool and the path tangent
Unit	-	
Description of parameter values	0	Offset angle according to value in OffAngTangT
	1	Offset angle in accordance with the current position of the rotary axis.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00250	
Data type	Byte0_1_t (byte) 1 byte integer, unsigned for MACODA compatibility	

27.5 Communication settings for PLC

27.5.1 NC-PLC communication before executing an intermediate block

NC-PLC communication before an intermediate block is executed		NcPlcComTangT
Specify whether the NC and the PLC should communicate when executing an intermediate block in the "Tangential tool guidance" [also refer to the documentation "MTX 15VRS Programming Manual" (DOK-MTX***-NC**PRO*V15-RERS-EN-P)].		
This allows the PLC to take the necessary steps before the control rotates the tool to the required position.		
Path	TTL/Ch[k]/NcPlcComTangT	
Value range	0	No
	1	Yes
Pre-assignment	0	No

Tangential tool guidance (TTL) -Optional parameters

Unit	-	
Description of parameter values	0	No No NC-PLC communication. The intermediate block is executed unconditionally.
	1	Yes NC-PLC communication exists.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00260	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

28 Thread cutting (THRD) - Optional parameters

28.1 Overview

The data group Thread cutting (THRD) is where the following channel-specific Ch[k] settings are made:

- [Cutting mode, page 389](#)
- [Velocity, page 390](#)
- [Acceleration, page 391](#)
- [Deceleration, page 391](#)
- [Fast retract, page 392](#)

28.2 Cut mode (ModeSel)

28.2.1 Axis mode

Axis mode		AxModeThrd
Specifies in which axis mode "thread cutting" shall be performed.		
Path	THRD/Ch[k]/ModeSel/AxModeThrd	
Value range	0	cutting axes follow spindle actual position
	1	cutting axes follow spindle command position
Pre-assignment	0	cutting axes follow spindle actual position
Unit	-	
Description of parameter values	0	Cutting axes follow the spindle actual position.
	1	Cutting axes follow the spindle actual position. For this, the main spindle must be switched to position-controlled spindle operation with "ThreadSet(SPC1)".
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00600 [1]	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

28.2.2 Spindle mode

Spindle mode		SpModeThrd
The main spindle is switched to position-controlled spindle operation with "ThreadSet(SPC1)". The switching sequence is specified by SpModeThrd.		
Path	THRD/Ch[k]/ModeSel/SpModeThrd	
Value range	0	Enable spindle positioning interface at full speed.
	1	Enable spindle positioning interface at standstill.
Pre-assignment	1	Enable spindle positioning interface at standstill.
Unit	-	

Thread cutting (THRD) - Optional parameters

Description of parameter values	0	Enable spindle positioning interface at full speed. Attention: The spindle drive has to support direct switching to the position interface.
	1	Enable spindle positioning interface at standstill. The spindle is stopped; the position interface is enabled and subsequently, the spindle is accelerated to its initial speed again.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00600 [3]	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

28.3 Velocity (Vel)

28.3.1 Velocity jump

Velocity jump		VelJumpThrd
Specifies the maximum velocity jump at the beginning and at the end of the thread.		
 During spindle runtime, the parameter can be overridden by "ThreadSet(DYN(x, ,))".		
This parameter acts in conjunction with parameters "AccThrd" (in the folder THRD/Ch[k]/Acc) and "DecThrd" (in the folder THRD/Ch[k]/Dec).		
Path	THRD/Ch[k]/Vel/VelJumpThrd	
Value range	0.000 .. 100000.000	
	0.000 .. 99999.999 for MACODA compatibility	
Pre-assignment	1000.000	
Unit	mm/min	
Description of parameter values	Maximum velocity jump at the start and end of the thread.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00610	
Data type	JumpVel_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

28.4 Acceleration (Acc)

28.4.1 Path acceleration

Path acceleration		AccThrd
Specifies the path acceleration at the start of the thread.		
At the start of the thread, the velocity jumps to the active jump velocity and is subsequently increased to the cutting velocity (speed * thread pitch) according to the active path acceleration.		
 At runtime, this parameter can be overridden by "ThreadSet(DYN(,y,))".		
Path	THRD/Ch[k]/Acc/AccThrd	
Value range	0.000 .. 1000.000	
	0.000 .. 99.999 for MACODA compatibility	
Pre-assignment	0.000	
Unit	m/s ²	
Description of parameter values	0.000 The NC autonomously calculates the path acceleration. >=0.001 The set path acceleration and the velocity jump selected in parameter "VelJumpThrd" (in Ordner THRD/Ch[k]/Vel) are active.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00615	
Data type	Acc_t (double, 3 decimal places) 8 byte float, 3 decimal places for MACODA compatibility	

28.5 Deceleration (Dec)

28.5.1 Path deceleration acceleration

Path deceleration		DecThrd
Specifies the path deceleration acceleration at the start of the thread.		
The cutting axis (axes) is (are) decoupled from the spindle at the end of the thread and decelerated to the active jump velocity according to the active deceleration.		
 During runtime, this parameter can be overridden by "ThreadSet(DYN(, ,z))".		
Path	THRD/Ch[k]/Dec/DecThrd	
Value range	0.000 .. 1000.000	
	0.000 .. 99.999 for MACODA compatibility	
Pre-assignment	0.000	

Thread cutting (THRD) - Optional parameters

Unit	m/s ²	
Description of parameter values	0.000	The NC autonomously calculates the path deceleration acceleration.
	>=0.001	The set deceleration acceleration and the velocity jump selected in parameter "VelJumpThrd" (in Ordner THRD/Ch[k]/Vel) are active.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00620	
Data type	Acc_t (double, 3 decimal places)	
	8 byte float, 3 decimal places for MACODA compatibility	

28.6 Fast retract (Retr)

28.6.1 Activation fast retract

Enable fast retract		EnablRetrThrd
 During spindle runtime, this parameter can be overridden by "ThreadSet(ROn1)" and "ThreadSet(ROn0)".		
Enables or disables "Fast retract" for thread cutting. The data of the retract motion are specified in the "axis number of the retract axis" "DrIndRetrA"x (in the THRD/Ch[k]/Retr/ folder) and "retract vector of the retract axis (main axis)" "RetrVectRetrAx" is specified (in the THRD/Ch[k]/Retr/ folder).		
Path	THRD/Ch[k]/Retr/EnablRetrThrd	
Value range	0	No
	1	Yes
Pre-assignment	0	No
Unit	-	
Description of parameter values	0	No
		Fast retract disabled
	1	Yes
		Fast retract enabled
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00640	
Data type	Byte0_1_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

28.6.2 Axis number of retract axis

System axis number for the retract axis		DrIndRetrAx
Defines which system axis (main axis) is selected for "Fast retract".		
 The retract axes (main and secondary axis) must represent a plane within the channel Ch[k].		
Path	THRD/Ch[k]/Retr/DrIndRetrAx	
Value range	0 .. 250	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Fast retract not enabled
	>0	Index of the system axis number Dr[i] (main axis). The assigned numbers for main and secondary axes must be different (see parameter "DrIndSeconAx" (in the folder THRD/Ch[k]/Retr/)).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00645	
Data type	Dr_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

28.6.3 Retract vector of retract axis (main axis)

Retract vector of retract axis (main axis)		RetrVectRetrAx
Specifies what retract path (including sign = direction) the axis specified in "DrIndRetrAx" (in the THRD/Ch[k]/Retr/ folder) is to follow.		
 At runtime, this parameter can be overridden by "Thread-Set(RD(,))".		
Path	THRD/Ch[k]/Retr/RetrVectRetrAx	
Value range	-1000000.000 .. 1000000.000	
	-999.999 .. 999.999 for MACODA compatibility	
Pre-assignment	0.000	
Unit	mm	
Description of parameter values	Retract path and direction (= sign) of the main axis.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	DrIndRetrAx (in folder THRD/Ch[k]/Retr/)	
Macoda ID	7050 00650	

Thread cutting (THRD) - Optional parameters

Data type Pos_t (double, 3 decimal places)
4 byte float, 3 decimal places for MACODA compatibility

28.6.4 Axis number of the secondary axis

System axis number for the secondary axis	DrIndSeconAx

Defines which system axis (secondary axis) is selected for "Fast retract".



The retract axes (main and secondary axis) must represent a plane within the channel Ch[k].

Path	THRD/Ch[k]/Retr/DrIndSeconAx	
Value range	0 .. 250	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Fast retract not enabled
	>0	Index of the system axis number Dr[i] (secondary axis). The assigned numbers for main and secondary axes must be different (see parameter "DrIndRetrAx" (in the folder THRD/Ch[k]/Retr/)).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00645	
Data type	Dr_t (byte) 1 byte integer, unsigned for MACODA compatibility	

28.6.5 Retract vector of retract axis (secondary axis)

Retract vector of retract axis (secondary axis)	RetrVectSeconAx

Specifies what retract path (including sign = direction) the axis specified in "RetrVectSeconAx" (in the THRD/Ch[k]/Retr/ folder) is to follow.



At runtime, this parameter can be overridden by "Thread-Set(RD(,))".

Path	THRD/Ch[k]/Retr/RetrVectSeconAx		
Value range	-1000000.000 .. 1000000.000		
	-999.999 .. 999.999 for MACODA compatibility		
Pre-assignment	0.000		
Unit	mm		
Description of parameter values	Retract path and direction (= sign) of the secondary axis.		
Access rights	MTB:	Write	
	Setter:	Read	
	Operator:	Read	
Activation	System reset (SR)		

Dependency	DrIndRetrAx (in folder THRD/Ch[k]/Retr/)
Macoda ID	7050 00650
Data type	Pos_t (double, 3 decimal places) 4 byte float, 3 decimal places for MACODA compatibility

29 Tool and workpiece management (TLWP) - Optional parameters -

29.1 General information

The data group "Tool and workpiece management" (TLWP) is where tool correction calculations are set.

The tool corrections L1, L2 and L3 (TLen[1-3]) are configured in one of the 3 different coordinate systems:

- in the active coordinate system, i.e. tool corrections in active WCS working plane (WcsActPlane),
- in the workpiece coordinate system, i.e. tool corrections in WCS coordinates - fixed assignment - (WcsFixed), or
- in the tool coordinate system, i.e. tool corrections in TCS coordinates - fixed assignment - (Tcs)

For this, select the **TLWP/TCorr/Ch[k]/** folder and, with the right mouse button, select the appropriate entry with the function "Convert to..." from the context menu . WcsActPlane, WcsFixed or Tcs.

Depending on the entry, the corresponding subfolders of the folder TLWP/TCorr/Ch[k]/ will be presented. By default, the workpiece coordinate system (WcsFixed) is set.

Macoda ID 7050 01300 (1 = WcsActPlane, 2 = WcsFixed, 3 = Tcs)



The selection of the coordinate systems is channel-specific; i.e. a different coordinate system can be defined for each channel for tool correction calculations.

29.2 Tool correction calculation

29.2.1 Tool corrections in the active WCS working plane (WcsActPlane)

Tool corrections in the active WCS working plane	WcsActPlane

If the entry "WcsActPlane" has been selected from the context menu of the folder TLWP/TCorr/Ch[k]/, the tool length corrections L1, L2 and L3 relate to the active working plane of the active coordinate system in which the workpiece is being machined; i.e.

- L1 to the main coordinates
- L2 to the secondary coordinates
- L3 to the vertical coordinates

In this case, the parameters "ChAxTCorrWcs" and "DirTCorrWcs" (in the folder TLWP/TCorr/Ch[k]/...) are of no relevance.

29.2.2 Tool corrections in WCS coordinates-fixed assignment - (WcsFixed)

Tool corrections in WCS coordinates -Fixed assignment-	WcsFixed

The tool length corrections relate to the active workpiece coordinate system (WCS).

29.2.3 Channel axis/Logic axis

Channel axis/Logic axis	ChAxTCorrWcs

Tool and workpiece management (TLWP) - Optional parameters -

	Specifies for tool length correction L1 (TLen[1]), L2 (TLen[2]) or L3 (TLen[3]) the channel axis (specified in CHAN/Ch[k]/Acs/ChAx[1..8]/ChAxName) that the respective correction shall be applied to.	
Path	TLWP/TCorr/Ch[k]/WcsFixed/TLen[1-3]/ChAxTCorrWcs	
Value range	0 .. 8	
Pre-assignment	1	.../TLen[1]/ChAxTCorrWcs
	2	.../TLen[2]/ChAxTCorrWcs
	3	.../TLen[3]/ChAxTCorrWcs
Unit	-	
Description of parameter values	Number of the channel axis that the respective tool length correction L1, L2, L3 is active for.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 01310	
Data type	ChAx_t (unsignedByte)	
	4 byte integer, signed for MACODA compatibility	

29.2.4 Correction direction

Direction of correction		DirTCorrWcs
	Specifies the direction of each tool length correction TLen[1-3] in the workpiece coordinate system (WCS) in which the correction shall act in.	
Path	TLWP/TCorr/Ch[k]/WcsFixed/TLen[1..3]/DirTCorrWcs	
Value range	0	Positive
	1	Negative
Pre-assignment	0	Positive
Unit	-	
Description of parameter values	0	Positive
		Tool length correction TLen[1-3] acts in positive direction
	1	Negative
		Tool length correction TLen[1-3] acts in negative direction
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 01310	
Data type	Byte_t (byte)	
	4 byte integer, signed for MACODA compatibility	

29.2.5 Tool corrections in TCS-coordinates - Fix assignment - (Tcs)

Tool corrections in TCS coordinates - Fix assignment-	Tcs

The tool length corrections relate to the current tool coordinate system (TCS).

29.2.6 TCS coordinate

	TCS coordinate	CoordTCorrTcs
	Specifies for tool length correction L1 (TLen[1]), L2 (TLen[2]) or L3 (TLen[3]) the TCS coordinates that the respective correction shall be applied to.	
Path	TLWP/TCorr/Ch[k]/Tcs/TLen[1-3]/CoordTCorrTcs	
Value range	0 (-) 1 (x) 2 (y) 3 (z)	
Pre-assignment	1 (x) .../TLen[1]/CoordTCorrTcs 2 (y) .../TLen[2]/CoordTCorrTcs 3 (z) .../TLen[3]/CoordTCorrTcs	
Unit	-	
Description of parameter values	0 (-) The respective tool length correction L1, L2, L3 does not act on any TCS coordinates 1 (x), 2 (y), 3 (z) TCS coordinates that respective tool length correction L1, L2, L3 is active for.	
	 When the tool length corrections L1, L2, L3 are assigned to TCS coordinates, a TCS coordinate may not be assigned more than once.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 01310	
Data type	Byte0_8_t (byte)	
	4 byte integer, signed for MACODA compatibility	

29.2.7 Correction direction

	Direction of correction	DirTCorrTcs
	Specifies the direction of each tool length correction TLen[1..3] in the tool coordinate system (TCS) in which the correction shall act in	
Path	TLWP/TCorr/Ch[k]/Tcs/TLen[1..3]/DirTCorrTcs	
Value range	0 Positive 1 Negative	
Pre-assignment	0 Positive	
Unit	-	
Description of parameter values	0 Positive Tool length correction TLen[1-3] acts in positive direction 1 Negative Tool length correction TLen[1-3] acts in negative direction	

Tool and workpiece management (TLWP) - Optional parameters -

Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 01310	
Data type	byte_t (byte)	
	4 byte integer, signed for MACODA compatibility	

30 Axis transformation (TRA) - Optional parameters

30.1 General information

In the data group "Axis transformations" (TRA), up to 20 axis transformations AxTrafo[1..20] can be configured.

The most essential settings are:

- [Axis transformation type, page 401](#)
- [Input axes/coordinates of forward transformation, page 403](#)
- [Joint length and angle parameters, page 404](#)
- [Axis/machine positions of reference position, page 404](#)
- [Axis classification of system axes/coordinates, page 405](#)
- [Epsilon environment, page 407](#)
- [Reference orientation of tool coordinate system TCS, page 407](#)



For further information about the parameter values for axis transformation, refer to "MTX Functional Description - Extension" (DOK-MTX***-NC*F*EX*V15-RERS-EN-P) and "MTX 15VRS Programming Manual" (DOK-MTX***-NC**PRO*V15-RERS-EN-P).

30.2 Axis transformation type (TrafoType)

Axis transformation type		TrafoType
Defines the axis transformation type for each axis transformation AxTrafo[1..20]		
Path	TRA/AxTrafo[1..20]/TrafoType	
Value range	0	(-)
	0032002	(5 axis transformation ALLLB, only tool correction)
	2002001	(RR axis transformation without BCS orientation)
	2002002	(RR axis transformation with BCS orientation)
	2011001	(Face transformation (x,y))
	2011003	(2-axis polar kinematics)
	2020001	[non-orthogonal axis transformation (X,Y)]
	2020002	[non-orthogonal axis transformation (Z,X,)]
	2103001	(RRR axis transformation with BCS orientation)
	2103002	(RRR 2D-delta kinematics with additional rotary axis in the tip)
	2103011	(SCARA axis transformation)
	2112001	(Scara on linear axis (3-axis))
	2121001	[Bipod (X,Y,PHI)]
	2121002	(3-axis transformation LLR, with linear orientation)
	3021001	(3-axis transformation, cylinder jacket unwinding)
	3021002	(3-axis transformation, cylinder jacket, project planning)
	3021003	(Face transformation (x,y,z))
	3021004	(3-axis transformation LLR, Cartesian)

Axis transformation (TRA) - Optional parameters

	3021006	(3-axis polar kinematics)
	3030001	[non-orthogonal axis transformation (X,Y,Z)]
	3030013	(5-axis transformation RRLLL less equipped)
	3032001	(5-axis transformation LLLRR, rotary axis orientation)
	3032101	(Up to MTX14V14: 5-axis transformation LLLRR, rotary axis orientation)
	3033001	(6-axis transformation LLLRRR, rotary axis orientation)
	3033101	(Up to MTX14V16: 6-axis transformation LLLRR, rotary axis orientation)
	3113011	(4-axis Scara)
	3122001	[Scara on linear axis (4-axis)]
	3131001	[Bipod axis transformation (X,Y,Z,PHI)]
	3131002	(4-axis transformation LLLR)
	3131003	(4-axis transformation RLLL)
	3131021	(4-axis transformation, cylinder jacket unwinding)
	3131022	(4-axis transformation, cylinder jacket, project planning)
	3232101	(5-axis transformation LLLRR, linear orientation)
	3232102	(5-axis transformation RLLLR, linear orientation)
	3232103	(5-axis transformation RRLLL, linear orientation)
	3232111	(5-axis transformation LLLRR, Cardanic, linear orientation)
	3232112	(5-axis transformation RLLLR, Cardanic, linear orientation)
	3232121	(5-axis transformation, cylinder jacket unwinding, linear orientation)
	3232122	(5-axis transformation, cylinder jacket project planning, linear orientation)
	3232201	(5-axis transformation LLLRR, standard, vector orientation)
	3232202	(5-axis transformation RLLLR, vector orientation)
	3232203	(5-axis transformation RRLLL, vector orientation)
	3232211	(5-axis transformation LLLRR, Cardanic, vector orientation)
	3232212	(5-axis transformation RLLLR, Cardanic, vector orientation)
	3232221	(5-axis transformation, cylinder jacket unwinding, vector orientation)
	3232222	(5-axis transformation, cylinder jacket projection, vector orientation)
	3250201	Tripod with wobble plate
	3333301	(6-axis transformation LLLRRR, tensor orientation)
Pre-assignment	3232201	(5-axis transformation LLLRR, standard vector orientation) TRA/AxTrafo[1]/TrafoType
	3032101	(5-axis transformation LLLRR, rotary axis orientation) TRA/AxTrafo[2]/TrafoType
	0	(-) TRA/AxTrafo[3..20]/TrafoType
Unit	-	
Description of parameter values	0	No axis transformation enabled

	>0	Selected axis transformation enabled: see value range for name
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1030 00110	
Data type	int	
	4 byte integer, unsigned for MACODA compatibility	

30.3 System axes/coordinates of forward transformation (FwdInCoord)

30.3.1 System axis coordinate index (SACI)[1..8]

System axis/coordinates index (SACI) [1..8]		FwdInCoordIndTrafo
	Defines the input axes of the forward transformation if the transformation belongs to the point of action AT1 or defines the input axes or input coordinates or both for the forward transformation if the transformation belongs to the point of action AT2. The number of input axes and coordinates is defined by the axis transformation type (TrafoType). The index of the respective system axis or system coordinate is parameterized.	
Path	TRA/AxTrafo[1..20]/FwdInCoordIndTrafo[1..8]	
Value range	-6, -5, -4, -3, -2, -1	Index of the system coordinate
	0	No axis
	1..250	Index of the system axis
Pre-assignment	0	
Unit	-	
Description of parameter values	1..250	Index of the system axis for forward transformation
	-6, -5, -4, -3, -2, -1	Index of the system coordinate for forward transformation
		-6: Linear machine coordinate X
		-5: Linear machine coordinate Y
		-4: Linear machine coordinate Z
		-3: Rotary machine coordinate PHI
		-2: Rotary machine coordinate THETA
		-1: Rotary machine coordinate PSI
	0	No axis defined for forward transformation
	Otherwise	<not supported>
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	

Axis transformation (TRA) - Optional parameters

Dependency	-
Macoda ID	1030 00120
Data type	Short_t (2 bytes) 2 byte integer, signed for MACODA compatibility

30.4 Joint length and angle parameters (JointPar)

30.4.1 Length parameter/angle parameter

Length/Angle parameters		JointParTrafo[1..16]
Defines the geometry for each axis transformation AxTrafo[1..20], i.e. the distances and angles JointParTrafo[1..16] of the transformation axes in the reference position.		
 The meaning of the individual parameters is defined by the axis transformation type "TrafoType" (in the folder TRA/AxTrafo[1..20]).		
Path	TRA/AxTrafo[1..20]/JointPar/JointParTrafo[1..16]	
Value range	-1000000.0000000 .. 1000000.0000000	
Pre-assignment	100.0000000	TRA/AxTrafo[1..2]/JointPar/JointParTrafo[1]
	-100.0000000	TRA/AxTrafo[1..2]/JointPar/JointParTrafo[6]
	0.0000000	TRA/AxTrafo[1..2]/JointPar/JointParTrafo[2..5]
		TRA/AxTrafo[1..2]/JointPar/JointParTrafo[7..16]
		TRA/AxTrafo[3..20]/JointPar/JointParTrafo[1..16]
Unit	mm or ° (degrees)	
Description of parameter values	Length/angle parameters JointParTrafo[1..16] for joint parameters of the axis transformation AxTrafo[1..20].	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1030 00140	
Data type	Pos_t (double, 7 decimal places)	
	8 byte float, 4 decimal places for MACODA compatibility	

30.5 Axis/machine positions of reference position (RefPos)

30.5.1 Reference position[1..8]

Reference position[1..8]	RefPosTrafo[1..8]
Specifies the reference position during an active axis transformation AxTrafo[1..20] for each reference position[1..8]:	
TCP position: x=0, y=0, z=0 and	
TCS orientation: Phi = 0, Theta = 0, Psi = 0.	



The meaning of the individual parameters is defined by the axis transformation type "TrafoType" (in the folder TRA/AxTrafo[1..20]/).

Path	TRA/AxTrafo[1..20]/RefPos/RefPosTrafo[1..8]		
Value range	-1000000.000 .. 1000000.000		
Pre-assignment	-100.000	TRA/AxTrafo[1..20]/RefPos/RefPosTrafo[1]	
	0.000	TRA/AxTrafo[1..20]/RefPos/RefPosTrafo[2]	
	-100.000	TRA/AxTrafo[1..20]/RefPos/RefPosTrafo[3]	
	0.000	TRA/AxTrafo[1..20]/RefPos/RefPosTrafo[4..8]	
Unit	mm or ° (degrees)		
Description of parameter values	Reference position of transformation axis		
Access rights	MTB:	Write	
	Setter:	Read	
	Operator:	Read	
Activation	System reset (SR)		
Dependency	-		
Macoda ID	1030 00130		
Data type	Pos_t (double, 3 decimal places)		
	8 byte float, 3 decimal places for MACODA compatibility		

30.6 Axis classification of system axes/coordinates (FwdInCoord-Class)

30.6.1 System axis/system coordinate[1..8]

Classification system axis/coordinate

System axis/Coordinate classification	CoordClass
Defines each system axis/coordinate FwdInCoord[1..8]/CoordClass involved in the transformation AxTrafo[1..20].	



The meaning of the individual parameters is defined by the axis transformation type "TrafoType" (in the folder TRA/AxTrafo[1..20]/).

Path	TRA/AxTrafo[1..20]/FwdInCoordClass/FwdInCoord[1..8]/CoordClass	
Value range	0	(no technical relevance)
	1	(X)
	2	(Y)
	3	(Z)
	100	[A (rotary)]
	200	[B (rotary)]
	300	[C (rotary)]
	111	[A (endless, no NC axis)]
	222	[B (endless, no NC axis)]

Axis transformation (TRA) - Optional parameters

	333	[C (endless, no NC axis)]
Pre-assignment	0	(no technical relevance)
Unit	-	
Description of parameter values	0	(no technical relevance) System axis has no technical relevance to the transformation.
	Otherwise	Technical significance of the system axis for the transformation. The following values are relevant only to the less-equipped 5-axis transformation of type 3030013; these entries are not permitted in any other transformations: 111 (A (endless, no NC axis)), 222 (B (endless, no NC axis)) and 333 (C (endless, no NC axis))
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1030 00125	
Data type	int	
	4 byte integer, signed for MACODA compatibility	

Motion direction

Direction of movement		CoordDir
Defines the direction of traveling or rotation of each system axis/coordinate FwdInCoord[1..8] involved in the transformation AxTrafo[1..20].		
 The meaning of the individual parameters is defined by the axis transformation type "TrafoType" (in the folder TRA/AxTrafo[1..20]).		
Path	TRA/AxTrafo[1..20]/FwdInCoordClass/FwdInCoord[1..8]/CoordDir	
Value range	0	Positive
	1	Negative
Pre-assignment	0	Positive
Unit	-	
Description of parameter values	0	Positive Positive direction of traveling or rotation of the system axis/coordinate FwdInCoord[1..8]
	1	Negative Negative direction of traveling or rotation of the system axis/coordinate FwdInCoord[1..8]
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Activation	System reset (SR)
Dependency	-
Macoda ID	1030 00125
Data type	Byte0_1_t (byte) 4 byte integer, signed for MACODA compatibility

30.7 Epsilon environment

Epsilon environment	EpsAxTrafo

Defines capture ranges around singularities of axis transformations AxTrafo[1..20].



The parameter is of relevance to 5 axis transformations with vector orientation (Type 3232201: 5-axis transformation LLLRR, standard vector orientation and type 3232211: 5-axis transformation LLLRR cardanic vector orientation).

For more information, refer to "MTX 15VRS Functional Description - Extension" (DOK-MTX***-NC*F*EX*V15-RERS-EN-P) under "Vector orientation".

Path	TRA/AxTrafo[1..20]/EpsAxTrafo
Value range	-9999.999 .. 9999.999
Pre-assignment	0.000
Unit	mm or ° (degrees)
Description of parameter values	Epsilon value for capture range
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	1030 00160
Data type	float (3 decimal places) 4 byte float, 3 decimal places for MACODA compatibility

30.8 Reference orientation of the tool coordinate system TCS(TcsRefOri)

30.8.1 TCS coordinate X

Direction of TCS_X in BCS

Direction of TCS_X in BCS	CoordTcsXBcs

Specifies the "direction" of the tool coordinates TCS_X (in the tool coordinate system TCS) that shall run parallel to one of the coordinates x, y and z of the basic workpiece coordinate system BCS in the reference position of the corresponding active axis transformation AxTrafo[1..20] (see also Parameter "RefPosTrafo[1..8]" in the folder TRA/AxTrafo[1..20]/RefPos/).

Path	TRA/AxTrafo[1..20]/TcsRefOri/TcsX/CoordTcsXBcs

Axis transformation (TRA) - Optional parameters

Value range	0 (-) 1 (x) 2 (y) 3 (z)
Pre-assignment	1 (x)
Unit	-
Description of parameter values	Direction of the TCS coordinate X 0 (-) no direction specified 1 (x) in x direction of the BCS. 2 (y) in y direction of the BCS. 3 (z) in z-direction of the BCS.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	-
Macoda ID	1030 00150
Data type	Byte0_3_t (byte) 1 byte integer, signed for MACODA compatibility

Sign of TCS_X in BCS

Coordinate of TCS_X in BCS		DirTcsXBcs
Specifies the "sign of the direction" of the tool coordinates TCS_X (in the tool coordinate system TCS) that shall run parallel to one of the coordinates x, y and z of the basic workpiece coordinate system BCS in the reference position of the corresponding active axis transformation AxTrafo[1..20] (see also Parameter "RefPosTrafo[1..8]" in the folder TRA/AxTrafo[1..20]/RefPos/).		
Path	TRA/AxTrafo[1..20]/TcsRefOri/TcsX/DirTcsXBcs	
Value range	0	Positive
	1	Negative
Pre-assignment	0	Positive
Unit	-	
Description of parameter values	0	Positive Direction of the TCS coordinate X in "positive" direction with regard to the respective basic tool coordinate of the BCS, set in parameter "CoordTcsXBcs" (in folder TRA/AxTrafo[1..20]/TcsRefOri/TcsX/).
	1	Negative Direction of the TCS coordinate X in "negative" direction with regard to the respective basic tool coordinate of the BCS, set in parameter "CoordTcsXBcs" (in folder TRA/AxTrafo[1..20]/TcsRefOri/TcsX/).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read

Activation	System reset (SR)
Dependency	-
Macoda ID	1030 00150
Data type	Byte0_1_t (byte) 1 byte integer, signed for MACODA compatibility

30.8.2 TCS coordinate Y

Direction of TCS_Y in BCS

Direction of TCS_Y in BCS		CoordTcsYBcs
Specifies the "direction" of the tool coordinates TCS_Y (in the tool coordinate system TCS) that shall run parallel to one of the coordinates x, y and z of the basic workpiece coordinate system BCS in the reference position of the corresponding active axis transformation AxTrafo[1..20] (see also Parameter "RefPosTrafo[1..8]" in the folder TRA/AxTrafo[1..20]/RefPos/).		
Path	TRA/AxTrafo[1..20]/TcsRefOri/TcsY/CoordTcsYBcs	
Value range	0 (-) 1 (x) 2 (y) 3 (z)	
Pre-assignment	2 (y)	
Unit	-	
Description of parameter values	Direction of the TCS coordinate Y 0 (-) no direction specified 1 (x) in x direction of the BCS. 2 (y) in y direction of the BCS. 3 (z) in z-direction of the BCS.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1030 00150	
Data type	Byte0_3_t (byte) 1 byte integer, signed for MACODA compatibility	

Sign of TCS_Y in BCS

Coordinate of TCS_Y in BCS		DirTcsYBcs
Specifies the "sign of the direction" of the tool coordinates TCS_Y (in the tool coordinate system TCS) that shall run parallel to one of the coordinates x, y and z of the basic workpiece coordinate system BCS in the reference position of the corresponding active axis transformation AxTrafo[1..20] (see also Parameter "RefPosTrafo[1..8]" in the folder TRA/AxTrafo[1..20]/RefPos/).		
Path	TRA/AxTrafo[1..20]/TcsRefOri/TcsY/DirTcsYBcs	
Value range	0 Positive 1 Negative	

Axis transformation (TRA) - Optional parameters

Pre-assignment	0	Positive
Unit	-	
Description of parameter values	0	Positive
		Direction of the TCS coordinate Y in "positive" direction with regard to the respective basic tool coordinate of the BCS, set in parameter "CoordTcsYBcs" (in folder TRA/AxTrafo[1..20]/TcsRefOri/TcsY/).
	1	Negative
		Direction of the TCS coordinate Y in "negative" direction with regard to the respective basic tool coordinate of the BCS, set in parameter "CoordTcsYBcs" (in folder TRA/AxTrafo[1..20]/TcsRefOri/TcsY/).
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1030 00150	
Data type	Byte0_1_t (byte)	
	1 byte integer, signed for MACODA compatibility	

30.8.3 TCS coordinate Z

Direction of TCS_Z in BCS

Direction of TCS_Z in BCS		CoordTcsZBcs
	Specifies the "direction" of the tool coordinates TCS_Z (in the tool coordinate system TCS) that shall run parallel to one of the coordinates x, y and z of the basic workpiece coordinate system BCS in the reference position of the corresponding active axis transformation AxTrafo[1..20] (see also Parameter "RefPosTrafo[1..8]" in the folder TRA/AxTrafo[1..20]/RefPos/).	
Path	TRA/AxTrafo[1..20]/TcsRefOri/TcsZ/CoordTcsZBcs	
Value range	0 (-)	
	1 (x)	
	2 (y)	
	3 (z)	
Pre-assignment	3 (z)	
Unit	-	
Description of parameter values	Direction of the TCS coordinate Z	
	0 (-)	no direction specified
	1 (x)	in x direction of the BCS.
	2 (y)	in y direction of the BCS.
	3 (z)	in z-direction of the BCS.
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	

Dependency	-
Macoda ID	1030 00150
Data type	Byte0_3_t (byte)
	1 byte integer, signed for MACODA compatibility

Sign of TCS_Z in BCS

	Coordinate of TCS_Z in BCS	DirTcsZBcs
	Specifies the "sign of the direction" of the tool coordinates TCS_Z (in the tool coordinate system TCS) that shall run parallel to one of the coordinates x, y and z of the basic workpiece coordinate system BCS in the reference position of the corresponding active axis transformation AxTrafo[1..20] (see also Parameter "RefPosTrafo[1..8]" in the folder TRA/AxTrafo[1..20]/RefPos/).	
Path	TRA/AxTrafo[1..20]/TcsRefOri/TcsZ/DirTcsZBcs	
Value range	0	Positive
	1	Negative
Pre-assignment	0	Positive
Unit	-	
Description of parameter values	0	Positive
	Direction of the TCS coordinate Z in "positive" direction with regard to the respective basic tool coordinate of the BCS, set in parameter "CoordTcsZBcs" (in folder TRA/AxTrafo[1..20]/TcsRefOri/TcsZ/).	
	1	Negative
	Direction of the TCS coordinate Z in "negative" direction with regard to the respective basic tool coordinate of the BCS, set in parameter "CoordTcsZBcs" (in folder TRA/AxTrafo[1..20]/TcsRefOri/TcsZ/).	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	1030 00150	
Data type	Byte0_1_t (byte)	
	1 byte integer, signed for MACODA compatibility	

31 Velocity-dependent output (VDO) - Optional parameters

31.1 General information

The velocity-dependent output (VDO) is used for laser power control.

The following settings can be made:

- [Limit velocities, page 413](#)
- [Command value at analog output, page 414](#)
- [Coordinate selection for path velocity calculation, page 415](#)

31.2 Limit velocities (Vel)

31.2.1 Upper velocity limit value

Upper velocity limit value	UpVelLim

Defines for each channel Ch[k] the upper velocity limit for laser power control.



This parameter only has an effect in conjunction with "LowVelLim" (in the folder VDO/Ch[k]/Vel/).

Path	VDO/Ch[k]/Vel/UpVelLim
Value range	0.000 .. 10000000000.000 0.0 .. 99999.9 for MACODA compatibility
Pre-assignment	1000.000
Unit	mm/min
Description of parameter values	Upper velocity limit value: For speeds exceeding this value, the upper power limit value is output at the analog output.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	LowVelLim (in folder VDO/Ch[k]/Vel/)
Macoda ID	7050 00815 [2]
Data type	Vel_t (double, 3 decimal places) 4 byte float, 1 decimal place for MACODA compatibility

31.2.2 Lower velocity limit value

Lower velocity limit value	LowVelLim

Defines for each channel Ch[k] the lower velocity limit for laser power control.



This parameter only has an effect in conjunction with "UpVelLim" (in the folder VDO/Ch[k]/Vel/).

Path	VDO/Ch[k]/Vel/LowVelLim
Value range	0.000 .. 10000000000.000

Velocity-dependent output (VDO) - Optional parameters

	0.0 .. 99999.9 for MACODA compatibility
Pre-assignment	0.000
Unit	mm/min
Description of parameter values	Lower velocity limit value: For speeds below this value, the lower power limit value is output at the analog output.
Access rights	MTB: Write Setter: Read Operator: Read
Activation	System reset (SR)
Dependency	UpVelLim (in folder VDO/Ch[k]/Vel/)
Macoda ID	7050 00815 [1]
Data type	Vel_t (double, 3 decimal places) 4 byte float, 1 decimal place for MACODA compatibility

31.3 Command value at the analog output (Out)

31.3.1 Command value when reaching the upper velocity limit value

Command value when reaching the upper velocity limit value		AnaOutUpVelLim
Defines for each channel Ch[k] the upper power limit for laser power control that shall apply in conjunction with the velocity limits ("LowVelLim" and "UpVelLim" (in the folder VDO/Ch[k]/Vel/).		
The laser power is adjusted according to the path velocity and kept between the lower and upper power limit values.		
Path	VDO/Ch[k]/Out/AnaOutUpVelLim	
Value range	0.00 .. 100.00	
Pre-assignment	100.00	
Unit	%	
Description of parameter values	Upper power limit value in %. 100% corresponds to a voltage of 10 V at the analog output.	
Access rights	MTB: Write Setter: Read Operator: Read	
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00810 [2]	
Data type	Float0_100_t (float, 2 decimal places) 4 byte float, 1 decimal place for MACODA compatibility	

31.3.2 Command value when reaching the lower velocity limit value

Command value when the lower velocity limit is reached	AnaOutLowVelLim
Defines for each channel Ch[k] the lower power limit for laser power control that shall apply in conjunction with the velocity limits ("LowVelL" and "UpVelL" (in the folder VDO/Ch[k]/Vel/).	

	The laser power is adjusted according to the path velocity and kept between the lower and upper power limit values.	
Path	VDO/Ch[k]/Out/AnaOutLowVelLim	
Value range	0.00 .. 100.00	
Pre-assignment	0.00	
Unit	%	
Description of parameter values	Lower power limit value in %.	
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00810 [1]	
Data type	Float0_100_t (float, 2 decimal places)	
	4 byte float, 1 decimal place for MACODA compatibility	

31.4 Coordinate selection for path velocity calculation (CoordSel)

Coordinate selection for path velocity calculation		CoordSel
	Specifies whether the movements in the selected plane or in the selected space are taken into account for calculation of the actual speed.	
Path	VDO/Ch[k]/CoordSel	
Value range	0	No coordinates
	2	Active machining plane
	3	Active working space
Pre-assignment	2	Active machining plane
Unit	-	
Description of parameter values	0	No coordinate selection
	2	Active plane (corresponds to main and secondary axis)
	3	Active space (corresponds to main, secondary and normal axes)
Access rights	MTB:	Write
	Setter:	Read
	Operator:	Read
Activation	System reset (SR)	
Dependency	-	
Macoda ID	7050 00820	
Data type	Byte0_3_t (byte)	
	1 byte integer, unsigned for MACODA compatibility	

32 System data

32.1 General information

System data are variable data structures that can be accessed from anywhere in the system. They can be changed at runtime. If they are defined as "permanent", the content is retained even after a system restart.

The following internal system data structures are defined in the IndraMotion MTX:

- [chapter 32.2.1 "System Simulation /Sys_Simulation" on page 420](#)
- [chapter 32.2.2 "Status of bit events /SysActBitEv" on page 421](#)
- [chapter 32.2.3 "Control data of the function programmable distance-dependent signal /SysAdvDistSig" on page 421](#)
- [chapter 32.2.4 "Advanced control data of the function "Programmable distance-dependent signal"/SysAdvDistSigE" on page 428](#)
- [chapter 32.2.5 "Additional data of the function programmable distance-dependent signal /SysAdvDistSigA" on page 429](#)
- [chapter 32.2.6 "Automatic program selection after a system restart / SysAPrg" on page 431](#)
- [chapter 32.2.7 "Actual data of the system axes /SysAxActData" on page 434](#)
- [chapter 32.2.8 "System axis coupling command/SysAxCoupleCmd" on page 435](#)
- [chapter 32.2.9 "System axis coupling control/SysAxCoupleCtr" on page 453](#)
- [chapter 32.2.10 "System axis coupling status/SysAxCoupleSta" on page 456](#)
- [chapter 32.2.11 "System axis coupling tables/SysAxCoupleTab" on page 467](#)
- [chapter 32.2.12 "Block pre-run data for axes /SysAxSRun" on page 469](#)
- [chapter 32.2.13 "Data of the active axis transformation /SysAxTrafo1 and /SysAxTrafo2" on page 472](#)
- [chapter 32.2.13 "Data of the active axis transformation /SysAxTrafo1 and /SysAxTrafo2" on page 472](#)
- [chapter 32.2.15 "SysCfgInfo /SysCfgInfo" on page 478](#)
- [chapter 32.2.16 "Active channel spindle data /SysChSpActData" on page 478](#)
- [chapter 32.2.17 "Spindle pre-run data /SysChSpSRun" on page 481](#)
- [chapter 32.2.18 "Block pre-run data/SysChSRun" on page 506](#)
- [chapter 32.2.21 "Coordinate system display of the tool editor / SysCoordSystem" on page 524](#)
- [chapter 32.2.22 "System control /SysCtrl" on page 527](#)
- [chapter 32.2.23 "Control data for handwheel mode in WCS / SysHandwWCS" on page 527](#)
- [chapter 32.2.24 "System hardware status information /SysHwStates" on page 530](#)
- [chapter 32.2.25 "System interpolations axis positions/SysIpoAcsPos" on page 532](#)

- chapter 32.2.26 "System interpolation basic workpiece positions /SysIpoBcsPos" on page 533
- chapter 32.2.27 "System interpolations machine positions/SysIpoMc-sPos" on page 535
- chapter 32.2.28 "System interpolation path velocities/SysIpoPathFeed" on page 536
- chapter 32.2.29 "System interpolations workpiece positions/SysIpoWc-sPos" on page 538
- chapter 32.2.30 "NC kernel diagnostics /SysKernelDgn" on page 540
- chapter 32.2.31 "Maximum number of NC kernel-controlled drives / SysMaxAxNbr" on page 542
- chapter 32.2.32 "Number of edges /SysMaxEd displayed" on page 542
- chapter 32.2.33 "Control data for NC diagnostics /SysNcDgnCtrl" on page 543
- chapter "Commissioning data of the probe/SysPrBase" on page 546
- chapter "Parameters for manual probing/SysPrMan" on page 547
- chapter "Event data of the axis-parallel measurement /SysPrResult" on page 551
- chapter 32.2.35 "Return to path logic data /SysReentry" on page 552
- chapter 32.2.36 "Retrace logic data /SysRetrace" on page 557
- chapter 32.2.37 "List of the Sercos devices /SysScsDevice connected" on page 559
- chapter 32.2.38 "IP addresses of the Sercos interface /SysScsIpAdr" on page 560
- chapter 32.2.39 "Settings for automatic Sercos parameterization/ SysScsDrPar" on page 562
- chapter 32.2.40 "Control data to re-initialize selected SD elements after NC restart /SysSDForce" on page 563
- chapter 32.2.41 "Data of the memory space assignment of system data /SysSdInfo" on page 565
- chapter 32.2.42 "Actual data of system spindles /SysSpActData" on page 569
- chapter 32.2.43 "Command data for system spindles /SysSpCmdData" on page 570
- chapter 32.2.44 "Block pre-run data/SysSRun" on page 579
- chapter 32.2.45 "System status information /SysStates" on page 587
- chapter 32.2.46 "Data for cycle Swiveling G230 /SysSwivel" on page 589
- chapter 32.2.47 "Target information /SysTargetInfo" on page 593
- chapter 32.2.48 "Tool correction tool edge numbers/SysTlActMaxEd" on page 601
- chapter 32.2.49 "Active and preselected tool /SysTool" on page 602
- chapter 32.2.50 "Tool Corrections per edge /SysToolEd" on page 614
- chapter 32.2.51 "SysToolInsert /SysToolInsert" on page 617
- chapter 32.2.52 "Active and preselected tool at block pre-run / SysToolSRun" on page 618

- chapter 32.2.53 "Tool type data of tools in the tool catalog /SysTool-Type" on page 631
- chapter 32.2.54 "Variables for the tool change cycle TCH /SysTch" on page 633
- chapter 32.2.55 "Displaying the SRAM assignment /SysSRamMemInfo" on page 640
- chapter 32.2.55 "Displaying the SRAM assignment /SysSRamMemInfo" on page 640
- chapter 32.2.57 "Data of the zero point offsets (global) /SysZOData" on page 641
- chapter 32.2.58 "Data of the zero point offsets (axis-related) /SysZODataA" on page 643
- chapter 32.2.59 "Data of the zero point offsets (channel-related) /SysZODataC" on page 644



- For **permanent system data**, the memory in the static memory (SRAM) is limited to 512 Kbyte (524288 bytes) or to 1,25 Mbyte (1310720 bytes), depending on the hardware. It is generated and initialized once. Its content is retained even after a system restart.
- For **volatile system data**, the memory in the dynamic memory (DRAM) is limited to 2.4 Mbyte (2516584 bytes) (memory class **volatile**). It is generated and initialized again at every system restart.
- Of the system data memories, 75% are assigned to customer system data and 25% to the internal system data. If the memory for the customer system data is exceeded, a warning is output.
- If more than 240 axes and more than 55 channels are configured at the same time, 75% customer memory cannot be guaranteed for permanent system data. It can be reduced to up to one percent point.
- For a detailed description of the functional operation and application of system data, refer to the respective chapter in "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P).

In the following, the path in XML syntax (X-Path) is given for the access path of the corresponding system data. To access from the CPL, a different syntax has to be used.

Example:

Difference between X-path and CPL syntax

X-Path: /SysStates/BdTime/OpTime

CPL syntax: SD.SysStates.BdTime.OpTime

The type names defined in the corresponding XML scheme file are specified for the system data types. Depending on the access point (NC kernel, IndraLogic, CPL), other type declarations have to be used. The assignment of these data types is summarized in the "Managing schema files" chapter in "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P).

For some system data defined as arrays in the uppermost level, the array size is specified dynamically using keywords ("Channel", "Drive", "Spindle"). As a result, the array size has a maximum limit. This limit depends on the parameterization in the machine parameters.

The following applies for the array size n:

- "Channel": $1 \leq n \leq \text{/MAIN/NofCh}$
- "Drive": $1 \leq n \leq \text{/MAIN/NofDr}$
- "Spindle": $1 \leq n \leq (\text{number of available Sp[i] elements in /SP/SpFunc Sp[i]})$



- **Permanent system data** for which the data type or the sub-type of the variables was modified are automatically created and initialized the next time that the control is started. If necessary, these system data may have to be archived **before** restarting the system.
- **Permanent system data** defined as array with the keywords **Channel**, **Drive** or **Spindle**, are automatically generated and initialized after the next control startup if the number of channels, drives or spindles has changed due to changes in the machine parameters.

32.2

System data description

32.2.1

System Simulation "/Sys_Simulation"

Simulation

Simulation		Sys_Simulation
This variable can affect the program sequence if it is simulated using an emulation. Query this variable to create a program branch in the main program as well as in the subroutines to avoid instructions causing errors, program stand-still or unintended results during an emulation.		
The variable Sys_Simulation has no reference to the NC kernel. It is neither set nor evaluated by the kernel.		
Path	/Sys_Simulation[<Channel>] CPL: SD.Sys_Simulation[<Channel>]	
Value range	FALSE TRUE	
Pre-assignment	FALSE	
Unit	-	
Description of the variable values	TRUE	During the simulation
	FALSE	Remaining program runtime
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	This variable does not affect the NC kernel.	

Update	-
Dependency	-
Data type	Boolean_t (1 byte)

32.2.2 Status of bit events "/SysActBitEv"

Status of bit events

	Status of bit events	SysActBitEv
	Indicates the status of bit events as character encoding. There is "0" for disabled and "1" for enabled. They are place-coded (5th place -> 5th bit event). This system data is available at delivery and stored as volatile.	
Path	/SysActBitEv CPL: SD.SysActBitEv	
Value range	"0" or "1" as IsoLatin1String of 96 characters	
Pre-assignment	96 * "0"	
Unit	-	
Description of the variable values	Status of bit events	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	Is always active.	
Update	At every change of the bit events (e.g. SEV/REV)	
Dependency	-	
Data type	IsoLatin1String (100 bytes for text)	

32.2.3 Control data of the function programmable distance-dependent signal "/SysAdvDistSig"

General information

The system data structure SysAdvDistSig is created for each channel in the volatile memory.

It contains control data for the NC function "Programmable distance-dependent signal" ADS. In case of an active function (ADSENA(.)), a signal is set for a certain duration in a specific distance before a programmed mark (ADSLAB(.)).

The system data structure consists of two parts:

- General data
- Signal data

The following structural elements are defined in the "sdadvdistsig.xsd_" schema:

General data (SysAdvDistSig):

Structural element	Description
SignalStatus	Status of all signals (bit-coded)
Unit	Unit of the distance (mm or inch)

Tab. 32-1: Overview on structural elements SysAdvDistSig (general data)

Data of the signals (/SysAdvDistSig[CHANNEL]/Sig[1..8]):

Structural element	Description
DistOn	Distance to Set mark ADSLAB(<Signal>)
DistOff	Distance to Reset mark ADSLAB(-<Signal>)
TimeOn	Time to Set mark ADSLAB(<Signal>)
TimeOff	Time to Reset mark ADSLAB(-<Signal>)
DistCalc	Type of distance calculation
Duration	Signal duration
Type	Signal type
Index	Index of the PLC signal

Tab. 32-2: Overview on structural elements SysAdvDistSig (signal data)



For more information, refer to the "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P) in the chapter "Programmable distance-dependent signal".

Status of all signals

Status of all signals (bit-coded)	SignalStatus

In addition, to the PLC signals, the ADS function signalizes that the specified distance to the mark is fallen below.

Path /SysAdvDistSig[<Channel>]/SignalStatus

CPL: SD.SysAdvDistSig[<Channel>].SignalStatus

Value range

-

Pre-assignment

0

Unit

-

Description of the variable values

Provides an overview of the currently set signals.

Signal 1 has the valence 1

Signal 2 has the valence 2

Signal 3 has the valence 4

Signal 4 has the valence 8

Signal 5 has the valence 16

Signal 6 has the valence 32

Signal 7 has the valence 64

Signal 8 has the valence 128

The valences of the single bit add up.

Example:

Signal 3 and 5 are active.

Signal status has the value 20 (4+16).

Access rights

MTB:	Read
Setter:	Read
User:	Read
NC prog.:	Read
PLC prog.:	Read

Activation

-

Update

A SignalStatus bit is set or deleted when reaching the specified distance to a mark.

When using the signal duration (Duration) the bit is deleted again after the time has lapsed.

Dependency

The function "Programmable distance-dependent signal" is applied and activated.

Data type

UnsignedInt_t (4 bytes)

Unit of the distance

Unit of the distance	Unit

Unit /SysAdvDistSig[]/Sig[1..8]/DistOn and DistOff.

Path

/SysAdvDistSig[<Channel>]/Unit

CPL: SD.SysAdvDistSig[<Channel>].Unit

Value range

0, 1

Pre-assignment

0

Unit

-

Description of the variable values

0: mm

1: inch

Access rights

MTB: Read and write

Setter: Read and write

User: Read and write

NC prog.: Read and write

PLC prog.: Read and write

Activation

The next programming of AdvDistSigEnable()/ADSENA() uses the set value.

Update

-

Dependency

-

Data type

UnsignedInt_t (4 bytes)

Distance to the programmed Set mark ADSLAB

Distance to the programmed Set markADSLAB	Sig[].DistOn

Distance to the programmed mark in mm or inch.

Path

/SysAdvDistSig[<Channel>]/Sig[1..8]/DistOn

CPL: SD.SysAdvDistSig[<Channel>].Sig[1..8].DistOn

Value range

>=0

Pre-assignment

0

Unit

mm or inch

System data

Description of the variable values	Distance to the mark ADSLAB(<Signal>). The effective distance (EffDistOn) is generated from this value and from the distance (TimeDistOn) calculated from Sig[].TimeOn. The signal sets EffDistOn to 1 using the programmed mark before reaching the end of the block.
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	The next programming of AdvDistSigEnable()/ADSENA() uses the set value.
Update	-
Dependency	The unit of DistOn is specified by /SysAdvDist[<Channel>]/Unit.
Data type	Float_t (4 bytes)

Distance to the programmed Reset mark ADSLAB

Distance to the programmed Reset markADSLAB		Sig[].DistOff
	Distance to the programmed mark in mm or inch.	
Path	/SysAdvDistSig[<Channel>]/Sig[1..8]/DistOff CPL: SD.SysAdvDistSig[<Channel>].Sig[1..8].DistOff	
Value range	>=0	
Pre-assignment	0	
Unit	mm or inch	
Description of the variable values	Distance to the mark ADSLAB(--<Signal>). The effective distance (EffDistOff) is generated from this value and from the distance (TimeDistOff) calculated from Sig[].TimeOff. The signal sets EffDistOff to 0 using the programmed mark before reaching the end of the block.	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	The next programming of AdvDistSigEnable()/ADSENA() uses the set value.	
Update	-	
Dependency	The unit of DistOff is specified by /SysAdvDist[<Channel>]/Unit.	
Data type	Float_t (4 bytes)	

Time interval to the programmed Set mark ADSLAB

Time interval to the programmed Set markADSLAB		Sig[].TimeOn
	Distance to programmed mark in ms.	
Path	/SysAdvDistSig[<Channel>]/Sig[1..8]/TimeOn CPL: SD.SysAdvDistSig[<Channel>].Sig[1..8].TimeOn	
Value range	>=0	

Pre-assignment	0
Unit	ms
Description of the variable values	Distance to the mark ADSLAB(<Signal>). A distance (TimeDistOn) is calculated from the time TimeOn by using the last programmed feed. The sum of this value and Sig[].DistOn returns the effective distance (EffDistOn) to the mark ADSLAB(<Signal>). The signal sets EffDistOn to 1 using the programmed mark before reaching the end of the block.
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	The next programming of AdvDistSigEnable()/ADSENA() uses the set value.
Update	-
Dependency	If the bit is set in the /SysAdvDistSigE[<Channel>]/OveForTime, the position of the feed potentiometer during block preparation is considered when converting from TimeOn and TimeOff into one distance.
Data type	Float_t (4 bytes)

Time interval to the programmed Reset mark ADSLAB

Time interval to the programmed Reset mark ADSLAB		Sig[].TimeOff
	Distance to programmed mark in ms.	
Path	/SysAdvDistSig[<Channel>]/Sig[1..8]/TimeOff CPL: SD.SysAdvDistSig[<Channel>].Sig[1..8].TimeOff	
Value range	>=0	
Pre-assignment	0	
Unit	ms	
Description of the variable values	Distance to the mark ADSLAB(--<Signal>). A distance (TimeDistOff) is calculated from the time TimeOff by using the last programmed feed. The sum of this value and Sig[].DistOff returns the effective distance (EffDistOff) to the mark ADSLAB(<Signal>). The signal sets EffDistOff to 0 using the programmed mark before reaching the end of the block.	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	The next programming of AdvDistSigEnable()/ADSENA() uses the set value.	
Update	-	
Dependency	If the bit is set in the /SysAdvDistSigE[<Channel>]/OveForTime, the position of the feed potentiometer during block preparation is considered when converting from TimeOn and TimeOff into one distance.	
Data type	Float_t (4 bytes)	

Type of distance calculation

Type of distance calculation		Sig[].DistCalc
The comparison value for /SysAdvDistSig[<Channel>]/Sig[1..8]/DistOn and DistOff is calculated using the method selected here.		
Path	/SysAdvDistSig[<Channel>]/Sig[1..8]/DistCalc CPL: SD.SysAdvDistSig[<Channel>].Sig[1..8].DistCalc	
Value range	0, 1, 2	
Pre-assignment	0	
Unit	-	
Description of the variable values	0: A distance is generated in the active plane. 1: The distance is generated in the active plane. 2: The distance is generated via all coordinates.	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	The next programming of AdvDistSigEnable()/ADSENA() uses the set value.	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Signal duration

Signal duration		Sig[].Duration
The signal is set for the specified duration.		
Path	/SysAdvDistSig[<Channel>]/Sig[1..8]/Duration CPL: SD.SysAdvDistSig[<Channel>].Sig[1..8].Duration	
Value range	0.5 - 10000.0 -1	
Pre-assignment	0	
Unit	ms	
Description of the variable values	0.5 - 10000.0: Signal duration of the high level in ms. The time is rounded to the next larger multiple of the NC cycle time. -1 : The signal level is only switched on by ADSLAB().	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	The next programming of AdvDistSigEnable()/ADSENA() uses the set value.	
Update	-	
Dependency	The unit of DistOn is specified by /SysAdvDist[<Channel>]/Unit.	

Data type Float_t (4 bytes)

Signal type selection

Signal type selection		Sig[].Type
Determines the signal type (PLC and HighSpeed).		
Path	/SysAdvDistSig[<Channel>]/Sig[1..8]/Type CPL: SD.SysAdvDistSig[<Channel>].Sig[1..8].Type	
Value range	0, 1, 10, 11	
Pre-assignment	0	
Unit	-	
Description of the variable values	0: iCh_Customx 1: iCh_Cplx 10: iCh_Customx + The HS output of the PosDepHSOut function. 11: iCh_Cplx + The HS output of the PosDepHSOut function.	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	The next programming of AdvDistSigEnable()/ADSENA() uses the set value.	
Update	-	
Dependency	The index of the PLC signal is determined by /SysAdvDistSig[<Channel>]/Sig[1..8]/Index.	
Data type	UnsignedShort_t (2 bytes)	

Index of the selected signal

Index of the selected signal		Sig[].Index
Determines the index of the PLC signal selected with /SysAdvDistSig[<Channel>]/Sig[1..8]/Type.		
Path	/SysAdvDistSig[<Channel>]/Sig[1..8]/Index CPL: SD.SysAdvDistSig[<Channel>].Sig[1..8].Index	
Value range	0 - 16	
Pre-assignment	0	
Unit	-	
Description of the variable values	0: No PLC signal is set. 1 - 8: Index for Sig[]/Type = 0 (iCh_Custom1 .. iCh_Custom8). 1 - 16: Index for Sig[]/Type = 1 (iChCpl01 .. iChCpl16).	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	The next programming of AdvDistSigEnable()/ADSENA() uses the set value.	
Update	-	

Dependency Determines the used PLC signal together with /SysAdvDistSig[<Channel>]/Sig[1..8]/Type.

Data type UnsignedShort_t (2 bytes)

32.2.4 Advanced control data of the function "Programmable distance-dependent signal"/SysAdvDistSigE"

General information

When creating the system data structure SysAdvDistSig, the extension SysAdvDistSigE is also created automatically.

It contains optional extensions for the NC function "Programmable distance-dependent signal" ADS.

The following structural elements are defined in the "sdadvdistsig.xsd_" schema:

Structural element	Description
OveForTime	Note feed potentiometer for TimeOn and TimeOff (bit-coded)
ReqLookAhead	Refer to the block look-ahead for "Programmable distance-dependent signal"

Tab. 32-3: Overview on SysAdvDistSigE structural elements



For more information, refer to the "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P) in the chapter "Programmable distance-dependent signal".

Note feed potentiometer for TimeOn and TimeOff

Note feed potentiometer for TimeOn and TimeOff	OveForTime

The position of the feed potentiometer during block preparation is considered when converting from TimeOn and TimeOff into one distance.

Path /SysAdvDistSigE[<Channel>]/OveForTime

CPL: SD.SysAdvDistSigE[<Channel>].OveForTime

Value range 0.. 255

Pre-assignment 0

Unit -

Description of the variable values If the bit is set, the position of the feed potentiometer during block preparation is considered when converting from TimeOn and TimeOff into one distance.

Signal 1 has the valence 1

Signal 2 has the valence 2

Signal 3 has the valence 4

...

Signal 8 has the valence 128

The valences of the single bit add up. If the value is 255, the feed potentiometer is considered for all signals.

Access rights **MTB:** Read and write

Setter: Read and write

	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	The next programming of AdvDistSigEnable()/ADSENA() uses the set value of the bit.	
Update	-	
Dependency	-	
Data type	UnsignedInt_t (4 bytes)	

Block look-ahead for "Programmable distance-dependent signal"

Refer to the block look-ahead for "Programmable distance-dependent signal"		ReqLookAhead
The block look-ahead range to search for programmed marks is 20 programmed NC blocks. If this number of blocks is not sufficient in some cases, the block look-ahead can be extended of up to 100 NC blocks using this parameter.		
Path	/SysAdvDistSigE[<Channel>]/ReqLookAhead CPL: SD.SysAdvDistSigE[<Channel>].ReqLookAhead	
Value range	0, 21 .. 100	
Pre-assignment	0	
Unit	-	
Description of the variable values	0:	The block look-ahead range consists of 20 programmed NC blocks.
	21 .. 100:	The block look-ahead range consists of 21 ... 100 programmed NC blocks.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	The "first" programming of AdvDistSigEnable()/ADSENA() enables the set value.	
Update	-	
Dependency	The values of the machine parameters MaxBlkMiscFunc and PercMiscFunc must be sufficiently high. A block look-ahead of 100 NC blocks requires at least 205 NC blocks for "other functions".	
Data type	UnsignedInt_t (4 bytes)	

32.2.5 Additional data of the function "programmable distance-dependent signal ""/SysAdvDistSigA"

General information

The system data structure SysAdvDistSigA can be created optionally for each channel in the volatile memory.

This data provides additional information on a FlyCut function block.

The following structural elements are defined in the "sdadvdistsig.xsd_" schema:

The index [1..8] corresponds to the index of /SysAdvDistSig[CHANNEL]/Sig[1..8]:

Structural element	Description
TimeStamp	Time stamp of the last update
Validity[1..8]	Validity of the remaining time "RemTime"
RemTime[1..8]	Remaining time up to the switching point

Tab. 32-4: Overview on SysAdvDistSigA structural elements



For more information, refer to the "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P) in the chapter "Programmable distance-dependent signal".

Time stamp of the last update

Time stamp of the last update	TimeStamp

Within a temporal context of an ADS signal, the time stamp is also updated together with each Validity[] and RemTime[] update.

Path /SysAdvDistSigA[<Channel>]/TimeStamp

CPL: SD.SysAdvDistSigA[<Channel>].TimeStamp

Value range 0.. 4294967295

Pre-assignment 0

Unit -

Description of the variable values Copy of the interrupt counter upon the last update of Validity[] and RemTime[]

Access rights **MTB:** Read

Setter: Read

User: Read

NC prog.: Read

PLC prog.: Read

Activation Refer to the "Programmable distance-dependent signal"

Update Automatically within the temporal context of an ADS signal.

Dependency The function "Programmable distance-dependent signal" is applied and activated.

Data type UnsignedInt_t (4 bytes)

Validity of the remaining time "RemTime"

Validity of the remaining time "RemTime"	Validity[]

Within a temporal context of an ADS signal (rising or falling edge of a signal), the validity of the remaining time "RemTime" is shown.

Path /SysAdvDistSigA[<Channel>]/Validity[<Signal>]

CPL: SD.SysAdvDistSigA[<Channel>].Validity[<Signal>]

Value range 0, 1, -1

Pre-assignment 0

Unit -

Description of the variable values	0:	Invalid
	1:	A rising edge is processed
	-1 :	A falling edge is processed
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	Refer to the "Programmable distance-dependent signal"	
Update	Automatically within the temporal context of an ADS signal.	
Dependency	The function "Programmable distance-dependent signal" is applied and activated.	
Data type	8 * Short_t (8 * 2 bytes)	

Remaining time up to the switching point

Remaining time up to the switching point		RemTime[]
Within a temporal context of an ADS signal, the time up to the signal switching or after the signal switching is shown.		
Path	/SysAdvDistSigA[<Channel>]/RemTime[<Signal> CPL: SD.SysAdvDistSigA[<Channel>].RemTime[<Signal>	
Value range	-2147483648 .. 2147483647	
Pre-assignment	0	
Unit	µs	
Description of the variable values	>0:	Time up to signal switching
	<=0:	Time after signal switching
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	Refer to the "Programmable distance-dependent signal"	
Update	Automatically within the temporal context of an ADS signal.	
Dependency	The function "Programmable distance-dependent signal" is applied and activated. The value is only valid for /SysAdvDistSigA/Validity[] 1 or -1.	
Data type	8 * Int_t (8 * 4 bytes)	

32.2.6 Automatic program selection after a system restart "/SysAPrg"

Part program for automatic program selection after system restart

Part program for automatic program selection after system restart	LastSelProg

Defines for each channel the path and name of the part program for automatic program selection after system restart and also provides the part program selected last for this function.

System data

This system date is stored permanently.

If it contains the entry "NoAutoProgSel", no program is selected after control startup. Any other entry starts a search for a program with this name and its selection. In each subsequent program selection, "LastSelProg" is updated with path and name of the selected program. The program selected last is selected again after the next control startup as soon as a program selection is possible again.

Path	/SysAPrg[<Channel>]/LastSelProg CPL: SD.SysAPrg[<Channel>].LastSelProg or SD.SysAPrg.LastSelProg in the individual channel	
Value range	Characters from IsoLatin1String	
Pre-assignment	"NoAutoProgSel"	
Unit	-	
Description of the variable values	NoAutoProgSel	No automatic program selection after system restart and no entry updating.
	<invalid program name>	Automatic program selection after system restart is enabled and it selects the specified program. The entry is updated the next time the program is selected.
	<invalid program name>	At first, automatic program selection fails after system restart (runtime warning). However, the entry is updated each time the program is selected, so that automatic program selection can be executed successfully after the next system restart.
		An invalid program name can be used as initial state. At runtime, such a state occurs only if the program selected last could not be found.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	By entering an initial value not equal to "NoAutoProgSel"	
Update	At every program selection (if the entry is not equal to "NoAutoProgSel")	
Dependency	<i>The automatic program selection is only active</i> - if the automatic program reselection is applied in the machine parameters ("AutoProgSel" in the path "CHAN/Ch[i]/Ini") and - if the interface signal "qCh_ReSelOff" (automatic reselection off) is in the state "0".	
Data type	IsoLatin1String (100 bytes for text + 3 bytes + 1 byte for end identification)	

Program selection flags for automatic program selection after system restart

Program selection flags for automatic program selection after system restart	ProgSelFlags

Saves the program selection flags for automatic program selection channel-specifically after a system restart.

This system date is stored permanently. The program selection flags correspond to those of the CPL and NCS. This system variable is only used to save the setting that was active last and should be changed only using the function itself.

Path	/SysAPrg[<Channel>]/ProgSelFlags CPL: SD.SysAPrg[<Channel>].ProgSelFlags - or - SD.SysAPrg.ProgSelFlags in the individual channel		
Value range	-		
Pre-assignment	0		
Unit	-		
Description of the variable values	The program selection flags correspond to those of the CPL and NCS.		
Access rights	MTB:	Read	
	Setter:	Read	
	User:	Read	
	NC prog.:	Read	
	PLC prog.:	Read	
Activation	Implicitly by the automatic program selection function.		
Update	By the automatic program selection function.		
Dependency	See system date "LastSelProg"		
Data type	UnsignedInt_t (4 bytes)		

Fixing part program for automatic program selection after a system restart

Fixing part program for automatic program selection after system restart		fixDefault
Allows to permanently specify the part program for automatic program selection channel-specifically after system restart.		
This system date is stored permanently. After setting to "1", the part program entered in the system date "LastSelProg" is no longer set at program selection.		
Path	/SysAPrg[<Channel>]/fixDefault CPL: SD.SysAPrg[<Channel>].fixDefault - or - SD.SysAPrg.fixDefault in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	The part program entered in system date "LastSelProg" is updated at each program selection.
	1	The part program entered in system date "LastSelProg" remains unchanged. It is no longer updated at program selection.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write

Activation	Implicitly by the automatic program selection function
Update	-
Dependency	See system data "LastSelProg"
Data type	Int_t (4 bytes)

32.2.7 Actual data of the system axes "/SysAxActData"

General information

The system data structure SysAxActData is created for each NC-controlled drive in the volatile memory.

It contains specified actual data of the system axes such as names of the active correction data files for the lead screw error compensation (LSEC) or for the cross compensation (CCOMP).

The following structural elements are defined in the "sdaxis.xsd" schema:

Structural element	Description
LSEC_FileName	Active LSEC table
CCOMP_FileName1	Active CCOMP table 1
CCOMP_FileName2	Active CCOMP table 2

Tab. 32-5: Overview on structural elements "SysAxActData"

Active LSEC table

Active LSEC file	LSEC_FileName

Returns the file name of the active LSEC table (without path specification) with regard to the lead screw error compensation.

Path /SysAxActData[<Drive>]/LSEC_FileName

CPL: SD.SysAxActData[<Drive>].LSEC_FileName

Value range -

Pre-assignment <EmptyString>

Unit -

Description of the variable values

<Empty-String>	No LSEC table active
<otherwise>	Name of the active LSEC table

Access rights

MTB:	Read
Setter:	Read
User:	Read
NC prog.:	Read
PLC prog.:	Read

Activation -

Update By the NC kernel during each change of an LSEC table

Dependency -

Data type isoStr31_t (30 bytes for text + 1 byte + 1 byte for end identification)

Active CCOMP table 1 and 2

Active CCOMP table 1 and 2	CCOMP_FileName1 CCOMP_FileName2

Provides the file name of the active CCOMP table 1 and the active CCOMP table 2 (without path specification) with regard to the cross compensation.	
Path	/SysAxActData[<Drive>]/CCOMP_FileName1 /SysAxActData[<Drive>]/CCOMP_FileName2 CPL: SD.SysAxActData[<Drive>].CCOMP_FileName1 CPL: SD.SysAxActData[<Drive>].CCOMP_FileName2
Value range	-
Pre-assignment	<EmptyString>
Unit	-
Description of the variable values	<Empty-String> No CCOMP table 1 (or CCOMP table 2) active <otherwise> Name of the active CCOMP table 1 (or CCOMP table 2)
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	By the NC kernel during each change of an CCOMP table 1 (or CCOMP table 2)
Dependency	-
Data type	isoStr31_t (30 bytes for text + 1 byte + 1 byte for end identification)

32.2.8 System axis coupling command"/SysAxCoupleCmd"

Switching system axis coupling on/off

Switch system axis coupling on/off		OnOff
Switches system axis coupling on/off for the respective system axis. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/OnOff	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0 Switch system axis coupling off 1 Switch system axis coupling on	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	

System data

Data type Byte_t (1 byte)

Resetting the system axis coupling

Reset the system axis coupling		Reset
Allows resetting the positions of the system axis coupling for the respective system axis. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Reset	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No resetting
	1	Reset error handling of the system axis coupling.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Byte_t (1 byte)	

System axis coupling with additional interpolation

System axis coupling with additional interpolation		IpoEnable
Allows the system axis coupling to activate additional interpolation for the respective system axis. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/IpoEnable	
Value range	-	
Pre-assignment	false	
Unit	-	
Description of the variable values	false	No additional interpolation
	true	The interpolated axis value is added to the master value.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	

Data type Boolean_t (1 byte)

System axis coupling with recoupling in case of asynchronism

System axis coupling with recoupling in case of asynchronism		RecoupleEnable
Specifies whether recoupling is to be permitted for the respective system axis when synchronism no longer exists during system axis coupling.		
This system data is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/RecoupleEnable	
Value range	-	
Pre-assignment	false	
Unit	-	
Description of the variable values	false	Lacking synchronism leads to error.
	true	Recoupling is permitted for asynchronism.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Boolean_t (1 byte)	

Maximum acceleration when coupling and decoupling in case of system axis coupling

Maximum acceleration when coupling and decoupling in case of system axis coupling		AccCouple
Specifies the maximum acceleration when coupling and decoupling for the respective system axis during system axis coupling.		
This system data is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/AccCouple	
Value range	-	
Pre-assignment	0	
Unit	m/s ²	
Description of the variable values	Maximum acceleration when coupling and decoupling	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	

System data

Data type Double_t (4 bytes)

Maximum velocity when coupling and decoupling in case of system axis coupling

Maximum velocity when coupling and decoupling in case of system axis coupling	VelCouple

Specifies the maximum velocity when coupling and decoupling for the respective system axis during system axis coupling.

This system date is stored permanently.

Path /SysAxCoupleCmd[<Drive>]/VelCouple

Value range -

Pre-assignment 0

Unit mm/min or °/min

Description of the variable values Maximum velocity when coupling and decoupling.

Access rights MTB: Read and write

Setter: Read and write

User: Read and write

NC prog.: Read and write

PLC prog.: Read and write

Activation After applying the system axis coupling data with system date "Validate".

Update -

Dependency -

Data type Double_t (4 bytes)

Tooth pitch of the slave axis

Tooth pitch of the slave axis	GearTeeth

Specifies the tooth pitch for the respective system axis. This results in shorter coupling distances and thus shorter coupling periods. The coupling is carried out on the next tooth.

This system date is stored permanently.

Path /SysAxCoupleCmd[<Drive>]/GearTeeth

Value range -

Pre-assignment 0

Unit -

Description of the variable values	0 - No tooth pitch	When coupling/recoupling, the absolute master value position is used for synchronization.
	>1 - Number of slave axis teeth	When coupling/recoupling modulo axes, a relative master value position to the next tooth of the slave axis is specified.

Access rights MTB: Read and write

Setter: Read and write

User: Read and write

NC prog.: Read and write

PLC prog.: Read and write

Activation	After applying the system axis coupling data with system date "Validate".
Update	-
Dependency	-
Data type	Double_t (8 bytes)

Synchronization mode for slave axis

Synchronization mode for slave axis		SyncMode
Synchronization mode to couple the slave axis to its master value. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/SyncMode	
Value range	0-4	
Pre-assignment	0	
Unit	-	
Description of the variable values	0 - shortest distance	For modulo axes, the slave axis couples to the synchronous position with the least distance (preceding or successive). Non-modulo axes always have to be position-synchronized with this mode.
	1 - Time-optimum distance	For the position adjustment of modulo axes, the slave axis selects the synchronous position (preceding or successive) that can be reached first considering the velocity limitation.
	2 - Velocity coupling	Only one velocity adjustment to the master value is carried out. The position difference of the slave axis to the master value is saved in the status date /SysAxCoupleSta[<Drive>]/ActOffset after the velocity synchronism has been reached. Non-modulo axes can also be coupled to this mode.
	3 - Positive distance	For modulo axes, the slave axis selects the synchronous position for position adjustment. The synchronous position is in positive velocity direction.
	4 - Negative distance	For modulo axes, the slave axis selects the synchronous position for position adjustment. The synchronous position is in negative velocity direction.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	

System data

Data type Int_t (4 bytes)

Coupling rigidity of the system axis coupling after successful coupling

Coupling rigidity of the system axis coupling after successful coupling		Rigidity
Specifies the coupling rigidity of system axis coupling for the respective system axis after coupling. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Rigidity	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0 - Soft coupling	Coupling/recoupling at most with the acceleration/velocity values entered in system data "AccCouple" or "VelCouple".
	1 - Hard coupling	Coupling/recoupling at most with the acceleration/velocity values entered in the machine parameters "FeedAxAcc" and "RapidAxAcc" or "MaxVel".
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Byte_t (1 byte)	

Selecting the permitted synchronization deviation of the system axis coupling

Selecting permitted synchronization deviation of system axis coupling		PosWinSel
Specifies whether synchronization deviation is monitored and with which synchronization window for the respective system axis during system axis coupling. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/PosWinSel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No synchronous run monitoring
	1	Monitoring synchronization deviation in synchronization window 1 (e.g. for fine deviation)
	2	Monitoring synchronization deviation in synchronization window 2 (e.g. for fine deviation)

	3	Monitoring the synchronization deviation in the two synchronization windows 1 and 2 (with hysteresis response)
	11	Monitoring the command value deviation in the monitoring window 1 (e.g. for rough deviation)
	12	Monitoring the command value deviation in the monitoring window 1 (e.g. for fine deviation)
	13	Monitoring the command value deviation in the two monitoring windows 1 and 2 (with hysteresis response)
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Byte_t (1 byte)	

Definition of possible synchronization deviations of the system axis coupling

Definition of possible synchronization deviations of the system axis coupling		PosWin[1..2]
Defines the possible values of the synchronization deviations and command value deviations of the system axis coupling for the affected system axis. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/PosWin[1..2]	
Value range	-	
Pre-assignment	0	
Unit	mm	
Description of the variable values	PosWin[1]	Position deviation for synchronization window 1 (z. B. deviation rough)
	PosWin[2]	Position deviation for synchronization window 2 (z. B. deviation fine)
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	2 x Double_t (2 x 8 bytes)	

Triggering coupling errors in the system axis coupling

Trigger coupling error in system axis coupling		ExtError
This system date is stored permanently.		
Permits a coupling error to be triggered for the respective system axis, thus, aborting coupling.		
Path	/SysAxCoupleCmd[<Drive>]/ExtError	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No termination
	Otherwise	Cancellation
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Short_t (2 bytes)	

Reaction-triggering error sources of the system axis coupling

Reaction-triggering error sources of the system axis coupling		AbortEnable
Specifies which error sources of the system axis coupling should trigger error handling for the respective system axis.		
This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/AbortEnable	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No error handling
	Bit 2 ⁰ (1)	Reaction in case of drive error
	Bit 2 ¹ (2)	Reaction in case of coupling error
	Bit 2 ² (4)	Reaction in case of external error (ExtError)
	Bit 2 ³ (8)	Reaction in case of error in interpolation channel
	Bit 2 ⁴ (16)	Reaction in case of error in a slave axis
	Bit 2 ⁵ (32)	Reaction in case of error of the master axis 1
	Bit 2 ⁶ (64)	Reaction in case of error of the master axis 2
	Bit 2 ⁷ (128)	Reaction in case of error of the master axis 3
	Bit 2 ⁸ (256)	Reaction in case of error of the master axis 4
	Bit 2 ⁹ (512)	Reaction in case of error of the master axis 5

	Bit 2¹⁶ (65536)	Response in case of formula error zero division
	Bit 2¹⁷ (131072)	Response in case of formula error overflow
	Bit 2¹⁸ (262144)	Response in case of formula error definition range violation
	Bit 2¹⁹ (524288)	Response in case of formula error underflow
	Bit 2²⁰ (1048576)	Response in case of formula error unknown operation
	Bit 2²⁴ (16777216)	Reaction when exiting the determined accuracy window of the synchronous run monitoring
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Delay before error handling of the system axis coupling

Delay before error handling of the system axis coupling		AbortDelay
Specifies the waiting time for the respective system axis until error handling is triggered.		
This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/AbortDelay	
Value range	-	
Pre-assignment	0	
Unit	ms	
Description of the variable values	Delay time	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Asynchronous subroutine for system axis coupling errors

Asynchronous subroutine for system axis coupling errors		AbortAsupNo
Specifies which asynchronous subroutine is to be triggered in case of an error for the respective system axis of the system axis coupling.		
This system date is stored permanently. If the axis is defined as an asynchronous axis, no asynchronous subroutine is started.		
Path	/SysAxCoupleCmd[<Drive>]/AbortAsupNo	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No asynchronous subroutine is started.
	1.. 8	Number of the asynchronous subroutine that starts (effective only for asynchronous axis).
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Byte_t (1 byte)	

Channel for asynchronous subroutine for system axis coupling errors

Channel for asynchronous subroutine for system axis coupling errors		AbortAsupChan
Specifies in which channel an asynchronous subroutine is to be called in case of an error for the affected system axis of system axis coupling.		
This system date is stored permanently. If the axis is defined as an asynchronous axis, no asynchronous subroutine is started.		
Path	/SysAxCoupleCmd[<Drive>]/AbortAsupChan	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Call in the current interpolation channel of the axis
	1.. 60	Channel index
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	

Update -
 Dependency -
 Data type Byte_t (1 byte)

Error reactions of the system axis coupling

Error reactions of the system axis coupling		AbortAction
Specifies which error reactions can be triggered for the respective system axis during system axis coupling.		
Path	This system data is stored permanently. /SysAxCoupleCmd[<Drive>]/AbortAction	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	System axis coupling continues to run
	- or -	
	Bit 2 ⁰ (1)	Retraction absolute after AbortRetractPos
	- or -	
	Bit 2 ¹ (2)	Retraction relative by AbortRetractPos
	- or -	
	Bit 2 ² (4)	Send channel error
	- or -	
	a combination of the bits 3-8:	
	Bit 2 ³ (8)	Disable interpolation value
	Bit 2 ⁴ (16)	Switch off master value 1
	Bit 2 ⁵ (32)	Switch off master value 2
	Bit 2 ⁶ (64)	Switch off master value 3
	Bit 2 ⁷ (128)	Switch off master value 4
	Bit 2 ⁸ (256)	Switch off master value 5
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system data "Validate".	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Retract position for error handling of system axis coupling

Retract position for error handling of system axis coupling	AbortRetractPos
Defines a retract position that is approached in certain error reactions for the respective system axis.	

System data

	This system date is stored permanently.	
Path	/SysAxCoupleCmd[<Drive>]/AbortRetractPos	
Value range	-	
Pre-assignment	0	
Unit	mm or °	
Description of the variable values	Retract position	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Double_t (8 bytes)	

Formula for the master value calculation in the system axis coupling

Formula for the master value calculation in the system axis coupling		Formula
Defines a mathematic formula to calculate a complete master value from the individual master values for the respective system axis.		
The formula string allows the operands AC, L1, ..., L5, V1, ..., V5, SPOS, PI, TI, interpolation-synchronous permanent CPL variables, interpolation-synchronous system data as well as the constants.		
These can be linked to the fundamental arithmetics "+", "-", "*", "/", "MOD".		
For expressions, the following mathematical functions can be used: ABS, INT, SGN, ACC, FEED, INCH, SPEED, ACOS, ASIN, ATAN, COS, SIN, TAN, ACOSH, ASINH, ATANH, COSH, SINH, TANH, ATAN2, DIST, CBRT, CUB, EXP, LN, SQR, SQRT, MIN and MAX.		
The relational operators "<", "<=", "=", ">", ">=", ">" and the logical operators AND, OR and NOT, the following conditioned expressions can be shown:		
<Condition>?<Expression condition met>:<Expression condition not met>		
are implemented.		
This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Formula	
Value range	-	
Pre-assignment	""	
Unit	-	
Description of the variable values	""	
	No secondary formula processing active. The total master value of the slave axis results from the sum of the individual master values and the interpolation value of the slave axis.	
	" <Formula>"	
	The total master axis of the slave axis results due to the analysis of the formula.	
Access rights	MTB:	Read and write
	Setter:	Read and write

User:	Read and write
NC prog.:	Read and write
PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".
Update	-
Dependency	-
Data type	String_t (250 bytes)

Enabling the master axis for system axis coupling

Enable master axis for system axis coupling		Enable
Specifies whether it participates in system axis coupling as the master axis with the specified master axis index (1...5) for the respective system axis. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Master[1..5]/Enable	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	false	System axis with the specified master axis index is not enabled as the master axis.
	true	System axis with the specified master axis index participates as the master axis in system axis coupling.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Boolean_t (1 byte)	

System axis index of the master axis of the system axis coupling

System axis index of the master axis of the system axis coupling		AxIndex
Specifies which axis should be the master axis in system axis coupling for the respective system axis. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Master[1..5]/AxIndex	
Value range	0..250	
Pre-assignment	0	
Unit	-	
Description of the variable values	System axis index of master axis	

System data

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate" and met real-time condition "SwitchCond".	
Update	-	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Real-time conditioned master axis switching of a system axis coupling

Real-time conditioned master axis switching in the system axis coupling		SwitchCond
Specifies the real-time conditioned switching behavior of a master axis of the system axis coupling for the respective system axis. To activate the master axis coupling rule consisting of AxIndex, N_Input, N_Output, Offset, Shift and Tab is delayed until the real-time condition is received.		
This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Master[1..5]/SwitchCond	
Value range	0, 1, 2, 100, ..., 131	
Pre-assignment	0	
Unit	-	
Description of the variable values	ValueCondition	Activating the coupling rule according to "Validate"
	0 Unconditioned	Immediately
	1 Position-conditioned	As soon as SwitchMin <= Position(AxIndex) <= SwitchMax
	2 Velocity-conditioned	As soon as SwitchMin <= Velocity(AxIndex) <= SwitchMax
	100 Externally triggered	As soon as SysAxCoupleCtr.ExtCond Bit 2 ⁰ is active.
	... 131 Externally triggered	As soon as SysAxCoupleCtr.ExtCond Bit 2 ³¹ is active.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	
Update	-	
Dependency	-	
Data type	Short_t (2 bytes)	

Lower switching position or switching velocity

Lower switching position or switching velocity in the system axis coupling	SwitchMin

Defines the lower switching position or switching velocity for the real-time conditioned master axis switching of the respective system slave axis.

This system date is stored permanently.

Path	/SysAxCoupleCmd[<Drive>]/Master[1..5]/SwitchMin		
Value range	-		
Pre-assignment	0.0		
Unit	mm or ° or mm/min or °/min		
Description of the variable values	Lower switching position (SwitchCond=1) or lower switching velocity (SwitchCond=2)		
Access rights	MTB:	Read and write	
	Setter:	Read and write	
	User:	Read and write	
	NC prog.:	Read and write	
	PLC prog.:	Read and write	
Activation	After applying the system axis coupling data with system date "Validate".		
Update	-		
Dependency	SwitchCond specifies the meanings of "switching position" or "switching velocity".		
Data type	Double_t (8 bytes)		

Upper switching position or switching velocity

Upper switching position or switching velocity in the system axis coupling		SwitchMax
Defines the upper switching position or switching velocity for the real-time conditioned master axis switching of the respective system slave axis. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Master[1..5]/SwitchMax	
Value range	-	
Pre-assignment	0.0	
Unit	mm or ° or mm/min or °/min	
Description of the variable values	Upper switching position (SwitchCond=1) or upper switching velocity (SwitchCond=2)	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	After applying the system axis coupling data with system date "Validate".	

System data

Update	-
Dependency	SwitchCond specifies the meanings of "switching position" or "switching velocity".
Data type	Double_t (8 bytes)

Master value calculation mode in the system axis coupling

Master value calculation mode in the system axis coupling		Mode
Specifies the master value calculation mode for the respective system axis. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Master[1..5]/Mode	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Calculation mode:	
	0	Master value is summed up at a linear slave axis.
	1	Modulo calculation of the master value if the master axis "Ax-Index" is an endless axis.
	Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
	Activation	After applying the system axis coupling data with system date "Validate".
Update	-	
Dependency	-	
Data type	Byte_t (1 byte)	

Offset of slave axis to master axis of the system axis coupling

Offset of slave axis to master axis of the system axis coupling		Offset
Specifies the positional offset of the slave axis compared to the master axis of the system axis coupling for the respective system axis. This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Master[1..5]/Offset	
Value range	-	
Pre-assignment	0	
Unit	mm or °	
Description of the variable values	Positional offset of slave axis to master axis	
	Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
	Activation	After applying the system axis coupling data with system date

"Validate" and fulfilled real-time condition "SwitchCond".

Update -
Dependency -
Data type Double_t (8 bytes)

Input revolutions of the master axis of the system axis coupling

Input revolutions of the master axis of the system axis coupling	N_Input

Specifies the number of input revolutions of the master axis of the system axis coupling for the respective system axis.

This system data is stored permanently.

Path /SysAxCoupleCmd[<Drive>]/Master[1..5]/N_Input
Value range -
Pre-assignment 0
Unit -
Description of the variable values Number of input revolutions of master axis
Access rights
MTB: Read and write
Setter: Read and write
User: Read and write
NC prog.: Read and write
PLC prog.: Read and write
Activation After applying the system axis coupling data with system date
 "Validate" and fulfilled real-time condition "SwitchCond".
Update -
Dependency -
Data type Double_t (8 bytes)

Output revolutions of slave axis in system axis coupling

Output revolutions of the slave axis of the system axis coupling	N_Output

Specifies the number of output revolutions of the slave axis of the system axis coupling for the respective system axis.

This system data is stored permanently.

Path /SysAxCoupleCmd[<Drive>]/Master[1..5]/N_Output
Value range -
Pre-assignment 0
Unit -
Description of the variable values Number of output revolutions of slave axis
Access rights
MTB: Read and write
Setter: Read and write
User: Read and write
NC prog.: Read and write
PLC prog.: Read and write
Activation After applying the system axis coupling data with system date
 "Validate" and fulfilled real-time condition "SwitchCond".
Update -

System data

Dependency -
Data type Double_t (8 bytes)

Master axis offset of the system axis coupling

Master axis offset of the system axis coupling		Offset
Specifies an offset for the respective master axis. The offset affects the table accesses in particular.		
This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Master[1..5]/Shift	
Value range	-	
Pre-assignment	0	
Unit	mm or °	
Description of the variable values	Number of output revolutions of slave axis	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	After applying the system axis coupling data with system date "Validate" and fulfilled real-time condition "SwitchCond".	
Update	-	
Dependency	-	
Data type	Double_t (8 bytes)	

Table for the system axis coupling

Table for the system axis coupling		Tab
Specifies a coupling table for the respective system slave axis and system master axis. This table has to be located in the system table memory "SysAx-CoupleTab".		
This system date is stored permanently.		
Path	/SysAxCoupleCmd[<Drive>]/Master[1..5]/Tab	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	<Empty-String> No table coupling activated. "<name>" Enable table coupling with table <name>. "<number>" Enable table coupling with table on <number>.	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	After applying the system axis coupling data with system date	

"Validate" and fulfilled real-time condition "SwitchCond".

Update -

Dependency -

Data type String_t (30 bytes for text + 1 byte + 1 byte for end identification)

32.2.9 System axis coupling control"/SysAxCoupleCtr"

Applying the system axis coupling

Apply system axis coupling		Validate
Starts the application of system axis coupling and reports that the job has been executed.		
Path	/SysAxCoupleCtr/Validate	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Job has been executed.
	1	Job for applying data in queue.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Immediately	
Update	After the job for data transfer has been executed.	
Dependency	-	
Data type	Byte_t (1 byte)	

Error when applying the system axis coupling data

Error when applying system axis coupling data		CfgErrNum
It provides the error state when applying system axis coupling data.		
Path	/SysAxCoupleCtr/CfgErrNum	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No error
	1	Circulation
	2	Nesting too deep
	3	Axis cannot become slave axis
	4	Master enablement for invalid master axis index
	5	Master enabled without input master revolution
	6	Master enabled without master output revolution
	7	Monitoring window index invalid
	8	ASUP number invalid
	9	ASUP channel does not exist

System data

	10	AbortAction contains invalid combination
	11	SyncMode (for axis type) not allowed
	12	Invalid coupling velocity
	13	Invalid coupling acceleration
	14	Invalid input speed
	15	Invalid output speed
	16	Invalid offset
	17	Invalid retract position
	18	Invalid InPos window 1
	19	Invalid InPos window 2
	20	Table is not loaded
	21	Invalid switching condition
	22	Invalid lower switching limit
	23	Invalid upper switching limit
	24	Switching range too small
	25	Invalid modulo range: Table to slave axis
	26	Invalid modulo value: Master to table
	27	Invalid master axis shift
	28	Invalid master value calculation mode
	29	Syntax error in the formula
	30	Formula timeout
	31	Formula interpretation error
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel after the job for data transfer has been executed.	
Dependency	-	
Data type	Byte_t (1 byte)	

Incorrect axis when applying the system axis coupling data

Incorrect axis when applying the system axis coupling data		CfgErrAx
It provides the index of the first faulty axis in the application of the system axis coupling if an application error has occurred.		
Path	/SysAxCoupleCtr/CfgErrAx	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No incorrect axis
	1.. 250	Index of the first faulty axis
Access rights	MTB:	Read

Setter:	Read
User:	Read
NC prog.:	Read
PLC prog.:	Read
Activation	-
Update	By the NC kernel after the job for data transfer has been executed.
Dependency	-
Data type	UnsignedByte_t (1 Byte)

Incorrect master axis data when applying system axis coupling data

Incorrect master axis data when applying the system axis coupling data		CfgErrMaster
It provides the index of the first incorrect master axis data structure in the application of the system axis coupling if an application error has occurred.		
Path	/SysAxCoupleCtr/CfgErrMaster	
Value range	-	
Pre-assignment	-	
Unit	-	
Description of the variable values	0	There is no master axis-related error
	1.. 5	Index of the first incorrect master axis data structure "Master[m]" and for formula transfer (29 - syntax error or 31 - interpretation error in "CfgErrNum"):
	1 .. 250	Error position in the formula string "Formula"
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel after the job for data transfer has been executed.	
Dependency	-	
Data type	Byte_t (1 byte)	

External real-time switching conditions for the master axis switching of the system axis coupling

External real-time switching conditions for the master axis switching of the system axis coupling		ExtCond
Triggers the master axis switching for the switching conditions SwitchCond 100..131 respectively assigned bit by bit.		
Path	/SysAxCoupleCtr/ExtCond	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No external switching condition fulfilled

System data

	Bit 2⁰	Enables all waiting master axis conditions with Switch-Cond=100
	Bit 2¹	Enables all waiting master axis conditions with Switch-Cond=101
	...	
	Bit 2³¹	Enables all waiting master axis conditions with Switch-Cond=131
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Becomes active in the interpolator Interrupt.	
Update	-	
Dependency	-	
Data type	UnsignedInt_t (4 bytes)	

32.2.10 System axis coupling status"/SysAxCoupleSta"

Current system axis coupling data

Current system axis coupling data		all elements of SysAxCoupleCmd
These data supply the current transfer status of the system axis coupling command for the relevant system axis. The data structure below node "Cmd" is the same as that of system data "SysAxCoupleCmd" below the path /SysAxCoupleCmd[<Drive>]/... (Chapter 32.2.8 System axis coupling command "/SysAxCoupleCmd", page 435). Data is updated asynchronously in the interpolator interrupt. Accesses outside of the Interrupt are repeated if they were interrupted by an update in "SysAxCoupleSta" in the interpolator Interrupt.		
Path	/SysAxCoupleSta[<Drive>]/Cmd/...	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt.	

Axis coupling state

Axis coupling state	State
Provides the coupling state of the system axis coupling for the respective system axis. The system data is updated asynchronously in the interpolator interrupt. Accesses outside of the Interrupt are repeated if they were interrupted by an update in "SysAxCoupleSta" in the interpolator Interrupt.	

Path	/SysAxCoupleSta[<Drive>]/State	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Off
	1	Coupling (Hunting)
	2	Recoupling
	3	Synchronous (Coupled)
	4	Stopping
	5	Stopped
	6	Decoupling (Decoupling)
	7	Error
	1x	Abort
	2x	Real-time switching active (pending)
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt	
Dependency	-	
Data type	Byte_t (1 byte)	

Additional information on the axis coupling state

Additional information on the axis coupling state		StateAddOn
Provides additional information on the coupling state of the system axis coupling for the respective system axis.		
The system date is updated asynchronously in the interpolator interrupt. Accesses outside of the Interrupt are repeated if they were interrupted by an update in "SysAxCoupleSta" in the interpolator Interrupt.		
Path	/SysAxCoupleSta[<Drive>]/StateAddOn	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No additional information.
	1000	The actual position or the command value of the slave axis is in the synchronization window 1.
	2000	The actual position or the command value of the slave axis is in the synchronization window 2.
	3000	The actual position or the command value of the slave axis is in the synchronization windows 1 and 2.
	The values 1000, 2000 or 3000 can only be provided at an active synchronous run monitoring.	

System data

The following conditions have to be met:

- The system axis has to take part in the system axis coupling.
- The system axis has to be in one of the following states: "Coupling", "Recoupling" or "Synchronous".
- The respective synchronization windows have to be enabled for the system axis.

Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt	
Dependency	-	
Data type	Int_t (4 bytes)	

Abort state

Abort state	Abort

Provides the abort state of the system axis coupling for the respective system axis.

The system date is updated asynchronously in the interpolator interrupt. Accesses outside of the Interrupt are repeated if they were interrupted by an update in "SysAxCoupleSta" in the interpolator Interrupt.

Path	/SysAxCoupleSta[<Drive>]/Abort	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Inactive
	1	Abort delay
	2	Retraction active
	3	(Modified) coupling active
	4	Abort terminated, error state
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt.	
Dependency	-	
Data type	Byte_t (1 byte)	

Formula runtime error in the system axis coupling

Formula runtime error in the system axis coupling	FormulaDiag

Returns a detailed error code for a formula calculation error (Error=7) of the formula "Formula" for the respective error axis.

The system date is updated asynchronously with the interpolator interrupt.

Path	/SysAxCoupleSta[<Drive>]/FormulaDiag	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Detailed error code of the formula calculation	
	0	No error
	1	Zero division
	2	Overflow error
	3	Definition range violation
	4	Underflow error
	5-9	Formula interpretation error
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt.	
Dependency	-	
Data type	Byte_t (1 byte)	

Formula runtime error position in the system axis coupling

Formula runtime error position in the system axis coupling		FormulaDiagPos
Provides the position in the formula string "Formula" where the calculation error occurred from "FormulaDiag" for the respective axis.		
Path	/SysAxCoupleSta[<Drive>]/FormulaDiagPos	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Detailed error code of the formula calculation	
	1 .. 250	Error position in the formula string "Formula"
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt.	
Dependency	-	
Data type	Byte_t (1 byte)	

Formula expression diagnostics in the system axis coupling

Formula expression diagnostics in the system axis coupling		FormulaWatch
Returns the calculation result of the formula function WATCH for the respective program axis if this formula function was programmed in the formula string "Formula".		
Path	/SysAxCoupleSta[<Drive>]/FormulaWatch	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Calculation result of the formula expression "...WATCH(<Expression>)..."	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt.	
Dependency	-	
Data type	Double_t (8 bytes)	

Error code

Error code		Error
Provides the error state of the system axis coupling for the respective system axis.		
The system date is updated asynchronously in the interpolator interrupt. Accesses outside of the Interrupt are repeated if they were interrupted by an update in "SysAxCoupleSta" in the interpolator Interrupt.		
Path	/SysAxCoupleSta[<Drive>]/Error	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No error
	1	Coupling torn off
	2	Axis error
	3	Channel error of the axis owner
	4	Error due to external event
	5	Error in one of the slave axes
	6	Error in one of the master axes
	7	Error in the formula calculation
	8	Error at synchronous run monitoring of a master axis: Leaving the determined accuracy window
Access rights	MTB:	Read
	Setter:	Read
	User:	Read

	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt.	
Dependency	-	
Data type	Int_t (4 bytes)	

Synchronous run monitoring

Synchronous run monitoring		InPos
Provides the state of the synchronous run monitoring for the respective system axis.		
<i>The following conditions have to be met:</i>		
• The system axis has to take part in the system axis coupling. • The system axis has to be in one of the following states: "Coupling", "Recoupling" or "Synchronous". • At least one synchronization window has to be enabled for the system axis.		
If one condition is not met, "0" is provided as state.		
The system date is updated asynchronously in the interpolator interrupt. Accesses outside of the Interrupt are repeated if they were interrupted by an update in "SysAxCoupleSta" in the interpolator Interrupt.		
Path	/SysAxCoupleSta[<Drive>]/InPos	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Actual value outside of the accuracy window
	1	Actual value within the accuracy window
If both synchronization windows are selected, the "InPos" status signal shows a hysteresis behavior. InPos is set to "1" if the difference between the actual master value and the actual slave axis position is within both synchronization windows. InPos is set to "0" if the difference between the actual master value and the actual slave axis position is not within both synchronization windows.		
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt.	
Dependency	/SysAxCoupleSta[<Drive>]/State = 1..3	
	/SysAxCoupleCmd[<Drive>]/PosWinSel > 0	
Data type	Int_t (4 bytes)	

Current actual position value difference

Current actual position value difference	Lag

System data

Provides the current actual position value difference to the actual master value for the respective system axis.

The system date is updated asynchronously with the interpolator interrupt.

Path	/SysAxCoupleSta[<Drive>]/Lag
Value range	-
Pre-assignment	0
Unit	-
Description of the variable values	Current axis position
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	By the NC kernel asynchronously in the interpolator Interrupt
Dependency	-
Data type	Double_t (8 bytes)

Abort position

Abort position	AbortPos

Provides the position at which system axis coupling was interrupted for the respective system axis.

The system date is updated asynchronously in the interpolator interrupt. Accesses outside of the Interrupt are repeated if they were interrupted by an update in "SysAxCoupleSta" in the interpolator Interrupt.

Path	/SysAxCoupleSta[<Drive>]/AbortPos
Value range	-
Pre-assignment	0
Unit	-
Description of the variable values	Position at which system axis coupling was interrupted
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	By the NC kernel asynchronously in the interpolator Interrupt
Dependency	-
Data type	Double_t (8 bytes)

Current axis position

Current axis position	ActPos

Provides the current axis position for the respective system axis.

The system date is updated asynchronously in the interpolator interrupt. Accesses outside of the Interrupt are repeated if they were interrupted by an update in "SysAxCoupleSta" in the interpolator Interrupt.

Path	/SysAxCoupleSta[<Drive>]/ActPos	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Current axis position	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt	
Dependency	-	
Data type	Double_t (8 bytes)	

Current axis velocity

Current axis velocity	ActVel

Provides the current axis velocity for the respective system axis.

The system date is updated asynchronously in the interpolator interrupt. Accesses outside of the Interrupt are repeated if they were interrupted by an update in "SysAxCoupleSta" in the interpolator Interrupt.

Path	/SysAxCoupleSta[<Drive>]/ActVel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Current axis velocity	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt	
Dependency	-	
Data type	Double_t (8 bytes)	

Current master value offset

Current master value offset	ActOffset

Returns the current offset to the master value for the respective system axis in case of existing synchronism.

The offset results either from a velocity coupling (SyncMode=2) or from an exact tooth coupling (GearTeeth>1). This offset allows the repetition of an ex-

System data

act tooth coupling after a velocity coupling. The system date is updated asynchronously with the interpolator interrupt.

Path	/SysAxCoupleSta[<Drive>]/ActOffset	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Current axis position	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt.	
Dependency	-	
Data type	Double_t (8 bytes)	

Active system axis index of the master axis of the system axis coupling

Active system axis index of the master axis in system axis coupling	AxIndex

Returns the active system axis index of the master axis at system axis coupling.

This system date is updated with the master axis switching and especially with real-time-conditioned master axis switching.

Path	/SysAxCoupleSta[<Drive>]/ActMaster[1..5]/AxIndex	
Value range	-	
Pre-assignment	-	
Unit	-	
Description of the variable values	System axis index of master axis	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt	
Dependency	-	
Data type	Short (2 Byte)	

Active offset of slave axis to master axis of the system axis coupling

Active offset of slave axis to master axis in system axis coupling	Offset

Returns an active positional offset of the respective slave axis to the master axis.

This system date is updated with the master axis switching and especially with real-time-conditioned master axis switching.

Path	/SysAxCoupleSta[<Drive>]/ActMaster[1..5]/Offset	
Value range	-	
Pre-assignment	-	
Unit	mm or °	
Description of the variable values	Positional offset of slave axis to master axis.	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt	
Dependency	-	
Data type	Double_t (8 bytes)	

Active input revolutions of the master axis of the system axis coupling

Active input revolutions of the master axis of the system axis coupling	N_Input

Returns the active input revolution ratio of the master axis to the slave axis in the system axis coupling.

This system date is updated with the master axis switching and especially with real-time-conditioned master axis switching.

Path	/SysAxCoupleSta[<Drive>]/ActMaster[1..5]/N_Input	
Value range	-	
Pre-assignment	-	
Unit	-	
Description of the variable values	Active number of the input revolutions of the master axis to be performed per output revolutions (N_Output) of the slave axis.	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt	
Dependency	-	
Data type	Double_t (8 bytes)	

Active output revolutions of the slave axis of the system axis coupling

Active output revolutions of the slave axis of the system axis coupling	N_Output

It provides the active output revolution ratio of the master axis to the slave axis in the system axis coupling.

System data

This system date is updated with the master axis switching and especially with real-time-conditioned master axis switching.

Path	/SysAxCoupleSta[<Drive>]/ActMaster[1..5]/N_Output		
Value range	-		
Pre-assignment	-		
Unit	-		
Description of the variable values	Active number of output revolutions of the slave axis per input revolutions (N_Input) of the master axis performed.		
Access rights	MTB:	Read	
	Setter:	Read	
	User:	Read	
	NC prog.:	Read	
	PLC prog.:	Read	
Activation	-		
Update	By the NC kernel asynchronously in the interpolator Interrupt		
Dependency	-		
Data type	Double_t (8 bytes)		

Active master axis offset in the system axis coupling

Active master axis offset in the system axis coupling	Offset

Returns the active master axis position shift of the axis "AxIndex".
This system date is updated with the master axis switching and especially with real-time-conditioned master axis switching.

Path	/SysAxCoupleSta[<Drive>]/ActMaster[1..5]/Shift		
Value range	-		
Pre-assignment	-		
Unit	mm or °		
Description of the variable values	Active master axis position offset.		
Access rights	MTB:	Read	
	Setter:	Read	
	User:	Read	
	NC prog.:	Read	
	PLC prog.:	Read	
Activation	-		
Update	By the NC kernel asynchronously in the interpolator Interrupt		
Dependency	-		
Data type	Double_t (8 bytes)		

Active table for the system axis coupling

Active table for the system axis coupling	Tab

Returns the active coupling table of the respective system slave/system master axis.
This system date is updated with the master axis switching and especially with real-time-conditioned master axis switching.

Path	/SysAxCoupleSta[<Drive>]/ActMaster[1..5]/Tab	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	<Empty-String>	No table coupling active.
	"<tab>"	Activate table coupling with the <tab> table.
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel asynchronously in the interpolator Interrupt	
Dependency	-	
Data type	String_t (30 bytes for text + 1 byte + 1 byte for end identification)	

32.2.11 System axis coupling tables"/SysAxCoupleTab"

Table source file of the system axis coupling table

Table source file		FileName
Name and storage location of the original table file creating the loaded Spline table using CoupleSplineTab(...).		
Path	/SysAxCoupleTab[<CacheIndex>]/FileName	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	<Empty-String>	Table location is not assigned.
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	CST(TAB(<tab>),LOAD) - enters a new table	
	CST(TAB(<tab>),UNLOAD) - removes a table	
Dependency	-	
Data type	String_t (100 characters + 1 byte + 1 byte for end identification)	

State of the system axis coupling table

Coupling table state	State
The state indicates, whether the table loaded can be activated or deleted.	
Path	/SysAxCoupleTab[<CacheIndex>]/State

System data

Value range	-
Pre-assignment	0
Unit	-
Description of the variable values	0 Table location is not assigned 1 Table is currently loaded 2 Table is loaded 3 Table is active 4 Table is removed
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	CST(TAB(<tab>),LOAD) - enters a new table CST(TAB(<tab>),UNLOAD) - removes a table SysAxCoupleCmd[<axis>].Master[i].Tab = "<Tab>" - activates the table SysAxCoupleCmd[<axis>].Master[i].Tab = <empty string> - deactivates the table Note: SysAxCoupleCmd becomes only active when SysAxCoupleCtr.Valid = TRUE active
Dependency	-
Data type	Short_t (2 bytes)

Time stamp of the system axis coupling table

Time stamp		FileTime
Specifies the change date of the source file at the moment of loading.		
Path	/SysAxCoupleTab[<CacheIndex>]/FileTime	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	dd.mm.yy hh:mm:ss	
Description of the variable values	Time stamp in string format specifying	
	dd	1..31 day
	mm	1..12 month
	yy	00..99 year
	hh	00..24 hours
	mm	00..59 minutes
	ss	0..59 seconds
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read

Activation	-
Update	CST(TAB(<tab>),LOAD) - enters a new table CST(TAB(<tab>),UNLOAD) - removes a table
Dependency	-
Data type	String_t (18 bytes + 1 byte + 1 byte for end identification)

Memory assignment of the system axis coupling table

	Memory assignment	CacheSize
	Specifies the size of the assigned table memory.	
Path	/SysAxCoupleTab[<CacheIndex>]/CacheSize	
Value range	-	
Pre-assignment	0	
Unit	Bytes	
Description of the variable values	-	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	CST(TAB(<tab>),LOAD) - enters a new table CST(TAB(<tab>),UNLOAD) - removes a table	
Dependency	-	
Data type	Int_t (4 bytes)	

32.2.12 Block pre-run data for axes "/SysAxSRun"

General information

The following structural elements are defined in the "sdaxis.xsd" schema:

Structural element	Description
NotAdjusted	Axis not adjusted
PosNotAdjusted	Axis position not adjusted
ConfNotAdjusted	Channel assignment not adjusted
ReqPosAdjust	Position adjustment required
TargChan	Target channel
ReleaseChan	Release channel

Tab. 32-6: Overview on the "SysAxSRun" structural elements

Axis not adjusted

Axis not adjusted	NotAdjusted

Provides the information that an axis is not adjusted after block pre-run.

An axis is labeled with "not adjusted" if it is not in the channel. Using the technological axis adjustment to generate the signal, the "Axis position not adjusted" element can be considered.

System data

Path	/SysAxSRun[<Drive>]/NotAdjusted	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Axis is adjusted
	1	Axis is not adjusted
Access rights	MTB:	Read (NC kernel exclusively writes on the date)
	Setter:	Read (NC kernel exclusively writes on the date)
	Operator:	Read (NC kernel exclusively writes on the date)
Activation	-	
Update	After block pre-run in adjustment	
Dependency	-	
Data type	1 x Byte_t (1 Byte)	

Axis position not adjusted

Axis position not adjusted		PosNotAdjusted
Provides the information that the position of an axis is not adjusted after block pre-run.		
The signal is only formed for axes belonging to the target configuration of the channel in which the block pre-run was executed. The information "Axis position not adjusted" is only given when the axis is also in its target channel.		
Path	/SysAxSRun[<Drive>]/PosNotAdjusted	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Axis position is adjusted
	1	Axis position is not adjusted
Access rights	MTB:	Read (NC kernel exclusively writes on the date)
	Setter:	Read (NC kernel exclusively writes on the date)
	Operator:	Read (NC kernel exclusively writes on the date)
Activation	-	
Update	After block pre-run during adjustment.	
Dependency	-	
Data type	1 x Byte_t (1 Byte)	

Channel assignment not adjusted

Channel assignment not adjusted		ConfNotAdjusted
It provides the information that the channel assignment axis is not adjusted to its target position after block pre-run.		
<i>The channel assignment of an axis is not adjusted if</i>		
- the axis belongs to the target configuration and currently not to the block pre-run channel. - the axis does not belong to the target configuration but currently belongs to the block pre-run channel.		
Path	/SysAxSRun[<Drive>]/ConfNotAdjusted	

Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Channel assignment is adjusted
	1	Channel assignment is not adjusted
Access rights	MTB:	Read (NC kernel exclusively writes on the date)
	Setter:	Read (NC kernel exclusively writes on the date)
	Operator:	Read (NC kernel exclusively writes on the date)
Activation	-	
Update	After block pre-run during adjustment	
Dependency	-	
Data type	1 x Byte_t (1 Byte)	

Position adjustment required

Position adjustment required		ReqPosAdjust
Provides the information that an axis is required for a position adjustment after block pre-run.		
The control writes on the date after block pre-run depending on the channel system date /SysChSRun.		
Path	/SysAxSRun[<Drive>]/ReqPosAdjust	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Axis position does not have to be adjusted
	1	Axis position has to be adjusted
Access rights	MTB:	Read (NC kernel exclusively writes on the date)
	Setter:	Read (NC kernel exclusively writes on the date)
	Operator:	Read (NC kernel exclusively writes on the date)
Activation	-	
Update	After block pre-run during adjustment	
Dependency	-	
Data type	1 x Byte_t (1 Byte)	

Target channel

Target channel		TargChan
Channel to which the axis has to be assigned to after block pre-run.		
The date only indicates a value as long as the axis is not yet in the target channel.		
Path	/SysAxSRun[<Drive>]/TargChan	
Value range	0 - 60	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Axis is in the target channel
	1.. 60	Channel index of the target channel

System data

Access rights	MTB:	Read (read-only)
	Setter:	Read (read-only)
	Operator:	Read (read-only)
Activation	-	
Update	After block pre-run in adjustment	
Dependency	-	
Data type	1 x Int_t (1 Byte)	

Release channel

Release channel		ReleaseChan
Channel to which the axis has to be assigned to after block pre-run. The date only indicates a value as long as the axis has to be released from the target channel.		
Path	/SysAxSRun[<Drive>]/ReleaseChan	
Value range	0 - 60	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Axis has already been released from the target channel.
	1.. 60	Channel index of the target channel from which the axis has to be released.
Access rights	MTB:	Read (read-only)
	Setter:	Read (read-only)
	Operator:	Read (read-only)
Activation	-	
Update	After block pre-run in adjustment	
Dependency	-	
Data type	1 x Int_t (1 Byte)	

32.2.13 Data of the active axis transformation "/SysAxTrafo1" and "/SysAxTrafo2"

General information

System data structure SysAxTrafo1 contains the data for the axis transformation active on point of contact 1 (AT1).

SysAxTrafo2 has the same data structure and contains the data for the axis transformation active on point of contact 2 (AT2).

If no axis transformation is active, all entries of these system data receive a value of "0".

The following structural elements are defined in the "sdaxtrf.xsd" schema:

Structural element	Description
AxTrafoNo	Axis transformation number
Type	Type of axis transformation
AxisAssignment	System axes/coordinates of the axis transformation
AxisClassification	System axis/coordinate classification of the axis transformation

Structural element	Description
AxZeroPos	Axis position of the reference position
LenParam	Length and angle parameters of the axis transformation
TCSOrientation	Reference position of tool coordinate system
EpsilonRanges	Epsilon environments

Tab. 32-7: Overview on the structural elements "SysAxTrafo1", "SysAxTrafo2"

Axis transformation number

Axis transformation number	AxTrafoNo

For every channel, returns the number of the active axis transformation.

Path /SysAxTrafo1[<Channel>]/AxTrafoNo

/SysAxTrafo2[<Channel>]/AxTrafoNo

CPL: SD.SysAxTrafo1[<Channel>].AxTrafoNo

CPL: SD.SysAxTrafo2[<Channel>].AxTrafoNo

Value range -

Pre-assignment 0

Unit -

Description of the variable values

0 No axis transformation active.

1.. 20 Axis transformation from the respective MP block is active, e.g. Coord (4) is active.

1000 Axis transformation is activated via the system date, e.g. Coord(SD.MTBxAxTrafo) active.

Access rights **MTB:** Read

Setter: Read

User: Read

NC prog.: Read

PLC prog.: Read

Activation -

Update By the NC kernel as soon as the corresponding axis transformation becomes active.

Dependency -

Data type Int_t (4 bytes)

Type of axis transformation

Type of axis transformation	Type

For every channel, returns the type of the axis transformation.

Path /SysAxTrafo1[<Channel>]/Type

/SysAxTrafo2[<Channel>]/Type

CPL: SD.SysAxTrafo1[<Channel>].Type

CPL: SD.SysAxTrafo2[<Channel>].Type

Value range -

Pre-assignment 0

Unit -

System data

Description of the variable values	Corresponds to machine parameter "TrafoType" in the path "TRA/AxTrafo[1..20]"(Macoda no. 1030 00110).	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel as soon as the corresponding axis transformation becomes active.	
Dependency	-	
Data type	Int_t (4 bytes)	

System axes/coordinates of the axis transformation

System axes/coordinates of the axis transformation		AxisAssignment[1..8]
For every channel, provides the indices of the system axes/coordinates at the input of the forward transformation.		
Path	/SysAxTrafo1[<Channel>]/AxisAssignment[1..8] /SysAxTrafo2[<Channel>]/AxisAssignment[1..8] CPL: SD.SysAxTrafo1[<Channel>].AxisAssignment[1..8] CPL: SD.SysAxTrafo2[<Channel>].AxisAssignment[1..8]	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Corresponds to the machine parameter "FwdInCoordIndTrafo[1..8]" in the path "TRA/AxTrafo[1..20]/FwdInCoord" (Macoda no. 1030 00120).	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel as soon as the corresponding axis transformation becomes active.	
Dependency	-	
Data type	8 x Short_t (8 x 2 Byte)	

System axis/coordinate classification of the axis transformation

System axis/coordinate classification of the axis transformation	AxisClassification[1..8]
For every channel, provides the axes participating in the transformation, including the direction of rotation/movement.	
Path	/SysAxTrafo1[<Channel>]/AxisClassification[1..8]

	/SysAxTrafo2[<Channel>]/AxisClassification[1..8]
	CPL: SD.SysAxTrafo1[<Channel>].AxisClassification[1..8]
	CPL: SD.SysAxTrafo2[<Channel>].AxisClassification[1..8]
Value range	-
Pre-assignment	0
Unit	-
Description of the variable values	Corresponds to the machine parameters "CoordClass" and "CoordDir" in the path "TRA/AxTrafo[1..20]/FwdInCoordClass/FwdInCoord[1..8]" (Macoda no. 1030 00125).
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	By the NC kernel as soon as the corresponding axis transformation becomes active.
Dependency	-
Data type	8 x Int_t (8 x 4 bytes)

Axis position of the reference position

Axis position of the reference position		AxZeroPos[1..8]
	Returns the axis positions that result in the reference position when axis transformation is active for each axis position.	
Path	/SysAxTrafo1[<Channel>]/AxZeroPos[1..8] /SysAxTrafo2[<Channel>]/AxZeroPos[1..8] CPL: SD.SysAxTrafo1[<Channel>].AxZeroPos[1..8] CPL: SD.SysAxTrafo2[<Channel>].AxZeroPos[1..8]	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Corresponds to machine parameter "RefPosTrafo" in the path "TRA/AxTrafo[1..20]/RefPos"(Macoda no. 1030 00130).	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel as soon as the corresponding axis transformation becomes active.	
Dependency	-	
Data type	8 x Double_t (8 x 8 Bytes)	

Length and angle parameter of the axis transformation

Length and angle parameters of the axis transformation		LenParam[1..16]
	For every channel, provides the geometry of the axis kinematics, i.e. the distances and angles of the transformation axes in their reference positions.	
Path	/SysAxTrafo1[<Channel>]/LenParam[1..16] /SysAxTrafo2[<Channel>]/LenParam[1..16] CPL: SD.SysAxTrafo1[<Channel>].LenParam[1..16] CPL: SD.SysAxTrafo2[<Channel>].LenParam[1..16]	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Corresponds to the machine parameter "JointParTrafo[1..16]" in the path "TRA/AxTrafo[1..20]/JointPar" (Macoda no. 1030 00140).	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel as soon as the corresponding axis transformation becomes active.	
Dependency	The meaning of the individual values is determined by the axis transformation type.	
Data type	16 x Double_t (16 x 8 bytes)	

Reference orientation of tool coordinate system

Reference position of tool coordinate system		TCSOrientation[1..8]
	Returns the orientation of the tool coordinate system (TCS) in terms of the basic workpiece coordinate system (BCS) in the reference position for each channel.	
Path	/SysAxTrafo1[<Channel>]/TCSOrientation[1..8] /SysAxTrafo2[<Channel>]/TCSOrientation[1..8] CPL: SD.SysAxTrafo1[<Channel>].TCSOrientation[1..8] CPL: SD.SysAxTrafo2[<Channel>].TCSOrientation[1..8]	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Corresponds to the machine parameters in the path "TRA/AxTrafo[1..20]/TcsRefOri" (Macoda no. 1030 00150) The coordinate direction is used as the sign of the system datum.	
Access rights	MTB: Read Setter: Read User: Read	

	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel as soon as the corresponding axis transformation becomes active.	
Dependency	-	
Data type	8 x Byte_t (8 x 1 byte)	

Epsilon environments

Epsilon environments		EpsilonRanges
	Returns the "epsilon environment" of the active axis transformation for each channel.	
Path	/SysAxTrafo1[<Channel>]/EpsilonRanges /SysAxTrafo2[<Channel>]/EpsilonRanges CPL: SD.SysAxTrafo1[<Channel>].EpsilonRanges CPL: SD.SysAxTrafo2[<Channel>].EpsilonRanges	
Value range	-	
Pre-assignment	0	
Unit	mm or °	
Description of the variable values	Corresponds to machine parameter "EpsAxTrafo" in the path "TRA/AxTrafo[1..20]" (Macoda no. 1030 00160).	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel as soon as the corresponding axis transformation becomes active.	
Dependency	-	
Data type	Float_t (4 bytes)	

32.2.14 Channel command data "/SysChCmd"

Override feed2

Override feed2 (F2 value)		Feedrate2Ovr
	Feed2 (F2 value) can be scaled with the system date additionally to the default override (value between 0 and 100%). The effect is multiplicative. A change in the system date is applied directly in the interpolator.	
Path	/SysChCmd[<Channel>]/Feedrate2Ovr	
Value range	0 no feed 100 programmed F2 value	
Pre-assignment	100	
Unit	% value	
Description of parameter values	Override of the second feed (feed2)	

System data

Access rights	MTB:	Read, write
	Setter:	Read, write
	Operator:	Read, write
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Float_t (4 bytes)	

32.2.15 SysCfgInfo "/SysCfgInfo"

General information

The checksum is saved via the machine parameters in this volatile system date. The checksum is recalculated in case of reset or SoftReset of the control. If the machine parameter changes, a new checksum is generated after reset or SoftReset.

The structure for the checksum is defined in the "sdcfginfo.xsd" schema.

Structural element	Description
ConfDataChecksum	Checksum via machine parameters Data type: Double_t(8Bytes)

Tab. 32-8: Overview on the structure element "SysCfgInfo"

32.2.16 Active channel spindle data "/SysChSpActData"

General information

The system data structure SysChSpActData is created for each channel in the volatile memory.

It comprises active channel spindle data such as data from a spindle-caused channel waiting state.

The following structural elements are defined in the "sdspndl.xsd" schema:

Structural element	Description
WaitForSpindle	Data from a spindle-caused channel waiting state
WaitForSpindle.Active	Spindle-caused channel waiting state active
WaitForSpindle.ChanSpNo	Number of the channel spindle in waiting state
WaitForSpindle.SysSpNo	Number of the system spindle in waiting state
WaitForSpindle.Move	Active movement waiting condition
WaitForSpindle.Gear	Active gear waiting condition

Tab. 32-9: Overview on structural elements "SysChSpActData"

Spindle-caused channel waiting state active

Spindle-caused channel waiting state active	WaitForSpindle.Active

Indicates whether a spindle-caused channel waiting state is present.

Path /SysChSpActData[<Channel>]/WaitForSpindle/Active
 CPL: SD.SysChSpActData[<Channel>].WaitForSpindle.Active - or -
 SD.SysChSpActData.WaitForSpindle.Active in the individual channel

Value range -

Pre-assignment	0	
Unit	-	
Description of the variable values	0	No spindle-caused channel waiting state is active. The remaining data in "WaitForSpindle" refer to the latest active waiting state.
	1	A spindle-caused channel waiting state is active.
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Number of the channel spindle in waiting state

Number of the channel spindle in waiting state		WaitForSpindle.ChanSpNo
	Indicates the number of the channel spindle currently in waiting state.	
Path	/SysChSpActData[<Channel>]/WaitForSpindle/ChanSpNo CPL: SD.SysChSpActData[<Channel>].WaitForSpindle.ChanSpNo - or - SD.SysChSpActData.WaitForSpindle.ChanSpNo in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No channel spindle in waiting state
	>0	Number of the channel spindle in waiting state
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	The data is only valid if WaitForSpindle.Active > 0.	
Data type	UnsignedByte_t (1 Byte)	

Number of the system spindle in waiting state

Number of the system spindle in waiting state		WaitForSpindle.SysSpNo
	Indicates the respective system spindle number of the channel spindle currently in waiting state.	
Path	/SysChSpActData[<Channel>]/WaitForSpindle/SysSpNo CPL: SD.SysChSpActData[<Channel>].WaitForSpindle.SysSpNo - or - SD.SysChSpActData.WaitForSpindle.SysSpNo in the individual channel	

System data

Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No channel spindle in waiting state
	>0	System spindle number of the channel spindle in waiting state
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	The data is only valid if WaitForSpindle.Active > 0.	
Data type	UnsignedShort_t (2 bytes)	

Active movement waiting condition

Active movement waiting condition		WaitForSpindle.Move
	It contains the active movement waiting condition for the active spindle-caused channel waiting state.	
Path	/SysChSpActData[<Channel>]/WaitForSpindle/Move CPL: SD.SysChSpActData[<Channel>].WaitForSpindle.Move - or - SD.SysChSpActData.WaitForSpindle.Move in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No active movement waiting condition
	3	Channel waits for "Spindle speed reached" (iSp_ProgSpReach) for M3
	4	Channel waits for "Spindle speed reached" (iSp_ProgSpReach) for M4
	5	Channel waits for "Spindle stop" (iSp_Stop)
	19	Channel waits for "Spindle is oriented" (iSp_OrientateFinish)
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	The data is only valid if WaitForSpindle.Active > 0.	

Data type UnsignedByte_t (1 Byte)

Active gear waiting condition

	Active gear waiting condition	WaitForSpindle.Gear
	It contains the active gear waiting condition for the active spindle-caused channel waiting state.	
Path	/SysChSpActData[<Channel>]/WaitForSpindle/Gear CPL: SD.SysChSpActData[<Channel>].WaitForSpindle.Gear - or - SD.SysChSpActData.WaitForSpindle.Gear in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No active gear waiting condition
	1	Channel waits for the end of a programmed gear switching
	2	Channel waits for the end of an automatic gear switching
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	The data is only valid if WaitForSpindle.Active > 0.	
Data type	UnsignedByte_t (1 Byte)	

32.2.17 Spindle pre-run data "/SysChSpSRun"

General information

The system data structure "SysChSpSRun" is created for each channel in the volatile memory.

It contains the command state of the channel spindles at the end of a pre-run. This information can be used within an adjustment program (see SD.SysSRun.NameOfAsup).

The system data structure consists of three parts:

- General data
- Data channel spindles
- Data for spindle couplings

The following structural elements are defined in the schema "sdspndl.xsd".

General data (SysChSpSRun):

Structure element	Description
MaxSpNumber	Highest spindle number available in the channel
PrgSpNumber	Highest programmed spindle number in pre-run
PrgCpNumber	Highest programmed coupling number in pre-run
ConstCutStatus	Adjust G-function group G96/97

System data

Structure element	Description
SpeedStr	Speed syntax of a spindle
MoveStr	Movement syntax of a spindle
GearStr	Gear syntax of a spindle
SpConfNotAdjusted	Spindle configuration adjustment required

Tab. 32-10: Overview on structure elements "SysChSpSRun" (general data)
Data channel spindles (SysChSpSRun.Sp[<ChannelSpindleNumber>]):

Structure element	Description
SysSpNo	System spindle number
DriveNo	Drive number
IsProg	Spindle programmed in pre-run
NotAdjusted	Adjustment required
MoveNotAdj	Adjustment status of the movement function
GearNotAdj	Adjustment status of the gear
SpeedNotAdj	Adjustment status of the speed
LimitNotAdj	Adjustment status of the speed limitation
CcvNotAdj	Adjustment status of the G96 state
CAxisNotAdj	Adjustment status of c-axis required
OpModeNotAdj	Adjustment status of the drive operation mode
CpDataNotAdj	Adjustment status of the coupling data
CoupleNotAdj	Adjustment status of the coupling state
Move	Movement function
Gear	Gear level
AutoGearShift	Automatic gear shifting
CAxisOn	Switch on c-axis
CAxisType	C axis type
CcvOn	Switch on constant cutting velocity
CcvAxis	Channel reference axis for constant cutting velocity
CcvCoordSystem	Coordinate system for constant cutting velocity
PosCtrlOn	activates positioning interface
OriPos	Orientation position
CcvVelocity	Cutting velocity
ProgSpeed	Programmed speed
ProgMinSpeed	Minimum speed
ProgMaxSpeed	Maximum speed
CoupleDistance	Coupling distance
SyncWindow	Synchronization window for coupling
SyncErrorWindow	Synchronization window error for coupling

Structure element	Description
CouplePosOffset	Angular offset in case of active coupling
CouplePosOffsetSpeed	Speed to approach the angular offset

Tab. 32-11: Overview on structure elements "SysChSpSRun" (data for channel spindles)

Data for spindle couplings (SysChSpSRun.Cp[<CouplingNumber>]):

Structure element	Description
IsProg	Couple link programmed in pre-run
NotAdjusted	Adjustment status of spindle coupling
CoupleOn	Switch on couple link
Master	Channel spindle number of the master spindle
Slave[1..8]	Channel spindle number of the slave spindle

Tab. 32-12: Overview on structure elements "SysChSpSRun" (data for spindle couplings)

Highest spindle number available in the channel

Highest spindle number available in the channel	MaxSpNumber

Indicates the highest spindle number available in the channel.

Path /SysChSpSRun[<Channel>]/MaxSpNumber
 CPL: SD.SysChSpSRun[<Channel>].MaxSpNumber - or -
 SD.SysChSpSRun.MaxSpNumber in the individual channel

Value range -

Pre-assignment 0

Unit -

Description of the variable values Highest spindle number available in the channel

Access rights **MTB:** Read
Setter: Read
User: Read
NC prog.: Read
PLC prog.: Read

Activation -

Update By the NC kernel

Dependency -

Data type UnsignedByte_t (1 Byte)

Highest spindle number programmed in pre-run

Highest programmed spindle number in pre-run	PrgSpNumber

Indicates the highest spindle number programmed in the pre-run.

Path /SysChSpSRun[<Channel>]/PrgSpNumber
 CPL: SD.SysChSpSRun[<Channel>].PrgSpNumber - or -
 SD.SysChSpSRun.PrgSpNumber in the individual channel

Value range -

System data

Pre-assignment	0
Unit	-
Description of the variable values	Highest programmed spindle number in pre-run
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	By the NC kernel
Dependency	-
Data type	UnsignedByte_t (1 Byte)

Highest coupling number programmed in pre-run

Highest programmed coupling number in pre-run		PrgCpNumber
	Indicates the highest coupling number programmed in pre-run.	
Path	/SysChSpSRun[<Channel>]/PrgCpNumber CPL: SD.SysChSpSRun[<Channel>].PrgCpNumber - or - SD.SysChSpSRun.PrgCpNumber in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Highest programmed coupling number in pre-run	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Adjust G-function group G96/97

Adjust G-function group G96/97		ConstCutStatus
	Indicates whether the G-function group G96/97 has to be adjusted.	
Path	/SysChSpSRun[<Channel>]/ConstCutStatus CPL: SD.SysChSpSRun[<Channel>].ConstCutStatus - or - SD.SysChSpSRun.ConstCutStatus in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No action required

	1	Switch off constant cutting velocity
	2	Switch off and on constant cutting velocity.
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Speed syntax of a spindle

Speed syntax of a spindle		SpeedStr
	Indicates the speed syntax of a spindle (e.g. "S2="). The value is entered by the NC function "SPADJSTR <SpindleNo.>".	
Path	/SysChSpSRun[<Channel>]/SpeedStr CPL: SD.SysChSpSRun[<Channel>].SpeedStr -or - SD.SysChSpSRun.SpeedStr in the individual channel	
Value range	8 characters max. from isoLatin1String_t	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	Speed syntax of a spindle	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	Implicitly with the NC function "SPADJSTR"	
Dependency	-	
Data type	Str8_t (8 bytes for text + 1 byte for identification)	

Motion syntax of a spindle

Motion syntax of a spindle		MoveStr
	Indicates the motion syntax of a spindle (e.g. "M203"). The value is entered by the NC function "SPADJSTR <SpindleNo.>".	
Path	/SysChSpSRun[<Channel>]/MoveStr CPL: SD.SysChSpSRun[<Channel>].MoveStr - or - SD.SysChSpSRun.MoveStr in the individual channel	
Value range	8 characters max. from isoLatin1String_t	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	Motion syntax of a spindle	

System data

Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	Implicitly with the NC function "SPADJSTR"	
Dependency	-	
Data type	Str8_t (8 bytes for text + 1 byte for identification)	

Gear syntax of a spindle

Gear syntax of a spindle		GearStr
Indicates the gear stage syntax of a spindle (e.g. "M241"). The value is entered by the NC function "SPADJSTR <SpindleNo.>".		
Path	/SysChSpSRun[<Channel>]/GearStr CPL: SD.SysChSpSRun[<Channel>].GearStr -or - SD.SysChSpSRun.GearStr in the individual channel	
Value range	8 characters max. from isoLatin1String_t	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	Gear syntax of a spindle	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	Implicitly with the NC function "SPADJSTR"	
Dependency	-	
Data type	Str8_t (8 bytes for text + 1 byte for identification)	

System spindle number

System spindle number		Sp.SysSpNo
Indicates the system spindle number (1..60).		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/SysSpNo CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].SysSpNo - or - SD.SysChSpSRun.Sp[<ChanSpindle>].SysSpNo in the individual channel	
Value range	0..60	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No system spindle
	1..60	Valid system spindle number
Access rights	MTB:	Read

Setter:	Read
User:	Read
NC prog.:	Read
PLC prog.:	Read
Activation	-
Update	By the NC kernel
Dependency	-
Data type	UnsignedShort_t (2 bytes)

Drive number

Drive number		Sp.DriveNo
Indicates the drive number (1..250).		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/DriveNo CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].DriveNo -or - SD.SysChSpSRun.Sp[<ChanSpindle>].DriveNo in the individual channel	
Value range	0..250	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No drive
	1..250	Valid drive number
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedShort_t (2 bytes)	

Spindle programmed in pre-run

Spindle programmed in pre-run		Sp.IsProg
Indicates whether the spindle was programmed in pre-run.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/IsProg CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].IsProg -or - SD.SysChSpSRun.Sp[<ChanSpindle>].IsProg in the individual channel	
Value range	0 - FALSE	
	1 - TRUE	
Pre-assignment	0 - FALSE	
Unit	-	
Description of the variable values	TRUE	Spindle programmed in pre-run
	FALSE	Spindle not programmed in pre-run
Access rights	MTB:	Read

System data

	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Boolean_t (1 byte)	

Adjustment required

Adjustment required		Sp.NotAdjusted
Indicates whether an adjustment is required (consider all elements Sp.xxxNotAdj).		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/NotAdjusted CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].NotAdjusted -or - SD.SysChSpSRun.Sp[<ChanSpindle>].NotAdjusted in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Adjustment not required
	1	Adjustment required
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Spindle configuration adjustment required

Spindle configuration adjustment required		SpConfNotAdjusted
Indicates, whether a spindle configuration adjustment is required.		
Path	/SysChSpSRun[<Channel>]/SpConfNotAdjusted CPL: SD.SysChSpSRun[<Channel>].SpConfNotAdjusted - or - SD.SysChSpSRun.SpConfNotAdjusted in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Adjustment not required
	1	Adjustment required. It is adjusted by the calling the NC function SpAdjSp

Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By NC kernel when calling GetSpindle, RemSpindle, DefSpindle and SpAdjSp.	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Adjustment status of the movement function

Adjustment status of the movement function		Sp.MoveNotAdj
Indicates the adjustment status of the movement status.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/MoveNotAdj CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].MoveNotAdj -or - SD.SysChSpSRun.Sp[<ChanSpindle>].MoveNotAdj in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Movement function is adjusted
	1	Incorrect movement function
	2	Movement function is adjusted bit orientation position is incorrect
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Adjustment status of the gear stage

Adjustment status of the gear stage		Sp.GearNotAdj
Indicates the adjustment status of the gear.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/GearNotAdj CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].GearNotAdj - or - SD.SysChSpSRun.Sp[<ChanSpindle>].GearNotAdj in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Gear is adjusted

System data

	1	Incorrect gear
	2	AutoGear wrong
	3	Gear and AutoGear are wrong
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Adjustment status of the speed

Adjustment status of the speed		Sp.SpeedNotAdj
Indicates the adjustment status of the speed.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/SpeedNotAdj CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].SpeedNotAdj - or - SD.SysChSpSRun.Sp[<ChanSpindle>].SpeedNotAdj in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Speed is adjusted
	1	Wrong speed
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Adjustment status of the speed limitation

Adjustment status of the speed limitation		Sp.LimitNotAdj
Indicates the adjustment status of the speed limitation.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/LimitNotAdj CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].LimitNotAdj - or - SD.SysChSpSRun.Sp[<ChanSpindle>].LimitNotAdj in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	

Description of the variable values	0	Speed limitation is adjusted
	1	Lower limit is wrong
	2	Upper limit is wrong
	3	Both the limits are wrong
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Adjustment status of the G96 state

Adjustment status of the G96 state		Sp.CcvNotAdj
Indicates the adjustment status of the G96 state (see SD.SysChSpSRun[<Channel>].ConstCutChange).		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CcvNotAdj CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CcvNotAdj -or - SD.SysChSpSRun.Sp[<ChanSpindle>].CcvNotAdj in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	G96 state is adjusted
	1	Not adjusted, constant cutting velocity must be switched off
	2	Not adjusted, constant cutting velocity must be enabled
	4	Wrong coordinate
	8	Wrong coordinate system
	16	Wrong cutting velocity
	12, 20, 24, 28	Several data is wrong
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Adjustment status of the C-axis

Adjustment status of C-axis	Sp.CAxisNotAdj
Indicates the adjustment status of the C-axis.	

System data

Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CAxisNotAdj CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CAxisNotAdj -or- SD.SysChSpSRun.Sp[<ChanSpindle>].CAxisNotAdj in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	C-axis is adjusted
	1	Not adjusted, C-axis must be disabled
	2	Not adjusted, C-axis must be enabled (take SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CAxisType into consideration)
	4	Incorrect gear
	8	Incorrect Sercos operation mode
	16	Other spindle data is wrong (e.g. speed)
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Adjustment status of the drive operation mode

Adjustment status of the drive operation mode		Sp.OpModeNotAdj
Indicates the adjustment status of the drive operation mode.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/OpModeNotAdj CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].OpModeNotAdj -or- SD.SysChSpSRun.Sp[<ChanSpindle>].OpModeNotAdj in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Drive operation mode is adjusted
	1	Drive interface is wrong state
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	

Data type UnsignedByte_t (1 Byte)

Adjustment status of the coupling data

Adjustment status of the coupling data		Sp.CpDataNotAdj
Indicates the adjustment status of the coupling data.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CpDataNotAdj CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CpDataNotAdj - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CpDataNotAdj in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Coupling data is adjusted
	1	Coupling distance is wrong
	2	Synchronization window is wrong
	4	Synchronization error window is wrong
	3, 5, 6, 7	Several data is wrong
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Adjustment status of the coupling state

Adjustment status of the coupling state		Sp.CoupleNotAdj
Indicates the adjustment status of the coupling state.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CoupleNotAdj CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CoupleNotAdj - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CoupleNotAdj in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Coupling state is adjusted
	1	Not adjusted; coupling has to be disabled
	2	Not adjusted, coupling has to be enabled (also refer to SD.SysChSpSRun[<Channel>].Cp[<CoupleNmb>]). The command coupling direction can be derived from SysChSpSRun.Sp[<SpNmb>].Move or coupling-related from SD.SysChSpSRun[<Channel>].Cp[<CoupleNmb>].Slave[<SlaveNmb>].
	4	Active coupling distance is wrong

System data

	8	CpPosOffset is wrong
	16	CpPosOffsetSpeed is wrong
	24	Several data is wrong
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Motion function

Motion function		Sp.Move
	Specifies the command state of the movement function.	
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/Move CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].Move - or - SD.SysChSpSRun.Sp[<ChanSpindle>].Move in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	3	Clockwise rotation
	4	Counterclockwise rotation
	5	Stop
	13	Clockwise rotation with cooling
	14	Counterclockwise rotation with cooling
	19	Positioning
	254	Couple the slave spindle opposite the direction of rotation of the master spindle
	255	Couple the slave spindle in the direction of rotation of the master spindle
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Gear

Gear level	Sp.Gear
Specifies the command state of the gear.	
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/Gear

	CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].Gear - or - SD.SysChSpSRun.Sp[<ChanSpindle>].Gear in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Idle
	1..4	Gear level
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Automatic gear shift

Automatic gear shift		Sp.AutoGearShift
	Indicates whether an automatic gear shifting should take place.	
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/AutoGearShift CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].AutoGearShift - or - SD.SysChSpSRun.Sp[<ChanSpindle>].AutoGearShift in the individual channel	
Value range	0	FALSE
	1	TRUE
Pre-assignment	0	FALSE
Unit	-	
Description of the variable values	FALSE	No automatic gear shifting
	TRUE	Automatic gear shift
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Boolean_t (1 byte)	

Switching on the C-axis

Switch on C-axis	Sp.CAxisOn
Indicates whether the C-axis should be switched on.	

System data

Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CAxisOn CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CAxisOn - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CAxisOn in the individual channel	
Value range	0	FALSE
	1	TRUE
Pre-assignment	0	FALSE
Unit	-	
Description of the variable values	FALSE	Do not switch on C-axis
	TRUE	Switch on C-axis
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Boolean_t (1 byte)	

C-axis type

C axis type		Sp.CAxisType
Specifies the C-axis type for a C-axis to be enabled (if SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CAxisOn = TRUE).		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CAxisType CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CAxisType - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CAxisType in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Asynchronous C-axis
	1	Synchronous C-axis
	16	Asynchronous CST axis
	17	Synchronous CST axis
	33	Tapping C-axis (error!)
	<Value> AND 1	Synchronous axis
	<Value> AND 16	CST axis
	<Value> AND 32	Tapping
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read

Activation	-
Update	By the NC kernel
Dependency	-
Data type	UnsignedByte_t (1 Byte)

Switching on the constant cutting velocity

Switch on constant cutting velocity		Sp.CcvOn
Specifies whether the constant cutting velocity should be switched on (see SD.SysChSpSRun[<Channel>].ConstCutChange).		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CcvOn CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CcvOn - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CcvOn in the individual channel	
Value range	0	FALSE
	1	TRUE
Pre-assignment	0	FALSE
Unit	-	
Description of the variable values	FALSE	No constant cutting velocity
	TRUE	Switch on constant cutting velocity
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Boolean_t (1 byte)	

Channel reference axis for constant cutting velocity

Channel reference axis for constant cutting velocity		Sp.CcvAxis
Indicates the channel axis number of the reference axis for the constant cutting velocity activated with CcvOn (see SD.SysChSpSRun[<Channel>].ConstCutChange).		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CcvAxis CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CcvAxis - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CcvAxis in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	1..8	valid channel reference axis for constant cutting velocity
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read

System data

	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	It is only applied if SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CcvOn = TRUE	
Data type	UnsignedByte_t (1 Byte)	

Coordinate system for constant cutting velocity

Coordinate system for constant cutting velocity		Sp.CcvCoordSystem
Indicates the coordinate system for the constant cutting velocity activated with CcvOn (see SD.SysChSpSRun[<Channel>].ConstCutChange).		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CcvCoordSystem CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CcvCoordSystem - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CcvCoordSystem in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	1	PCS
	2	MCS1
	3	MCS0
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	It is only applied if SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CcvOn = TRUE	
Data type	UnsignedByte_t (1 Byte)	

Switching on the position interface

Activate position interface		Sp.PosCtrlOn
Indicates whether the positioning interface should be switched on.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/PosCtrlOn CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].PosCtrlOn - or - SD.SysChSpSRun.Sp[<ChanSpindle>].PosCtrlOn in the individual channel	
Value range	0	FALSE
	1	TRUE
Pre-assignment	0	FALSE
Unit	-	

Description of the variable values	FALSE	Do not switch on positioning interface
	TRUE	Activate position interface
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Boolean_t (1 byte)	

Orientation position

Orientation position		Sp.OriPos
Indicates the orientation position.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/OriPos CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].OriPos - or - SD.SysChSpSRun.Sp[<ChanSpindle>].OriPos in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	Degree	
Description of the variable values	Orientation position	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Float_t (4 bytes)	

Cutting velocity

Cutting velocity		Sp.CcvVelocity
Indicates the value of the cutting velocity for the constant cutting velocity activated with CcvOn (see SD.SysChSpSRun[<Channel>].ConstCutChange).		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CcvVelocity CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CcvVelocity - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CcvVelocity in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	m/min	

System data

Description of the variable values	Value for constant cutting velocity
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	By the NC kernel
Dependency	It is only applied if SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CcvOn = TRUE
Data type	Float_t (4 bytes)

Programmed speed

Programmed speed		Sp.ProgSpeed
	Indicates the programmed speed.	
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/ProgSpeed CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].ProgSpeed - or - SD.SysChSpSRun.Sp[<ChanSpindle>].ProgSpeed in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	rpm	
Description of the variable values	Programmed speed	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Float_t (4 bytes)	

Minimum speed

Minimum speed		Sp.ProgMinSpeed
	Indicates the minimum speed.	
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/ProgMinSpeed CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].ProgMinSpeed - or - SD.SysChSpSRun.Sp[<ChanSpindle>].ProgMinSpeed in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	rpm	
Description of the variable values	Minimum speed	

Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Float_t (4 bytes)	

Maximum speed

Maximum speed		Sp.ProgMaxSpeed
Indicates the maximum speed.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/ProgMaxSpeed CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].ProgMaxSpeed - or - SD.SysChSpSRun.Sp[<ChanSpindle>].ProgMaxSpeed in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	rpm	
Description of the variable values	Maximum speed	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Float_t (4 bytes)	

Coupling distance

Coupling distance		Sp.CoupleDistance
Indicates the coupling distance.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CoupleDistance CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CoupleDistance - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CoupleDistance in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	Degree	
Description of the variable values	Coupling distance	

System data

Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Float_t (4 bytes)	

Synchronization window for coupling

Synchronization window for coupling		Sp.SyncWindow
Indicates the synchronization window for the coupling.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/SyncWindow CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].SyncWindow - or - SD.SysChSpSRun.Sp[<ChanSpindle>].SyncWindow in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	Degree	
Description of the variable values	Synchronization window for coupling	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Float_t (4 bytes)	

Synchronization window error for coupling

Synchronization window error for coupling		Sp.SyncErrorWindow
Indicates the synchronization window error for coupling.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/SyncErrorWindow CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].SyncErrorWindow - or - SD.SysChSpSRun.Sp[<ChanSpindle>].SyncErrorWindow in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	Degree	
Description of the variable values	Synchronization window error for coupling	
Access rights	MTB:	Read

Setter:	Read
User:	Read
NC prog.:	Read
PLC prog.:	Read
Activation	-
Update	By the NC kernel
Dependency	-
Data type	Float_t (4 bytes)

Angular offset in case of active coupling

Angular offset in case of active coupling		Sp.CouplePosOffset
Indicates the angular offset in case of active coupling.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CouplePosOffset CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CouplePosOffset - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CouplePosOffset in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	Degree	
Description of the variable values	Angular offset in case of active coupling	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Float_t (4 bytes)	

Speed to enter the phase offset

Speed to enter the phase offset		Sp.CouplePosOffsetSpeed
Indicates the speed to enter angular offset specified in CouplePosOffset.		
Path	/SysChSpSRun[<Channel>]/Sp[<ChanSpindle>]/CouplePosOffsetSpeed CPL: SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>].CouplePosOffsetSpeed - or - SD.SysChSpSRun.Sp[<ChanSpindle>].CouplePosOffsetSpeed in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	rpm	

System data

Description of the variable values Speed to enter the phase offset

Access rights MTB: Read
 Setter: Read
 User: Read
 NC prog.: Read
 PLC prog.: Read

Activation -

Update By the NC kernel

Dependency -

Data type Float_t (4 bytes)

Couple link programmed in pre-run

Couple link programmed in pre-run	Cp.IsProg

Indicates whether the couple link was programmed in pre-run.

Path /SysChSpSRun[<Channel>]/Cp[<CoupleNmb>]/IsProg
 CPL: SD.SysChSpSRun[<Channel>].Cp[<CoupleNmb>].IsProg - or -
 SD.SysChSpSRun.Cp[<CoupleNmb>].IsProg in the individual channel

Value range 0 FALSE
 1 TRUE

Pre-assignment 0 FALSE

Unit -

Description of the variable values TRUE Couple link was programmed in pre-run
 FALSE Couple link was not programmed in pre-run

Access rights MTB: Read
 Setter: Read
 User: Read
 NC prog.: Read
 PLC prog.: Read

Activation -

Update By the NC kernel

Dependency -

Data type Boolean_t (1 byte)

Adjustment status of the spindle coupling

Adjustment status of spindle coupling	Cp.NotAdjusted

Indicates the adjustment status of the spindle coupling.

Path /SysChSpSRun[<Channel>]/Cp[<CoupleNmb>]/NotAdjusted
 CPL: SD.SysChSpSRun[<Channel>].Cp[<CoupleNmb>].NotAdjusted - or -
 SD.SysChSpSRun.Cp[<CoupleNmb>].NotAdjusted in the individual channel

Value range -

Pre-assignment 0

Unit -

Description of the variable values 0 Coupling is adjusted

	1	Coupling was not adjusted. Incorrect master spindle
	2	Coupling is not adjusted. Incorrect slave spindles or incorrect coupling directions (also refer to SD.SysChSpSRun[<Channel>].Sp[<ChanSpindle>])
	3	Incorrect coupling state (also refer to SD.SysChSpSRun[<Channel>].Cp[CoupleNmb].CoupleOn)
	4	Coupling is not adjusted. At least one slave spindle has an incorrect coupling distance
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Switching on the coupling link

Switch on couple link		Cp.CoupleOn
	Indicates whether the coupling link should be switched on.	
Path	/SysChSpSRun[<Channel>]/Cp[<CoupleNmb>]/CoupleOn CPL: SD.SysChSpSRun[<Channel>].Cp[<CoupleNmb>].CoupleOn - or - SD.SysChSpSRun.Cp[<CoupleNmb>].CoupleOn in the individual channel	
Value range	0	FALSE
	1	TRUE
Pre-assignment	0	FALSE
Unit	-	
Description of the variable values	FALSE	Do not switch on coupling link
	TRUE	Switch on couple link
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Boolean_t (1 byte)	

Channel spindle number of the master spindle

Channel spindle number of the master spindle	Cp.Master
Indicates the channel spindle number of the master spindle.	
Path	/SysChSpSRun[<Channel>]/Cp[<CoupleNmb>]/Master

System data

	CPL: SD.SysChSpSRun[<Channel>].Cp[<CoupleNmb>].Master - or - SD.SysChSpSRun.Cp[<CoupleNmb>].Master in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	1..8	valid channel spindle number of the master spindle
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	UnsignedByte_t (1 Byte)	

Channel spindle number of the slave spindle

Channel spindle number of the slave spindle		Cp.Slave
	Contains the respective channel spindle number for each of the eight possible slaves.	
	If the channel spindle number is positive, the slave spindle has to be coupled to the direction of the master spindle.	
	If the channel spindle number is negative, the slave spindle has to be coupled to the direction of the master spindle.	
Path	/SysChSpSRun[<Channel>]/Cp[<CoupleNmb>]/Slave[1..8] CPL: SD.SysChSpSRun[<Channel>].Cp[<CoupleNmb>].Slave[1..8] - or - SD.SysChSpSRun.Cp[<CoupleNmb>].Slave[1..8] in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	1..8	valid channel spindle number of a slave spindle
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	8 x Byte_t (1 byte)	

32.2.18 Block pre-run data"/SysChSRun"

General information

The following structural elements are defined in the "sdsrun.xsd" schema:

Structural element	Description
Prep/ChAxNames[1..8]	Channel axis names in the block preparation
Act/ChAxNames[1..8]	Active channel axis names in adjustment
Targ/ChAxNames[1..8]	Channel axis names in the target block
PosAdjust[1..8]	Axis names for position adjustment
TargPos[1..8]	Target position of the approaching block
AcsTargPos[1..8]	Target position in the axis coordinate system
ActAxConf[1..8]	Current axis configuration in adjustment
PrepAxConf[1..8]	Axis configuration in the block preparation
TargAxConf[1..8]	Axis configuration in the target block
TechAxAdjust	Technological axis adjustment
Active	Computation active
ActMode	Block pre-run mode
TargAdjPrgActive	Target block adjustment program (TargAdjPrg) active
InitAdjPrgActive	Init adjustment program (InitAdjPrg) active
LookAheadPreRun	Block pre-run active in look-ahead range after target block
ChanNotAdjusted	Channel is not adjusted
NcfNotAdjusted	Modal NC functions are not adjusted
AuxNotAdjusted	Modal auxiliary functions are not adjusted
AxConfNotAdjusted	Axis configuration is not adjusted
CoordNotAdjusted	Axis transformation is not adjusted
ActStatus	Active adjustment state
TargBlockSrc	Target block code
ReentryInfoDir	File path on re-entry data
TargetBlkAvailable	Backed up target block available
LockTargBlock	Target blocks locked

Tab. 32-13: Overview on structure elements "SysChSRun"

Channel Axis Names in the Block Preparation

Channel axis names in the block preparation	Prep/ChAxNames[1..8]

Provides the programmable axis channel names of the block preparation in the adjustment after block pre-run.

The channel axis names of the channel axes configured in the block preparation are stored in ascending order in ChAxNames. Prep/ChAxNames is intended for the initialization program via CPL access. An axis name can be of up to eight characters.

Path /SysChSRun[<Channel>]/Prep/ChAxNames[1..8]

CPL: SD.SysChSRun[<Channel>].Prep.ChAxNames[1..8] - or -

SD.SysChSRun.Prep.ChAxNames[1..8] for access in the individual channel.

Value range Up to eight characters from IsoLatin1String

System data

Pre-assignment	<EmptyString>
Unit	-
Description of parameter values	Channel axis names in the block preparation. Depending on what was assigned at the moment of the block preparation, it can be system axis names, channel axis names or optional axis names.
Access rights	MTB: Read Setter: Read Operator: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	By the NC kernel (after computation during the block pre-run adjustment)
Dependency	-
Data type	8 x IsoLatin1String (8 bytes for text + 1 null byte + 3 bytes alignment)

Active channel axis names in adjustment

Active channel axis names in adjustment		Act/ChAxNames[1..8]
Provides the active channel axis names in adjustment after block pre-run.		
The active channel axis names of the channel axes configured are stored in ascending order in ChAxNames . Act/ChAxNames is intended for the display in the adjustment screen. An axis name can be of up to eight characters.		
Path	/SysChSRun[<Channel>]/Act/ChAxNames[1..8] CPL: SD.SysChSRun[<Channel>].Act.ChAxNames[1..8] - or - SD.SysChSRun.Act.ChAxNames[1..8] for access in the individual channel.	
Value range	Up to eight characters from IsoLatin1String	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Active channel axis names in adjustment. Depending on what was assigned at the active point in time, it can be system axis names, channel axis names or optional axis names.	
Access rights	MTB: Read Setter: Read Operator: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel (after computation during the block pre-run adjustment)	
Dependency	-	
Data type	8 x IsoLatin1String (8 bytes for text + 1 null byte + 3 bytes alignment)	

Channel axis names in the target block

Channel axis names in the target block	Targ/ChAxNames[1..8]
Provides the channel axis name valid in the target block after block preparation.	

	The channel axis names of the channel axes configured in front of the target block are stored in ascending order in ChAxNames . Targ/ChAxNames does not change during the adjustment. An axis name can be of up to eight characters.	
Path	/SysChSRun[<Channel>]/Targ/ChAxNames[1..8] CPL: SD.SysChSRun[<Channel>].Targ.ChAxNames[1..8] - or - SD.SysChSRun.TargChAxNames[1..8] for access in the individual channel.	
Value range	Up to eight characters from IsoLatin1String	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Names of the channel axes in the target block. Depending on what was assigned in front of the target block at the point in time, it can be system axis names, channel axis names or optional axis names.	
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel (after block pre-run computation)	
Dependency	-	
Data type	8 x IsoLatin1String (8 bytes for text + 1 null byte + 3 bytes alignment)	

Axis names for position adjustment

Axis names for position adjustment		PosAdjust[1..8]
	Names of axes whose position has to be adjusted during adjustment. The specified axes have to be adjusted if bit 4 is set in the system date SysChSRun/TechAxAdjust.	
Path	/SysChSRun[<Channel>]/PosAdjust[1..8] CPL: SD.SysChSRun[<Channel>].PosAdjust[1..8] - or - SD.SysChSRun.PosAdjust[1..8] for access in the individual channel.	
Value range	Up to eight characters from IsoLatin1String	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Names of the channel axes in the target block. Depending on what was assigned in front of the target block at the point in time, it can be system axis names, channel axis names or optional axis names.	
Access rights	MTB:	Read and write
	Setter:	Read and write
	Operator:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	

Data type 8 x IsoLatin1String (8 bytes for text + 1 null byte + 3 bytes alignment)

Target position of the approaching block

Target position of the approaching block		TargPos[1..8]
It provides the target position channel-specifically that is approached after block pre-run in the part program. It is specified in the current workpiece coordinate system.		
Path	/SysChSRun[<Channel>]/TargPos[1..8] CPL: SD.SysChSRun[<Channel>].TargPos[1..8] - or - SD.SysChSRun.TargPos[1..8] for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	mm or °	
Description of parameter values	Target position value	
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel (after block pre-run computation)	
Dependency	-	
Data type	8 x Double_t (8 x 8 Bytes)	

Target position in the axis coordinate system

Target position in the axis coordinate system		AcsTargPos[1..8]
It provides the target position of the approaching block channel-specifically related to the axis coordinate system (ACS).		
Path	/SysChSRun[<Channel>]/AcsTargPos[1..8] CPL: SD.SysChSRun[<Channel>].AcsTargPos[1..8] - or - SD.SysChSRun.AcsTargPos[1..8] for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	mm or °	
Description of parameter values	Target position in the axis coordinate system	
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel (after block pre-run computation)	
Dependency	-	
Data type	8 x Double_t (8 x 8 Bytes)	

Current axis configuration in adjustment

Current axis configuration in adjustment		ActAxConf[1..8]
It provides the current axis configuration in adjustment after block pre-run. The system axis indices of the currently configured channel axes are stored in ascending order in ActAxConf. ActAxConf is intended for the initialization program via CPL access.		
Path	/SysChSRun[<Channel>]/ActAxConf[1..8] CPL: SD.SysChSRun[<Channel>].ActAxConf[1..8] - or - SD.SysChSRun.ActAxConf[1..8] for access in the individual channel.	
Value range	0	Channel axis not assigned
	1.. 250	System axis index
Pre-assignment	0	
Unit	-	
Description of parameter values	System axis in system axis index (1..250) of the channel axes	
Access rights	MTB: Read Setter: Read Operator: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel (after computation and during the block pre-run adjustment)	
Dependency	-	
Data type	8 x Int_t (8 x 4 bytes)	

Axis configuration in the block preparation

Axis configuration in the block preparation		PrepAxConf[1..8]
Provides the axis configuration of the block preparation in the adjustment after block pre-run. The system axis indices of the channel axes configured in the block preparation are stored in ascending order in PrepAxConf. PrepAxConf is intended for the initialization program via CPL access.		
Path	/SysChSRun[<Channel>]/PrepAxConf[1..8] CPL: SD.SysChSRun[<Channel>].PrepAxConf[1..8] - or - SD.SysChSRun.PrepAxConf[1..8] for access in the individual channel.	
Value range	0	Channel axis not assigned
	1.. 250	System axis index
Pre-assignment	0	
Unit	-	
Description of parameter values	System axis in system axis index (1..250) of the channel axes	
Access rights	MTB: Read Setter: Read Operator: Read NC prog.: Read	

	PLC prog.:	Read
Activation	-	
Update	By the NC kernel (after computation and during the block pre-run adjustment)	
Dependency	-	
Data type	8 x Int_t (8 x 4 bytes)	

Axis Configuration in the Target Block

Axis configuration in the target block of the block look-ahead		TargAxConf[1..8]
Provides the axis configuration of the target block during the adjustment after block pre-run.		
The channel axis names of the system axis indices configured in the target block are stored in ascending order in TargAxConf.		
TargAxConf is intended for the initialization program via CPL access.		
Path	/SysChSRun[<Channel>]/TargAxConf[1..8] CPL: SD.SysChSRun[<Channel>].TargAxConf[1..8] - or - SD.SysChSRun.TargAxConf[1..8] for accesses in the individual channel.	
Value range	0 Channel axis not assigned 1.. 250 System axis index	
Pre-assignment	0	
Unit	-	
Description of parameter values	System axis in system axis index (1..250) of the channel axes	
Access rights	MTB: Read Setter: Read Operator: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel (after computation and during the block pre-run adjustment)	
Dependency	-	
Data type	8 x Int_t (8 x 4 bytes)	

Technological axis adjustment

Technological axis adjustment		TechAxAdjust
This date can be used to determine the axes requiring a position adjustment within an adjustment after block pre-run.		
Bit-coded array to identify the axes whose position have to be adjusted.		
Path	/SysChSRun[<Channel>]/TechAxAdjust CPL: SD.SysChSRun[<Channel>]/TechAxAdjust - or - SD.SysChSRun.TechAxAdjust for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	Meaning of the bits: 2⁰ (= 1) Position of the main axis has to be adjusted.	

	2¹ (= 2)	Position of the secondary axis has to be adjusted.
	2² (= 4)	Position of the normal axis has to be adjusted.
	2³ (= 8)	Positions of all axes have to be adjusted.
	2⁴ (= 16)	Any axis selection: The positions of axes stored in the system data SysChSRun[<Channel>]/PosAdjust have to be adjusted (logic currently not implemented).
Access rights	MTB:	Read and write
	Setter:	Read and write
	Operator:	Read and write
	NC_Prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Computation active

Computation active		Active
Provides the channel-specific information that the computation of block pre-run is active in the channel.		
By querying in the program, NC or CPL parts (such as a tool change or modifications in the database) can be hidden during the computation.		
Path	/SysChSRun[<Channel>]/Active CPL: SD.SysChSRun[<Channel>]/Active - or - SD.SysChSRun.Active for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Block pre-run is not active.
	1	Block pre-run is active.
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Byte_t (1 byte)	

Computation mode

Computation mode	ActMode
Provides the channel-specific information which type of computation of block pre-run is active.	

System data

By querying in the program, NC or CPL parts (such as a program loop) can be hidden during the computation.

This date is mainly required for special instrumentations in the NC/CPL program code to hide it in case of an intended endless loop in the program if the block pre-run mode "2" was selected for example. Without instrumentation, a block pre-run with an endless loop in the CPL code of the NC program can run endlessly.

The block pre-run modes 0 and 1 result from the target block selection type selected by the user. Block pre-run mode 2 has to be supported by a suitable HMI transferring the special option to the CNC kernel.

Path	/SysChSRun[<Channel>]/ActMode CPL: SD.SysChSRun[<Channel>]/ActMode - or - SD.SysChSRun.ActMode for accesses in the individual channel.	
Value range	0-2	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Block pre-run up to target block defined by a file position
	1	Block pre-run up to the re-entry mark defined by an XML abort or RCB mark (callchain)
	2	Block pre-run up to the main program mode. Mainly used to check for a correct syntax, to simulate processing and to compute extensions of the NC program in the plane.
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel at the beginning of the block pre-run	
Dependency	-	
Data type	Byte_t (1 byte)	

Target block adjustment program active

Target block adjustment program active		TargAdjPrgActive
It provides channel-specific information whether the target block adjustment program is active in the block pre-run.		
By querying in the program, the general subroutines can for example be used as adjustment programs and their behavior can be adapted respectively in case of an adjustment.		
Path	/SysChSRun[<Channel>]/TargAdjPrgActive CPL: SD.SysChSRun[<Channel>]/TargAdjPrgActive - or - SD.SysChSRun.TargAdjPrgActive for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Target block adjustment program not active.

	1	Target block adjustment program active in block pre-run.
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Byte_t (1 byte)	

Init adjustment program (NameOfInitAdjPrg) Active

Init adjustment program (NameOfInitAdjPrg) active		InitAdjPrgActive
It provides channel-specific information whether the Init adjustment program is active in the block pre-run.		
By querying in the program, the general subroutines can for example be used as adjustment programs and their behavior can be adapted respectively in case of an adjustment.		
Path	/SysChSRun[<Channel>]/InitAdjPrgActive CPL: SD.SysChSRun[<Channel>]/InitAdjPrgActive - or - SD.SysChSRun.InitAdjPrgActive for access to the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Init adjustment program not active.
	1	Init adjustment program active in block pre-run.
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Byte_t (1 byte)	

Block pre-run active in look-ahead range after target block

Block pre-run active in look-ahead range after target block	LookAheadPreRun
For each channel, provides information about whether a block pre-run is active in the Look-Ahead range after the target block.	
The system date is used for queries in the part program. Program parts already completed in the block pre-run in the Look-Ahead range of the target block, are prepared again for the second time after the block pre-run. this can especially be undesired for CPL parts. Using the date, program parts such as write-access to tables and databases can be specifically hidden in the block pre-run.	

System data

Path	/SysChSRun[<Channel>]/LookAheadPreRun CPL: SD.SysChSRun[<Channel>]/LookAheadPreRun - or - SD.SysChSRun.LookAheadPreRun for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Block pre-run is not active in the Look-Ahead range after the target block.
	1	Block pre-run is active in the Look-Ahead area of the target block.
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Byte_t (1 byte)	

Channel is not adjusted

Channel is not adjusted		ChanNotAdjusted
Provides the channel-specific information that the channel has not been adjusted after the calculation.		
The channel is seen as adjusted when modal NC functions and auxiliary functions, axes, spindles, coordinates and tools are in their target state.		
Path	/SysChSRun[<Channel>]/ChanNotAdjusted CPL: SD.SysChSRun[<Channel>]/ChanNotAdjusted - or - SD.SysChSRun.ChanNotAdjusted for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Channel is adjusted.
	1	Channel is not adjusted.
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Byte_t (1 byte)	

Modal NC functions are not adjusted

Modal NC functions are not adjusted		NcfNotAdjusted
Provides the channel-specific information that the modal NC functions have not been adjusted after the calculation.		
To adjust individual modal NC functions, the NC command ADJ is for example available		
Path	/SysChSRun[<Channel>]/NcfNotAdjusted CPL: SD.SysChSRun[<Channel>]/NcfNotAdjusted - or - SD.SysChSRun.NcfNotAdjusted for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Modal NC functions are adjusted.
	1	Modal NC functions are not adjusted.
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Byte_t (1 byte)	

Modal auxiliary functions are not adjusted

Modal auxiliary functions are not adjusted		AuxNotAdjusted
Provides the channel-specific information that the modal auxiliary functions have not been adjusted after the calculation.		
Path	/SysChSRun[<Channel>]/AuxNotAdjusted CPL: SD.SysChSRun[<Channel>]/AuxNotAdjusted - or - SD.SysChSRun.AuxNotAdjusted for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Modal auxiliary functions are adjusted.
	1	Modal auxiliary functions are not adjusted.
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	

System data

Dependency -
Data type Byte_t (1 byte)

Axis configuration is not adjusted

Axis configuration is not adjusted		AxConfNotAdjusted
Provides the channel-specific information that the axis configuration has not been adjusted after the calculation.		
To adjust the axis configuration, the NC command TAX is available.		
Path	/SysChSRun[<Channel>]/AxConfNotAdjusted CPL: SD.SysChSRun[<Channel>]/AxConfNotAdjusted - or - SD.SysChSRun.AxConfNotAdjusted for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Axis configuration is adjusted.
	1	Axis configuration is not adjusted.
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Byte_t (1 byte)	

Axis transformation is not adjusted

Axis transformation is not adjusted		CoordNotAdjusted
Provides the channel-specific information that the axis transformation has not been adjusted after the calculation.		
To adjust the axis configuration, the NC command ADJ is available.		
Path	/SysChSRun[<Channel>]/CoordNotAdjusted CPL: SD.SysChSRun[<Channel>]/CoordNotAdjusted - or - SD.SysChSRun.CoordNotAdjusted for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Axis transformations are adjusted.
	1	Axis transformations are not adjusted.
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC_Prog.:	Read
	PLC prog.:	Read

Activation	-
Update	By the NC kernel
Dependency	-
Data type	Byte_t (1 byte)

Active adjustment state

Active adjustment state		ActStatus
Provides the state of the block active in the interpolator while adjusting the block pre-run.		
ActStatus is intended for the display in the adjustment screen. Current parameter value gaps can be filled with different states in future versions. It is always ensured that the current sequence is maintained and that the values of the valid status information up to now keep their validity.		
Path	/SysChSRun[<Channel>]/ActStatus CPL: SD.SysChSRun[<Channel>]/ActStatus - or - SD.SysChSRun.ActStatu to access in the individual channel.	
Value range	0.. 80	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Active block does not belong to adjustment
	10	Computation completed
	20	Active block belongs to init adjustment program
	30	Active block is a manual data input block in the adjustment
	40	Action block active
	60	Active block belongs to target adjustment program
	80	Approaching block active
Access rights	MTB:	Read
	Setter:	Read
	Operator:	Read
	NC prog.:	Read
	PLC prog:	Read
Activation	-	
Update	By the NC kernel (after block pre-run and during subsequent adjustment)	
Dependency	-	
Data type	Byte_t (1 byte)	

Target block code

Target block code	TargBlockSrc
NC code copy of the target block. It is written on the date after successful calculation.	
Only the first 100 characters are copied.	
Path	/SysChSRun[<Channel>]/TargBlockSrc CPL: SD.SysChSRun[<Channel>]/TargBlockSrc - or - SD.SysChSRun.TargBlockSrc for access in the individual channel.

System data

Value range	Characters from IsoLatin1String
Pre-assignment	<EmptyString>
Unit	-
Description of the variable values	Contains the first 100 characters of the target block.
Access rights	MTB: Read Setter: Read Operator: Read NC prog.: Read PLC prog: Read
Activation	-
Update	By the NC kernel when reaching the target block in the computation
Dependency	-
Data type	IsoLatin1String (100 bytes for text + 3 bytes + 1 byte for end identification)

Backed up target block available

Backed up target block available		TargetBlkAvailable
Provides the information that a backed up target block (reentry point) exists for a selected program.		
This information is only available if the respective bit ($2^0 = 1$) for checking the backed up RCB marks are set in the system date /SysSRun/TargetBlkCheck .		
Path	/SysChSRun[<Channel>]/TargetBlkAvailable CPL: SD.SysChSRun[<Channel>]/TargetBlkAvailable - or - SD.SysChSRun.TargetBlkAvailable for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0 No backed up target block available. 1 A backed up target block is available via an RCB mark.	
Access rights	MTB: Read Setter: Read Operator: Read NC prog.: Read PLC prog: Read	
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Int_t (4 bytes)	

Locking target blocks

Lock target blocks	LockTargBlock
Allows in the respective channel to lock areas for the target block selection in the part program.	
In the part program, ranges in which no target blocks are permissible can thus be locked. When the lock is activated, the target block is applied, but an NC start is not assumed. In this case, either a general warning or – if present	

– an individual note text (see **/SysSRun/LockWarning** system date) is output. The block pre-run cannot be continued on another target block.

The date is generally written in the NC program. The control resets it upon reset.

Path	/SysChSRun[<Channel>]/LockTargBlock CPL: SD.SysChSRun[<Channel>]/LockTargBlock - or - SD.SysChSRun.LockTargBlock for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of parameter values	0	Target blocks are not locked.
	1	Target blocks are disabled.
Access rights	MTB:	Read and write
	Setter:	Read and write
	Operator:	Read and write
	NC prog.:	Read and write
	PLC prog:	Read and write
Activation	Becomes active within the block preparation	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Storage directory of re-entry data

Storage directory of re-entry data		ReentryInfoDir
Directory that stores the re-entry data.		
For an easy access from NC programs or an HMI, the storage directory is stored in this system date. Together with the system date /SysSRun/ReEntryIndex , the latest saved abort information file can be addressed easily for example.		
The NC kernel writes on the date.		
Path	/SysChSRun[<Channel>]/ReentryInfoDir CPL: SD.SysChSRun[<Channel>]/ReentryInfoDir - or - SD.SysChSRun.ReentryInfoDir for accesses in the individual channel.	
Value range	Characters from Iso-Latin-1 string, length of 99 characters	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Directory specification	
Access rights	MTB:	Read and write
	Setter:	Read and write
	Operator:	Read and write
	NC prog.:	Read and write
	PLC prog:	Read and write
Activation	Written by the NC kernel upon the initial activation of the block pre-run.	
Update	-	

Dependency -

Data type IsoLatin1 string = length of 99 characters

32.2.19 Test mode "/SysChTestMode"

Enabling the test mode

Enabling the test mode		Enable
The kinematic limit within the block preparation only takes the test feed value in consideration if a signal is available. Only then it is possible that an increased test feed - when compared to the programmed feed - is traversed at runtime. Note: In case of an enabled test mode, an unfavorable reduction of the permissible acceleration and the permissible velocity can occur due to the kinematic limit for the normal mode. Thus, the test mode should only be enabled if the user wants to enable the test feed.		
Path	/SysChTestMode[<Channel>]/Enable CPL: SD.SysChTestMode[<Channel>]/Enable - or - SD.SysChTestMode.Enable for access in the individual channel.	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of parameter values	0 Kinematic limit ignores the test feed 1 Kinematic limit takes the test feed into consideration.	
Access rights	MTB: Read and write Setter: Read and write Operator: Read and write NC prog.: Read and write PLC prog: Read and write	
Activation	Becomes active within the block preparation	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Test feed

Test feed		TestFeedrate
Value used for test feed of the channel		
Path	/SysChTestMode[<Channel>]/TestFeedrate CPL: SD.SysChTestMode[<Channel>]/TestFeedrate- or - SD.SysChTestMode.TestFeedrate for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	mm/min	
Access rights	MTB: Read and write Setter: Read and write Operator: Read and write	

	NC prog.:	Read and write
	PLC prog:	Read and write
Activation	Becomes active within the block preparation	
Update	-	
Dependency	-	
Data type	Double_t (8 bytes)	

Test rapid traverse

Test rapid traverse		TestRapid
	Value used for test rapid traverse of the channel	
Path	/SysChTestMode[<Channel>]/TestRapid CPL: SD.SysChTestMode[<Channel>]/TestRapid - or - SD.SysChTestMode.TestRapid for access in the individual channel.	
Value range	-	
Pre-assignment	0	
Unit	mm/min	
Access rights	MTB: Read and write Setter: Read and write Operator: Read and write NC prog.: Read and write PLC prog: Read and write	
Activation	Becomes active within the block preparation	
Update	-	
Dependency	-	
Data type	Double_t (8 bytes)	

32.2.20 Waiting condition when waiting on bit events "/SysChWaitBitEv"

Waiting condition when waiting on bit events

Waiting condition when waiting on bit events		SysChWaitBitEv
	Indicates the waiting condition of a channel if this channel waits for a bit event. The waiting condition is stored as character string in the form of "(1 2 23)&(42 11)&17". A vertical dash " " stands for an OR operation and "&" for an AND operation. If the waiting condition is longer than the system date, it is stored as short form in the system date and three dots "..." are attached. The end of the character string is labeled by a zero byte. If the channel does not wait on bit events, there is an empty character string (first character is a zero byte) in this system date. This system date is available at delivery and stored as volatile.	
Path	/SysChWaitBitEv[<Channel>] CPL: SD.SysChWaitBitEv[<Channel>] or CPL: SD.SysChWaitBitEv in the current channel	
Value range	"()0123456789 &." as IsoLatin1String of 64 characters	
Pre-assignment	Empty character string	

System data

Unit	-
Description of the variable values	Waiting condition when waiting on bit events
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	Is always active.
Update	If a channel waits on bit events (->enter waiting condition) due to a WEV or WREV and fulfilled (-> delete waiting condition).
Dependency	-
Data type	IsoLatin1String (64 bytes for text)

32.2.21 Coordinate system display of the tool editor "/SysCoordSystem"

General information

The permanent system data defines the coordinate system in the tool editor.

For more information, refer to the "IndraMotion MTX Functional Description" documentation.

The structural elements described are defined in the "sdcrdsys.xsd" schema.

Structural element	Description
Value	Predefined settings
Horizontal	Direction of abscissa
Vertical	Direction of the ordinate
Calc	Type of coordinate system determination
X_axis_pref	X-axis as preferred axis
Y_axis_pref	Y-axis as preferred axis
Z_axis_pref	Z-axis as preferred axis

Tab. 32-14: Overview on the structure element SysCoordSystem

Predefined coordinate system settings of the tool editor

Predefined coordinate system settings of the tool editor		Value
Facilitates the selection of several preset coordinate system settings for the tool editor.		
Path	/SysCoordSystem/Value CPL: SD.SysCoordSystem.Value	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	See documentation "IndraMotion MTX Commissioning", chapter "Visualizing and configuring the tool catalog".	
Access rights	MTB: Read and write Setter: Read and write User: Read and write	

	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	A selected setting is only active if no automatic determination of the coordinate system has been specified in SD.SysCoordSystem.Calc No.	
Data type	Int_t (xs:int, 4 bytes)	

Direction of the abscissa in the tool editor coordinate system

Direction of abscissa in the tool editor coordinate system		horizontal
Specifies if the abscissa in the tool editor coordinate system points in positive or negative direction.		
Path	/SysCoordSystem/horizontal CPL: SD.SysCoordSystem.horizontal	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Abscissa in negative direction
	1	Abscissa in positive direction
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:byte	

Direction of the ordinate in the tool editor coordinate system

Direction of the ordinate in the tool editor coordinate system		vertical
Specifies if the ordinate in the tool editor coordinate system points in positive or negative direction.		
Path	/SysCoordSystem/vertical CPL: SD.SysCoordSystem.vertical	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Ordinate in negative direction
	1	Ordinate in positive direction
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write

System data

	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:byte	

Type of coordinate system determination in the tool editor

Type of coordinate system determination in the tool editor		Calc
Specifies if the tool editor coordinate system is to be determined automatically.		
Path	/SysCoordSystem/Calc CPL: SD.SysCoordSystem.Calc	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No automatic determination of the coordinate system
	1	Automatic determination of the coordinate system
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:byte	

Preferred axis in the tool editor

X-axis as preferred axis in the tool editor		Z_axis_pref
Y-axis as preferred axis in the tool editor		Z_axis_pref
Z-axis as preferred axis in the tool editor		Z_axis_pref
Specifies if the respective axis is the preferred axis of the coordinate system display of milling tools.		
Path	/SysCoordSystem/X_axis_pref	
	/SysCoordSystem/Y_axis_pref	
	/SysCoordSystem/Z_axis_pref	
	CPL: SD.SysCoordSystem.X_axis_pref	
	CPL: SD.SysCoordSystem.Y_axis_pref	
	CPL: SD.SysCoordSystem.Z_axis_pref	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	The axis is not the preferred axis
	1	The respective axis is the preferred axis

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:byte	

32.2.22 System control "/SysCtrl"

Setting time for the fan change

Set time for fan change	SetFanTime

Reset the fan runtime after replacing fan.



This value should be set only after the CPU fan has been replaced in the course of preventive maintenance measures!

Path	/SysCtrl/SetFanTime CPL: SD.SysCtrl.SetFanTime	
Value range	0	FALSE
	1	true
Pre-assignment	0	FALSE
Unit	-	
Description of the variable values	0	Output value is FALSE: Controller acknowledges the successful adoption of the reset date after a fan change by setting the datum to FALSE.
	1	Output value is TRUE: Set the fan runtime time to zero after replacing fan.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read
	PLC prog.:	Read
Activation	After writing of TRUE	
Update	After successful reset of the fan runtime by the control.	
Dependency	-	
Data type	Boolean_t (1 byte)	

32.2.23 Control data for handwheel mode in WCS "/SysHandwWCS"

General information

The system data structure SysHandwWCS is created for each channel in the volatile memory.

It contains control data for the handwheel mode in the WCS.

The following structural elements are defined in the "sdjog.xsd" schema:

Structural element	Description
PathMode	Activate contour handwheel
ContourType	Contour type
Angle	Straight line angle
MainCP	Position of the circle center point, main axis
ScndCP	Position of the circle center point, secondary axis

Tab. 32-15: Overview on structural elements "SysHandwWCS"

Activating the contour handwheel

Activate contour handwheel	PathMode

Allows the activation and deactivation of the contour handwheel.

By setting "PathMode", the coordinates of main and secondary axis of the active plane are determined.

Path /SysHandwWCS[<Channel>]/PathMode

CPL: SD.SysHandwWCS[<Channel>].PathMode - or -

SD.SysHandwWCS.PathMode in the individual channel

Value range -

Pre-assignment 0

Unit -

Description of the variable values 0 Deactivate contour handwheel

1 Activate contour handwheel;

The type of movement results from the content of the element "ContourType".

Access rights **MTB:** Read and write

Setter: Read and write

User: Read and write

NC prog.: Read and write

PLC prog.: Read and write

Activation -

Update -

Dependency -

Data type UnsignedByte_t (1 Byte)

Contour type

Contour type	ContourType

Indicates the contour type for the contour handwheel.

Path /SysHandwWCS[<Channel>]/ContourType

CPL: SD.SysHandwWCS[<Channel>].ContourType - or -

SD.SysHandwWCS.ContourType in the individual channel

Value range -

Pre-assignment 0

Unit

Description of the variable values 0 Straight line in the current plane:

		The positive direction of the straight line in the current plane is determined by the angle parameter "Angle".
	1	Arc in the current plane: The center point position of the arc is specified via the elements "MainCP" and "ScndCP".
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	PathMode = 1	
Data type	UnsignedByte_t (1 Byte)	

Straight line angle

Straight line angle		Angle
	If "ContourType" = 0, "Angle" specifies the angle between the positive main axis direction and the straight line.	
Path	/SysHandwWCS[<Channel>]/Angle CPL: SD.SysHandwWCS[<Channel>].Angle - or - SD.SysHandwWCS.Angle in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	Degree	
Description of the variable values	Angle between the positive main axis direction and the straight line	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	ContourType = 0	
Data type	Float_t (4 bytes)	

Circle center point position of the main axis

Position of the circle center point, main axis		MainCP
	If "ContourType" = 1, "MainCP" specifies the position of the circle center point in the main axis.	
Path	/SysHandwWCS[<Channel>]/MainCP CPL: SD.SysHandwWCS[<Channel>].MainCP - or - SD.SysHandwWCS.MainCP in the individual channel	
Value range	-	

System data

Pre-assignment	0
Unit	mm
Description of the variable values	Position of the circle center point in the main axis
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	-
Update	-
Dependency	ContourType = 1
Data type	Float_t (4 bytes)

Center point position of the secondary axis

Position of the circle center point, secondary axis		ScndCP
If "ContourType" = 1, "ScndCP" specifies the position of the circle center point in the secondary axis.		
Path	/SysHandwWCS[<Channel>]/ScndCP CPL: SD.SysHandwWCS[<Channel>].ScndCP - or - SD.SysHandwWCS.ScndCP in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	mm	
Description of the variable values		
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	-	
Update	-	
Dependency	ContourType = 1	
Data type	Float_t (4 bytes)	

32.2.24 System hardware status information "/SysHwStates"

Processor overtemperature warning

Processor overtemperature warning	OverTemp
Output of a warning upon "processor overtemperature".	
If the output value is TRUE (1), there is a danger of the processor overheating. This might result in an uncontrolled incorrect behavior of the controller.	
 The machine has to be shut down as soon as possible in order to avoid damage.	

Remedy:

Mounting or replacing a fan at the intended connection at the control component group

If the fan is okay and the warning is still active, the ambient temperature must be reduced (observe the CPU temperature!).

This value should not be overwritten; it is updated by the controller only if the status is changed.

Path	/SysHwStates/OverTemp CPL: SD.SysHwStates.OverTemp	
Value range	0	FALSE
	1	TRUE
Pre-assignment	0	FALSE
Unit	-	
Description of the variable values	0	Output value is FALSE, i.e. the temperature of the processor is within normal range.
	1	Output value is TRUE, i.e. overtemperature in the processor.
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel when the overtemperature warning is activated and deactivated.	
Dependency	-	
Data type	Boolean_t (1 byte)	

Battery warning

Battery warning	BattFail

Warning of battery undervoltage.

The battery is responsible for retaining the data of the ROOT file system, the permanent CPL and PLC variables, and the operation of the clock while the component is without voltage. If the battery monitor reports "Undervoltage", then data integrity is guaranteed as a rule for about another 4 weeks.



If there is a battery undervoltage, there is a risk of **data loss** if the controller is turned off without first replacing the battery.

This value should not be overwritten; it is updated by the controller only if the status is changed.

Path	/SysHwStates/BattFail CPL: SD.SysHwStates.BattFail	
Value range	0	FALSE
	1	TRUE
Pre-assignment	0	FALSE
Unit	-	
Description of the variable values	0	Output value is FALSE, i.e. the temperature of the battery is within normal range.

System data

	1	Output value is TRUE, i.e. battery undervoltage exists.
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel when the battery warning is activated and deactivated.	
Dependency	-	
Data type	Boolean_t (1 byte)	

32.2.25 System interpolations axis positions"/SysIpoAcsPos"

General information

The system data structure SysIpoAcsPos is created for each channel in the volatile memory.

It contains the real-time coordinates of the interpolator in the axis coordinate system (ACS).

The optional system date is defined in the "SDDefacscrd.xml_" file.

The structure is defined in the "sdrtpos.xsd_" scheme.

Structural element	Description
Active	Activation/deactivation
Coord[1..8]	Axis coordinates

Tab. 32-16: Overview on structural elements SysIpoAcsPos[<Channel>]



For more information, refer to the documentation "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P) in the chapter "Real-time coordinates".

Enabling the axis coordinate storage

Enabling the axis coordinate storage	Active

Enables and disables the ACS real-time coordinate storage for the affected channel.

Path /SysIpoAcsPos[<Channel>]/Active

CPL: SD.SysIpoAcsPos[<Channel>].Active

Value range 0 Deactivating

1 Activating

2 Enabling with permanent modulo computation for rotary axes (SysAxType=2) with modulo computation (EnablModCalc=1)

Pre-assignment 0 Axis coordinate storage inactive

Unit -

Description of the variable values 0 Disabled the axis coordinate storage.

1 Enabled the axis coordinate storage.

Access rights MTB: Read and write

	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		Immediately
Update		-
Dependency		-
Data type		Int_t (4 bytes)

ACS real-time coordinates

ACS real-time coordinates	Coord

Returns ACS coordinates in the interpolation cycle.

In SyslpoWcsPos[<Channel>]/Coord[1..8], up to eight channel coordinates of the axis coordinate system (ACS) are stored in an interpolation cycle. The coordinates are stored in an ascending channel order. The order is subject to change in case of an axis replacement. Unavailable axes return the value zero. The update is switched on/off per channel depending on SyslpoAcsPos[<Channel>]/Active.

Path	/SyslpoAcsPos[<Channel>]/Coord[1..8] CPL: SD.SyslpoAcsPos[<Channel>].Coord[1..8]
Value range	-
Pre-assignment	0
Unit	0.1 µm
Description of the variable values	Current axis coordinates of the channel
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	SyslpoAcsPos[<Channel>]/Active=1 or 2
Update	By NC kernel in an interpolation cycle
Dependency	See SyslpoAcsPos[<Channel>]/Active
Data type	Int_t (4 bytes)

32.2.26 System interpolation basic workpiece positions "/SyslpoBcsPos"

General information

The system data structure SyslpoBcsPos is created for each channel in the volatile memory.

It contains the real-time coordinates of the interpolator in the basic coordinate system (BCS).

The optional system date is defined in the "SDDefbcscrd.xml_" file.

The structure is defined in the "sdrtpos.xsd_" scheme.

Structural element	Description
Active	Activation/deactivation
Coord[1..8]	Basic coordinates

Tab. 32-17: Overview on structural elements SysIpoBcsPos[<Channel>]



For more information, refer to the documentation "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P) in the chapter "Real-time coordinates".

Enabling the basic workpiece positions

Enabling the basic workpiece positions		Active
Enables and disables the BCS real-time coordinate storage for the affected channel.		
Path	/SysIpoBcsPos[<Channel>]/Active CPL: SD.SysIpoBcsPos[<Channel>].Active	
Value range	0	Deactivating
	1	Activating
Pre-assignment	0	Basic workpiece coordinate storage inactive
Unit	-	
Description of the variable values	0	Disables the basic workpiece storage.
	1	Enables the basic workpiece storage.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Immediately	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

BCS real-time coordinates

BCS real-time coordinates		Coord
Returns BCS coordinates in the interpolation cycle.		
In SysIpoWcsPos[<Channel>]/Coord[1..8], up to eight channel coordinates of the basic workpiece coordinate system (BCS) are stored in an interpolation cycle. The coordinates are stored in an ascending channel order. The order is subject to change in case of an axis replacement. Unavailable coordinates return the value zero. The update is switched on/off per channel depending on SysIpoBcsPos[<Channel>]/Active.		
Path	/SysIpoBcsPos[<Channel>]/Coord[1..8] CPL: SD.SysIpoBcsPos[<Channel>].Coord[1..8]	
Value range	-	
Pre-assignment	0	

Unit	0.1 µm
Description of the variable values	Current basic workpiece coordinates of the channel
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	SyslpoBcsPos[<Channel>]/Active=1
Update	By NC kernel in an interpolation cycle
Dependency	See SyslpoBcsPos[<Channel>]/Active
Data type	Int_t (4 bytes)

32.2.27 System interpolations machine positions"/SyslpoMcsPos"

General information

The system data structure SyslpoMcsPos is created for each channel in the volatile memory.

It contains the real-time coordinates of the interpolator in the machine coordinate system (MCS).

The optional system date is defined in the "SDDefmcsrd.xml_" file.

The structure is defined in the "sdrtpos.xsd_" scheme.

Structural element	Description
Active	Activation/deactivation
Coord[1..8]	Machine coordinates

Tab. 32-18: Overview on structural elements SyslpoMcsPos[<Channel>]



For more information, refer to the documentation "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P) in the chapter "Real-time coordinates".

Enabling the machine coordinate storage

Enabling the machine coordinate storage		Active
Enables and disables the MCS real-time coordinate storage for the affected channel.		
Path	/SyslpoMcsPos[<Channel>]/Active CPL: SD.SyslpoMcsPos[<Channel>].Active	
Value range	0	Deactivating
	1	Activating
Pre-assignment	0	Machine coordinate storage inactive
Unit	-	
Description of the variable values	0	Disables the machine coordinate storage.
	1	Disables the machine coordinate storage.
Access rights	MTB: Read and write Setter: Read and write	

	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		Immediately
Update		-
Dependency		-
Data type		Int_t (4 bytes)

MCS real-time coordinates

MCS real-time coordinates	Coord

Returns MCS coordinates in the interpolation cycle.

In SysIpoWcsPos[<Channel>]/Coord[1..8], up to eight channel coordinates of the machine coordinate system (MCS) are stored in an interpolation cycle. The coordinates are stored in an ascending channel order. The order is subject to change in case of an axis replacement. Unavailable coordinates return the value zero. The update is switched on/off per channel depending on SysIpoMcsPos[<Channel>]/Active.

Path	/SysIpoMcsPos[<Channel>]/Coord[1..8] CPL: SD.SysIpoMcsPos[<Channel>].Coord[1..8]
Value range	-
Pre-assignment	0
Unit	0.1 µm
Description of the variable values	Current machine coordinates of the channel
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	SysIpoMcsPos[<Channel>]/Active=1
Update	By NC kernel in an interpolation cycle
Dependency	See SysIpoMcsPos[<Channel>]/Active
Data type	Int_t (4 bytes)

32.2.28 System interpolation path velocities"/SysIpoPathFeed"

General information

The system data structure SysIpoPathFeed can be created for each channel in the volatile memory.

It contains the real-time velocity data of the path interpolator.

The optional system date is defined in the "SDDefpathfeed.xml_" file.

The structural element described is defined in the "sdrtpos.xsd_" schema.

Structural element	Description
Active	Activation/deactivation
ProgFeed	Programmed path velocity
ActFeed	Current path feed

Tab. 32-19: Overview on structural elements SyslpoPathFeed



For more information, refer to the documentation "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P) in the chapter "Real-time path velocities".

Activating the path velocity

Activating the path velocity		Active
Enables and disables the ACS real-time coordinate storage for the affected channel.		
Path	/SyslpoPathFeed[<Channel>]/Active CPL: SD.SyslpoPathFeed[<Channel>].Active	
Value range	0	Deactivating
	1	Activating
Pre-assignment	0	Realtime path velocity storage
Unit	-	
Description of the variable values	0	Disables the real-time path velocity storage.
	1	Enables the time synchronization storage.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Immediately	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Programmed path velocity

Programmed path velocity		ProgFeed
Returns the programmed path feed in an interpolation cycle.		
The programmed feed without potentiometer evaluation is updated with the block change in the interpolator in SyslpoPathFeed[<Channel>]/ProgFeed. For G94 and G93, data is stored in an 0.1 µm/interpolation cycle. The programmed F-value is stored in G95 - Feed/rotation. The update is switched on/off per channel depending on SyslpoAcsPos[<Channel>]/Active. The last value is retained in case of deactivation.		
Path	/SyslpoPathFeed[<Channel>]/ProgFeed CPL: SD.SyslpoPathFeed[<Channel>].ProgFeed	
Value range	-	

System data

Pre-assignment	0
Unit	0.1 µm/interpolation cycle (G94,G93) mm/U, inch/U (G95/G71, G95/G70)
Description of the variable values	Programmed path velocity
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	SysIpoPathFeed[<Channel>]/Active=1
Update	By NC kernel with block change in the interpolator
Dependency	See SysIpoPathFeed[<Channel>]/Active
Data type	Float_t (4 bytes)

Current path velocity

Current path velocity		ActFeed
Returns the current path feed in an interpolation cycle.		
The current path feed is updated in an interpolation cycle in SysIpoPathFeed[<Channel>]/ActFeed. All limitations (potentiometer, kinematics, etc.) are taken into consideration in this value. The update is switched on/off per channel depending on SysIpoAcsPos[<Channel>]/Active. The last value is retained in case of deactivation.		
Path	/SysIpoPathFeed[<Channel>]/ActFeed CPL: SD.SysIpoPathFeed[<Channel>].ActFeed	
Value range	-	
Pre-assignment	0	
Unit	0.1 µm/interpolation cycle	
Description of the variable values	Current path velocity	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	SysIpoPathFeed[<Channel>]/Active=1	
Update	By the NC kernel at interpolation cycle	
Dependency	See SysIpoPathFeed[<Channel>]/Active	
Data type	Float_t (4 bytes)	

32.2.29 System interpolations workpiece positions"/SysIpoWcsPos"

General information

The system data structure SysIpoWcsPos can be created for each channel in the volatile memory.

It contains the real-time coordinates of the interpolator in the workpiece coordinate system (WCS).

The optional system data is defined in the "SDDefwscrd.xml_" file.

The structure is defined in the "sdrtpos.xsd_" scheme.

Structural element	Description
Active	Activation/deactivation
Coord[1..8]	Workpiece coordinates

Tab. 32-20: Overview on structural elements SysIpoWcsPos[<Channel>]



For more information, refer to the documentation "MTX 15VRS Functional Description - Special Functions" (DOK-MTX***-NC*F*SP*V15-RERS-EN-P) in the chapter "Real-time coordinates".

Enabling the workpiece coordinate storage

Enabling the workpiece coordinate storage		Active
Enables and disables the WCS real-time coordinate storage for the affected channel.		
Path	/SysIpoWcsPos[<Channel>]/Active CPL: SD.SysIpoWcsPos[<Channel>].Active	
Value range	0	Deactivating
	1	Activating
Pre-assignment	0	Workpiece coordinate storage inactive
Unit	-	
Description of the variable values	0	Disables the workpiece coordinate storage.
	1	Enables the workpiece coordinate storage.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Immediately	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

WCS real-time coordinates

WCS real-time coordinates		Coord
Returns WCS coordinates in the interpolation cycle.		
In SysIpoWcsPos[<Channel>]/Coord[1..8], up to eight channel coordinates of the workpiece coordinate system (MCS) are stored in an interpolation cycle. The coordinates are stored in an ascending channel order. The order is subject to change in case of an axis replacement. Unavailable coordinates return the value zero. The update is switched on/off per channel depending on SysIpoWcsPos[<Channel>]/Active.		
Path	/SysIpoWcsPos[<Channel>]/Coord[1..8] CPL: SD.SysIpoWcsPos[<Channel>].Coord[1..8]	
Value range	-	

System data

Pre-assignment	0
Unit	0.1 µm
Description of the variable values	Current workpiece coordinates of the channel
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	SysIpoWcsPos[<Channel>]/Active=1
Update	By NC kernel in an interpolation cycle
Dependency	See SysIpoWcsPos[<Channel>]/Active
Data type	Int_t (4 bytes)

32.2.30 NC kernel diagnostics "/SysKernelDgn"

General information

The system data structure "SysKernelDgn" is created in the volatile memory. It is used to specifically diagnose block cycle time problems.

The following structural elements are defined in the "sddiagn.xsd" schema:

Structural element	Description
BlkCycleTimeWrng	Runtime warning for block cycle time problems
ActBlkSwitchWrng	Runtime warning for blocked current block switching
IpoJlaWhileWrng	Runtime warning for uncompleted action at the current block switching
BlkCycleTimeCount[1..60]	Counter for block cycle time warnings

Tab. 32-21: Overview on structural elements "SysKernelDgn"

Runtime warning for block cycle time problems

Runtime warning for block cycle time problems		BlkCycleTimeWrng
Allows generating the runtime warning 3968 for block cycle time problems (velocity reduction, interpolator expecting prepared block but none available). After setting the system data, a system reset to activate the logic is required to generate the runtime warning.		
Path	/SysKernelDgn/BlkCycleTimeWrng CPL: SD.SysKernelDgn.BlkCycleTimeWrng	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0 Runtime warning deactivated. 1 Runtime warning activated.	
Access rights	MTB: Read and write Setter: Read and write User: Read (read-only) NC prog.: Read and write	

	PLC prog.:	Read and write
Activation	System reset	
Update	-	
Dependency	-	
Data type	Byte_t (1 byte)	

Runtime warning for blocked current block switching

Runtime warning for blocked current block switching		ActBlkSwitchWrng
Allows generating the runtime warning 3989 for block cycle time problems with effect on the current block switching (axis standstill, interpolator expects prepared block, but the current block cannot be switched further).		
After setting the system data, a system reset to activate the logic is required to generate the runtime warning.		
Path	/SysKernelDgn/ActBlkSwitchWrng CPL: SD.SysKernelDgn.ActBlkSwitchWrng	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Runtime warning deactivated.
	1	Runtime warning activated.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read (read-only)
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	System reset	
Dependency	-	
Data type	Byte_t (1 byte)	

Runtime warning for uncompleted actions at the current block switching

Runtime warning for uncompleted action at the current block switching		IpoJlaWhileWrng
Allows generating the runtime warning 3969 if not all programmed action have been completed (such as acknowledgement of help functions) during the current block switching.		
After setting the system data, a system reset to activate the logic is required to generate the runtime warning.		
Path	/SysKernelDgn/IpoJlaWhileWrng CPL: SD.SysKernelDgn/IpoJlaWhileWrng	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Runtime warning deactivated.
	1	Runtime warning activated.

System data

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read (read-only)
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	System reset	
Update	-	
Dependency	-	
Data type	Byte_t (1 byte)	

Counter for block cycle time warnings

Counter for block cycle time warnings		BlkCycleTimeCount
It channel-specifically counts the block switching failed due to block cycle time problems.		
Indicates how often the runtime warning on the block cycle time problem was output (BlkCycleTimeWrng, warning number 3968). The counter is automatically nulled each time the channel is reset.		
Path	/SysKernelDgn/BlkCycleTimeCount[<Channel>] CPL: SD.SysKernelDgn.BlkCycleTimeCount[<Channel>]	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Number of output runtime warnings	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read (read-only)
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	Block by block by the NC kernel for each block cycle time warning. Reset in case of each channel reset	
Dependency	-	
Data type	UnsignedInt_t (4 bytes)	

32.2.31 Maximum number of NC kernel-controlled drives "/SysMaxAxNbr"

Maximum number of NC kernel-controlled drives

Maximum number of NC kernel-controlled drives	SysMaxAxNbr

This volatile data element parameter is not supported (anymore).

32.2.32 Number of edges "/SysMaxEd" displayed

Maximum tool edge number

Maximum tool edge number	MaxEd

This volatile system data element allows read-only access on the number of edges of the active tool displayed by the user interface. This number applies to all the channels.

SD.SysMaxEd.MaxEd is initialized to the value specified in the <SysMaxEd> entry in the "SDDat.xml" file at startup. The value is 2 by default, but can be modified to any value between 1 to 16 by the user. A startup is required to apply such a modification. To display all edges for each tool, this value should be equal to the maximum number of edges.

The structure element described is defined in the "sdtlcorr.xsd" schema.

Path	/SysMaxEdMaxEd CPL: SD.SysMaxEd.MaxEd	
Value range	-	
Pre-assignment	2	
Unit	-	
Description of the variable values	Maximum edge number for which tool correction data exists in the system data (currently always 16).	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Int_t (4 bytes)	

32.2.33 Control data for NC diagnostics "/SysNcDgnCtrl"

General information

The system data architecture SysNcDgnCtrl is stored in the permanent memory and is used channel and axis-independently.

The following structural elements are defined in the "sdncdgn.xsd" schema:

Structural element		Description
AxesLoadDgn	/...	Control data for the axis load diagnostics
	VelLoadDgn/DgnMode	Mode of the velocity axis load diagnostics
	VelLoadDgn/PeakTm-pHoldTime	Dwell time for temporary velocity peak values
	AccLoadDgn/DgnMode	Mode of the acceleration-axis load diagnostics
	AccLoadDgn/PeakTm-pHoldTime	Dwell time for temporary acceleration peak values

Tab. 32-22: Overview on the "SysNcDgnCtrl" structural elements

Mode of the velocity axis load diagnostics

Mode of the velocity axis load diagnostics		VelLoadDgn/ DgnMode
Enables/disables the velocity axis load diagnostics for all axes and resets the maximum velocity peak values. The NC command MCODES(145,...) can be used to read the temporarily and maximum occurring velocity peak values and the nominal value (100% value) for all axes of a channel. If the axis participates in the system axis coupling as slave axis, the master value velocities are used. Otherwise, the interpolator command velocities are used.		
 Switch off again the velocity load diagnostics (VelLoadDgn/DgnMode = 0) if this diagnostics is not required anymore! This impedes unnecessary loads of the interpolator interrupt.		
Path	/SysNcDgnCtrl/AxesLoadDgn/VelLoadDgn/DgnMode CPL: SD.SysNcDgnCtrl.AxesLoadDgn.VelLoadDgn.DgnMode	
Value range	-	
Pre-assignment	0 (switched off)	
Unit	-	
Description of the variable values	0	Velocity axis load diagnostics switched off / switch-off.
	1	Switch on velocity axis load diagnostics and reset the maximum velocity peak values for all axes. It is acknowledged with the value 2.
	2	Velocity axis load diagnostics switched on / switch-on.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	In the next interpolator Interrupt	
Update	In the next interpolator Interrupt	
Dependency	-	
Data type	Short_t (4 bytes)	

Dwell time for temporary velocity peak values

Dwell time for temporary velocity peak values	VelLoadDgn/ PeakTmpHoldTime
Defines the dwell time for temporary velocity peak values of the velocity axis load diagnostics for all axes. If the current velocity value is smaller than the current temporary peak value, the current velocity value is applied as peak value after the dwell time had exceeded. If the current velocity value is higher than the current temporary value, the current velocity value is immediately applied as temporary peak value.	
Path	/SysNcDgnCtrl/AxesLoadDgn/VelLoadDgn/PeakTmpHoldTime

	CPL: SD.SysNcDgnCtrl.AxesLoadDgn.VelLoadDgn.PeakTmpHoldTime
Value range	-
Pre-assignment	200
Unit	ms
Description of the variable values	Dwell time
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	In the next interpolator Interrupt
Update	-
Dependency	-
Data type	Short_t (4 bytes)

Mode of the acceleration axis load diagnostics

Mode of the acceleration-axis load diagnostics		AccLoadDgn/ DgnMode
Enables/disables the acceleration axis load diagnostics for all axes and re-sets the maximum acceleration peak values. The NC command MCODS(146,...) can be used to read the temporarily and maximum occurring acceleration peak values and the nominal value (100% value) for all axes of a channel. If the axis participates in the system axis coupling as slave axis, the master value accelerations are used. Otherwise, the interpolator command accelerations are used.		
 Switch off again the acceleration load diagnostics (AccLoadDgn/DgnMode = 0) if this diagnostics is not required anymore! This impedes unnecessary loads of the interpolator interrupt.		
Path	/SysNcDgnCtrl/AxesLoadDgn/AccLoadDgn/DgnMode CPL: SD.SysNcDgnCtrl.AxesLoadDgn.AccLoadDgn.DgnMode	
Value range	-	
Pre-assignment	0 (switched off)	
Unit	-	
Description of the variable values	0 Acceleration axis load diagnostics switched off / switch-off. 1 Switch on acceleration axis load diagnostics and reset the maximum acceleration peak values for all axes. It is acknowledged with the value 2. 2 Acceleration axis load diagnostics switched on / switch-on.	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	

System data

Activation	In the next interpolator Interrupt
Update	In the next interpolator Interrupt
Dependency	-
Data type	Short_t (4 bytes)

Dwell time for temporary acceleration peak values

Dwell time for temporary acceleration peak values	AccLoadDgn/ PeakTmpHoldTime

Defines the dwell time for temporary acceleration peak values of the acceleration axis load diagnostics for all axes.

If the current acceleration value is smaller than the current temporary peak value, the current acceleration value is applied as peak value after the dwell time had exceeded. If the current acceleration value is higher than the current temporary value, the current acceleration value is immediately applied as temporary peak value.

Path /SysNcDgnCtrl/AxesLoadDgn/AccLoadDgn/PeakTmpHoldTime
CPL: SD.SysNcDgnCtrl.AxesLoadDgn.AccLoadDgn.PeakTmpHoldTime

Value range -

Pre-assignment 200

Unit ms

Description of the variable values Dwell time

Access rights

MTB:	Read and write
Setter:	Read and write
User:	Read and write
NC prog.:	Read and write
PLC prog.:	Read and write

Activation In the next interpolator Interrupt

Update -

Dependency -

Data type Short_t (4 bytes)

32.2.34 Data for axis-parallel measurement**Commissioning data of the probe"/SysPrBase"**

The system data structure SysPrBase is stored in the permanent memory and contains the commissioning data for two probes.

The following structural elements are defined in the "sdprbase.xsd" schema:

Structural element		Description
PROBE[1..2]	/...	Commissioning data for two probes
	DRVINP	Travel measuring blocks with G75.1 or G75.2
	TOOL	Use active data or data of the substructure GEO

Structural element		Description
	LREFTT	Reference probe length to tool tip or center of ball
	REMOVE	Retracting type
	SAVFEED	Positioning velocity
	ANGLE	Orientation angle of spindle
	GEO/...	Installed tool probe in working space
	GEO/L[1..3]	Probe lengths
	GEO/R	Probe ball radius
	GEO/BO[1..3]	Coordinates in working space
	LOG/...	Data for PLC communication and auxiliary functions
	LOG/AUXON	M-functions to enable the probe
	LOG/AUXOFF	M-functions to disable the probe
	LOG/RETRIG	Re-trigger the probe
	LOG/CUSBIT	Customer bit for communication between NC kernel and PLC

Tab. 32-23: Overview on structural elements SysPrBase

The data of the system structure SysPrBase are exclusively used by the IndraWorks Operation operating interface and by the measuring function. Direct user access is not provided. Thus, the individual structural elements are not described in detail.



For more information about the SysPrBase system data structure, refer to the "IndraMotion MTX Measuring Functions" documentation.

Parameters for manual probing"/SysPrMan"

The system data structure SysPrBase is stored in the permanent memory and contains the commissioning data for two probes.

The following structural elements are defined in the "sdprman.xsd" schema:

Structural element		Description
	FUNCNO	
	FUNCRUN	
	RESET	
INIT	/...	Basic settings
	CHANNO	Channel number
	PROBENO	Probe number

System data

Structural element		Description
	FEED	Calibration and measuring feed
	SPINID	Spindle identifier
	AUXSPIORI	Auxiliary function for spindle orientation
	ROTAX[1..3]/...	Rotary axis settings
	ROTAX[1..3]/ID	Rotary axis ID
	ROTAX[1..3]/DIRCHG	Direction of rotation
	ROTAX[1..3]/AUXCLAMP	Auxiliary function for clamping
	ROTAX[1..3]/AUXCLAMP	Auxiliary function for unclamping
	LINAX[1..3]/...	Linear axis settings
	LINAX[1..3]/BASEPOINT-POS	Positive calibration position on the machine
	LINAX[1..3]/BASEPOINT-NEG	Negative calibration position on the machine
ZOT	/...	Zero offset
	TABLEPATH	Path and name of the zero point table
	PAGE	Zero point offset block
	ZO	Zero point offset value
CAL	/...	Calibration settings
	POINT/...	Calibrate point
	POINT/AXIS[1..3]/ENABLE	Enable axis for calibration
	POINT/AXIS[1..3]/DIR	Calibration direction
	POINT/BLOCKLEN	End measure length
	POINT/BLOCKSHIFT	Offset
	POINT/OFFSET	Activate zero offset
	RING/...	Calibrate in ring
	RING/PLANE	Interpolation level
	RING/RINGDIA	Calibration ring diameter
MEAS	/...	Measurement settings
	POINTCOUNT	Measuring point counter

Structural element		Description
	POINTS[1..4]/...	Measuring points
	POINTS[1..4]/MAINAXIS	Main axis
	POINTS[1..4]/AUXAXIS	Secondary axis
	ROTAX/...	Probe zero point rotary axis
	ROTAX/AXIS[1..3]/ ENABLE	Enable axis for measuring
	ROTAX/AXIS[1..3]/DIR	Measuring direction
	ROTAX/NO	Selected rotary axis
	ROTAX/REVAXANG	Reference axis
	ROTAX/MOVE	Traverse rotary axis
	ROTAX/FEED	Measuring feed
	ROTAX/G1	Positioning type
	ROTAX/ANGLE	Set angle
	ROTAX/RESULT	Measuring result of the angle
	CIRSTU/...	Probe zero point of circular spigot
	CIRSTU/PLANE	Interpolation level
	CIRSTU/DIA	Spigot diameter
	CIRSTU/MAINAXSHIFT	Main axis offset
	CIRSTU/AUXAXSHIFT	Auxiliary axis offset
	CIRSTU/INFDEPTH	Feed increment till measuring height
	CIRSTU/SAVDIS	Safety distance
	CIRSTU/RESULT[1..3]	Measuring result in the axis descriptions X, Y, Z
	RECSTU/...	Probe zero point of rectangular spigot
	RECSTU/PLANE	Interpolation level
	RECSTU/LENGTH	Length
	RECSTU/WIDTH	Width
	RECSTU/MAINAXSHIFT	Main axis offset
	RECSTU/AUXAXSHIFT	Auxiliary axis offset
	RECSTU/INFDEPTH	Feed increment till measuring height
	RECSTU/SAVDIS	Safety distance
	RECSTU/RESULT[1..3]	Measuring result in the axis descriptions X, Y, Z

System data

Structural element		Description
	RECPOCK/...	Probe ZP (rectangular) pocket
	RECPOCK/PLANE	Interpolation level
	RECPOCK/LIFT	Obstacle in pocket
	RECPOCK/MAINAX-SHIFT	Main axis offset
	RECPOCK/AUXAXSHIFT	Auxiliary axis offset
	RECPOCK/RESULT[1..3]	Measuring result in the axis descriptions X, Y, Z
	NOSE/...	Probe zero point of land
	NOSE/PLANE	Interpolation level
	NOSE/WIDTH	Land width
	NOSE/MOVEAX	Measuring axis
	NOSE/SHIFTAX	Offset
	NOSE/INFDEPTH	Feed increment till measuring height
	NOSE/SAVDIS	Safety distance
	NOSE/RESULT[1..3]	Measuring result in the axis descriptions X, Y, Z
	GROOVE/...	Probe zero point of groove
	GROOVE/PLANE	Interpolation level
	GROOVE/LIFT	Obstacle in groove
	GROOVE/MOVEAX	Measuring axis
	GROOVE/SHIFTAX	Offset
	GROOVE/RESULT[1..3]	Measuring result in the axis descriptions X, Y, Z
	BORE/...	Probe zero point of boring
	BORE/PLANE	Interpolation level
	BORE/LIFT	Obstacle in boring
	BORE/MAINAXSHIFT	Main axis offset
	BORE/AUXAXSHIFT	Auxiliary axis offset
	BORE/RESULT[1..3]	Measuring result in the axis descriptions X, Y, Z
	POINT/...	Probe zero point of point
	POINT/AXIS[1..3]/ENABLE	Enable axis for measuring, index corresponds to axis description

Structural element		Description
	POINT/AXIS[1..3]/DIST	Offset
	POINT/AXIS[1..3]/DIR	Measuring direction
	POINT/AXIS[1..3]/RESULT	Measuring result
	CORNER/...	Probe zero point of corner
	CORNER/PLANE	Interpolation level
	CORNER/ORI	Position external/internal
	CORNER/POS	Position down/up
	CORNER/NO	Corner number
	CORNER/DIST	Offset
	CORNER/MAINAXSHIFT	Main axis offset
	CORNER/AUXAXSHIFT	Auxiliary axis offset
	CORNER/INFAXSHIFT	Infeed axis offset
	CORNER/RESULT[1..3]	Measuring result in the axis descriptions X, Y, Z
	BALL/...	Measuring the zero point of a ball
	BALL/DIR	Measuring axes and infeed direction. The amount corresponds to the axis description and the sign corresponds to the direction.
	BALL/RAD	Ball radius
	BALL/SAVDIS	Safety distance
	BALL/MAINAXSHIFT	Main axis offset
	BALL/AUXAXSHIFT	Auxiliary axis offset
	BALL/INFAXSHIFT	Infeed axis offset
	BALL/RESULT[1..3]	Measuring result in the axis descriptions X, Y, Z

Tab. 32-24: Overview on structural elements SysPrMan

The data of the system structure SysPrMan are exclusively used by the IndraWorks Operation operating interface and by the measuring function. Direct user access is not provided. Thus, the individual structural elements are not described in detail.



For more information about the SysPrMan system data structure, refer to the "IndraMotion MTX Measuring Functions" documentation.

Event data of the axis-parallel measurement "/SysPrResult"

The system data structure SysPrResult is stored in the permanent memory and contains the event data of the axis parallel measurement.

The following structural elements are defined in the "sdprresult.xsd" scheme:

Structural element		Description
CAL[1..2]/ AXIS[1..3]/ DIR[1..2]	/...	Switching offset values determined by the calibration cycle for each axis in positive and negative axis direction
	CALI	Calibration cycle completed successfully
	VAL	Switching offset values
	PRPOS	Switching point in machine coordinates
	FEED	Calibration velocity
	YEAR	Year of calibration
	DATE	Day and month of calibration
	TIME	Time of calibration
	RUNS	Number of measurements after the last calibration
AXISRES[1..3]	-	Measuring result of the basic measurement cycle STP_PrBase for the respective axis description X, Y or Z
RESULTB	-	Boolean measuring result for the automatic measuring cycle G323
RESULT[1..10]	-	Measuring results of the automatic measuring cycles (floating number format with double accuracy)

Tab. 32-25: Overview on structural elements SysPrResult

The data of the system structure SysPrMan are exclusively used by the IndraWorks Operation operating interface and by the measuring function. Direct user access is not provided. Thus, the individual structural elements are not described in detail.



For more information about the SysPrResult system data structure, refer to the "IndraMotion MTX Measuring Functions" documentation.

32.2.35 Return to path logic data "/SysReentry"

General information

The system data structure SysReentry is created in the volatile memory and it is used for all channels. If it should subsequently be returned in several channels, the entries have to be adjusted each time in SysReentry if required.

The following structural elements are defined in the "sdreentr.xsd" schema:

Structure element	Description
FeedVal	Individual return to path velocity
OptFeedVal	Optional velocity if no feed was programmed
TangCircleRadius	Return to path radius
NameOfAsup	Adjustment program name of the return to path logic (ReentryAsup)
Approach	Return to contour
FeedMode	Return to path velocity selection
FeedHold	Activate feed stop after a return to contour
AsupActive	Adjustment program of the return to path logic (ReentryAsup) active

Tab. 32-26: Overview on structure elements "SysReentry"

Individual return to path velocity

Individual return to path velocity		FeedVal
Indicates an individual return to path velocity. It becomes active if the value 2 is entered in "/SysReentry/FeedMode".		
The value can be modified in the part program.		
Path	/SysReentry/FeedVal	
	CPL: SD.SysReentry.FeedVal	
Value range	-	
Pre-assignment	0	
Unit	mm/min	
Description of the variable values	Approaching velocity	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	Immediately	
Update	-	
Dependency	The system date is applied if the value 2 is entered in "/SysReentry/FeedMode".	
Data type	Double_t (8 bytes)	

Optional return to path velocity

Optional return to path velocity	OptFeedVal
Indicates an optional return to path velocity. It is applied if a value of 1 is entered in "/SysReentry/FeedMode" and no valid F-value is applied in the freeze block or if time programming is active (G93).	
If an invalid value is parameterized for the optional return to path velocity, the emergency strategy to use the low jog velocity of the respective first axis in the channel as optional return to path velocity.	

System data

It is recommended to enter a reasonable velocity value. The value can also be entered in the part program.

Path	/SysReentry/OptFeedVal CPL: SD.SysReentry.OptFeedVal
Value range	-
Pre-assignment	0
Unit	mm/min
Description of the variable values	Approaching velocity
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	Immediately
Update	-
Dependency	The system date is applied if the value 1 is entered in "/SysReentry/FeedMode". and if time programming is enabled in the freeze block or if no valid F-value is applied.
Data type	Double_t (8 bytes)

Return to path radius

Return to path radius		TangCircleRadius
Define the radius of the tangential approach circle in case there is a tangential return to path. The value can be modified in the part program.		
Path	/SysReentry/TangCircleRadius CPL: SD.SysReentry.TangCircleRadius	
Value range	-	
Pre-assignment	0	
Unit	mm	
Description of the variable values	Radius of the approaching circle	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	Immediately	
Update	-	
Dependency	-	
Data type	Double_t (8 bytes)	

Name of the adjustment program when returning to path (ReentryAsup)

Name of the adjustment program when returning to path (ReentryAsup)		NameOfAsup
Defines the path and name of the subroutine that is to be called as the adjustment program of the return to path logic.		
In case of missing specifications, no adjustment program of the return to path logic is called.		
Path	/SysRetrace/NameOfAsup CPL: SD.SysRetrace.NameOfAsup	
Value range	100 characters max. from isoLatin1String	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	<EmptyString>	No Asup of the return to path logic is called.
	<Path and name of the ReentryAsups>	Asup of the return to path logic is called.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency		
Data type	isoLatin1String (100 bytes for text + 3 bytes + 1 byte for end identification)	

Return to contour

Return to contour		Approach
Determines if the approaching point is to be approached tangentially or linearly in the arc.		
The value can be modified in the part program.		
Path	/SysReentry/Approach CPL: SD.SysRetrace.Approach	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	1	The approaching point is approached tangentially in the arc.
	0, otherwise	The approaching point is approached linearly.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Immediately	

System data

Update	-
Dependency	-
Data type	Byte_t (1 byte)

Return to path velocity selection

Return to path velocity selection		FeedMode
Defines how the approaching velocity is to be determined. The value can be modified in the part program.		
Path	/SysReentry/FeedMode CPL: SD.SysRetrace.FeedMode	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Approaching velocity = jog velocity
	1	Approaching velocity = active feed in the freeze block (programmed feed)
	2	Approaching velocity = value from "/SysReentry/FeedVal"
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Immediately	
Update	-	
Dependency	-	
Data type	Byte_t (1 byte)	

Feed stop after returning to contour

Activate feed stop after a return to contour		FeedHold
Defines if a feed stop is enabled after a return to contour. Facilitates the activation of intervention functions, e.g. retrace, after a return to contour. The main program can subsequently be continued using an NC start.		
Path	/SysReentry/FeedHold CPL: SD.SysRetrace.FeedHold	
Value range	0..1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Regular program processing after a return to contour
	1	Activate feed stop after a return to contour
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write

	PLC prog.:	Read and write
Activation		Immediately
Update		-
Dependency		-
Data type		Byte_t (1 byte)

Adjustment program of the return to path logic (ReentryAsup) active

Adjustment program of the return to path logic (ReentryAsup) active		AsupActive
The variable indicates whether a channel is in the asup of the return to path logic.		
To determine whether the program is currently processed as ReentryAsup, a prompt is sent.		
Path	/SysReentry/AsupActive	
	CPL: SD.SysReentry.AsupActive	
Value range	-	
Pre-assignment	-	
Unit	-	
Description of the variable values	0	There is no channel in the asup of the return to path logic
	1	There is one channel in the asup of the return to path logic
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Byte_t (1 byte)	

32.2.36 Retrace logic data "/SysRetrace"

General information

The volatile system data structure SysRetrace is created for each channel. The following structural elements are defined in the "sdreentr.xsd" schema:

Structural element	Description
NameOfAsup	Name of the adjustment program of the retrace logic (RetraceAsup)
AsupActive	Adjustment program of the retrace logic (RetraceAsup) active

Tab. 32-27: Overview on structural elements "SysRetrace"

Name of the adjustment program of the retrace logic (RetraceAsup)

Name of the adjustment program of the retrace logic (RetraceAsup)	NameOfAsup

System data

For each channel, defines the path and name of the subroutine that is to be called as the adjustment program of the retrace logic.

In case of missing specifications, no adjustment program of the retrace logic is called.

Path	/SysRetrace/NameOfAsup CPL: SD.SysRetrace[<Channel>].NameOfAsup - or - SD.SysRetrace.NameOfAsup in the individual channel	
Value range	100 characters max. from isoLatin1String	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	<EmptyString>	No Asup of the retrace logic is called.
	<Path and name of the asup>	Asup of the retrace logic is called.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		
Update	-	
Dependency		
Data type	isoLatin1String (100 bytes for text + 3 bytes + 1 byte for end identification)	

Adjustment program of the retrace logic (RetraceAsup) active

Adjustment program of the retrace logic (RetraceAsup) active		AsupActive
The variable is not 0 if the channel is in the asup of the retrace logic. To determine whether the program is currently processed as RetraceAsup, a prompt is sent.		
Path	/SysRetrace/AsupActive CPL: SD.SysRetrace[<Channel>].AsupActive - or - SD.SysRetrace.AsupActive in the individual channel	
Value range	-	
Pre-assignment	-	
Unit	-	
Description of the variable values	unequal to 0	Channel is in the asup of the retrace logic
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	CPL:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel	
Dependency	-	

Data type Byte_t (1 byte)

32.2.37 List of the Sercos devices "/SysScsDevice" connected

General information

The volatile system data structure "SysScsDevice" is created via the definition file "SDDef.xml".

SysScsDevice is provided with entries for up to 315 Sercos devices (250 axes + 64 I/O + 1). This includes data from the IP devices connected to the control. These devices communicate via the Sercos UC channel. This data is saved in ring topology order.

The following structural elements are defined in the "sdscsdev.xsd" schema:

Structural element	Description
LogAxisNmb	Logic axis number
ScsAddress	Sercos address
IpAddress	IP address

Tab. 32-28: Overview on structural elements "SysScsDevice"

Logic axis number

Logic axis number	LogAxisNmb

It contains the logic axis number of the Sercos IP devices.

Path /SysScsDevice[1..164]/LogAxisNmb

CPL: SD.SysScsDevice[1..164].LogAxisNmb

Value range 0, 1..254

Pre-assignment 0

Unit -

Description of the variable values Logic axis number

Access rights

MTB:	Read
Setter:	Read
User:	Read
NC prog.:	Read
PLC prog.:	Read

Activation -

Update By the NC kernel during control startup.

Dependency -

Data type UnsignedShort_t (2 bytes)

Sercos address

Sercos address	ScsAddress

It contains the Sercos address of the Sercos IP devices.

Path /SysScsDevice[1..164]/ScsAddress

CPL: SD.SysScsDevice[1..164].ScsAddress

Value range 0, 1..254

Pre-assignment 0

System data

Unit	-
Description of the variable values	Sercos address
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	By the NC kernel during control startup.
Dependency	-
Data type	UnsignedShort_t (2 bytes)

IP address

	<table border="1"> <tr> <th>IP address</th><th>IpAddress</th></tr> </table>	IP address	IpAddress
IP address	IpAddress		
	Contains the IP address of the Sercos IP devices.		
Path	/SysScsDevice[1..164]/IpAddress CPL: SD.SysScsDevice[1..164].IpAddress		
Value range	Characters from IsoLatin1String		
Pre-assignment	<EmptyString>		
Unit	-		
Description of the variable values	IP address		
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read		
Activation	-		
Update	By the NC kernel during control startup.		
Dependency	-		
Data type	IsoLatin1String (15 bytes for text + 1 byte for end identification)		

32.2.38 IP addresses of the Sercos interface "/SysScsIpAdr"

General information

The permanent system data structure "SysScsIpAdr" is created via the definition file "sddefsct.xml".

It contains the IP address of the Sercos interface and the one of the respective subnet used to run the Sercos UC channel communication. Additionally, the automatic IP specification is enabled. The values are entered during installation.

The following structural elements are defined in the "sdscsdev.xsd" schema:

Structural element	Description
IPAddMtx	Sercos IP address of the control
IPAddSubDev	Subnet mask of the Sercos IP subnet
AutoIPAddr	Enabling the automatic IP address specification

Tab. 32-29: Overview on structural elements "SysScslpAdr"

Sercos IP address of the control

Sercos IP address of the control		IPAddMtx
	It contains the IP address of the Sercos interface on the NC control.	
Path	/SysScslpAdr/IPAddMtx CPL: SD.SysScslpAdr.IPAddMtx	
Value range	Characters from IsoLatin1String	
Pre-assignment	192.168.143.254	
Unit	-	
Description of the variable values	IP address of the control	
Access rights	MTB:	Write
	Setter:	Write
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	-	
Dependency	-	
Data type	IsoLatin1String (15 bytes for text + 1 byte for end identification)	

Sercos IP address of the subnet

Subnet mask of the Sercos IP subnet		IPAddSubDev
	It indicates the subnet mask of the subnet belonging to the Sercos interface on the NC control. The subnet mask is fixedly set to the value "255.255.255.0".	
Path	/SysScslpAdr/IPAddSubDev CPL: SD.SysScslpAdr.IPAddSubDev	
Value range	Characters from IsoLatin1String	
Pre-assignment	255.255.255.0	
Unit	-	
Description of the variable values	Subnet mask of the Sercos IP subnet	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	-	

Dependency -

Data type IsoLatin1String (15 bytes for text + 1 byte for end identification)

Sercos IP address of the devices

Enabling the automatic IP address specification		AutoIPAddr
The automatic IP address specification automatically assigns the IP addresses to the devices in the Sercos ring. The IP addresses of the Sercos devices orient themselves at the parameters of the IP address of the Sercos interface ("/SysScsIpAdr") and the subnet mask ("/IPAdSubDev"). These two parameters define the IP address range of the subnet. The Sercos devices are numbered according to their Sercos address in this subnet.		
Path	/SysScsIpAdr/AutoIPAddr CPL: SD.SysScsIpAdr.AutoIPAddr	
Value range	0 (FALSE) 1 (TRUE)	
Pre-assignment	1 (TRUE)	
Unit	-	
Description of the variable values	Enabling the automatic IP address specification	
Access rights	MTB: Write Setter: Write User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	-	
Dependency	-	
Data type	Boolean_t (1 byte)	

32.2.39 Settings for automatic Sercos parameterization"/SysScsDrPar"

General information

The permanent system data structure "SysScsIpAdr" is created via the definition file "SDDef.xml".

The number of entries is dynamic and depends on the number of NC-controlled drives.

The following structural elements are defined in the "sdscsdev.xsd" schema:

Structural element	Description
ParkAx	Parking axis

Tab. 32-30: Overview on structural elements SysScsDrPar

Parking axis

Parking axis	ParkAx
Specifies if the function "Parking axis" (command S-0-0139) is to be enabled in the drive during the sercos startup in phase "CP2". If no motor/encoder is connected to the drive amplifier, freezing in phase "CP3" of the Sercos start-up is prevented after switching the system on.	

Path	/SysScsDrPar[<Drive>]/ParkAx CPL: SD.SysScsDrPar[<Drive>].ParkAx	
Value range	0 (FALSE) 1 (TRUE)	
Pre-assignment	0 (FALSE)	
Unit	-	
Description of the variable values	0 (FALSE)	Parking axis is not enabled
	1 (TRUE)	Parking axis is enabled
Access rights	MTB:	Write
	Setter:	Write
	User:	Read
	NC prog.:	Write
	PLC prog.:	Write
Activation	-	
Update	-	
Dependency	-	
Data type	Boolean_t (1 byte)	

32.2.40 Control data to re-initialize selected SD elements after NC restart "/SysSDForce"

General information

Permanent system data (SD) are newly created once (upon the first restart) and initialized with "0", empty string or values from a default setting file SDDat*.xml.

System data (permanent and volatile SD) can be specifically re-initialized irrespective of the default setting data using the following two files (search path: "root directory --> /usrfep", templates in "/feprom"):

SDForceByCmd1SnglRestart.xml → One-time re-initialization of selected SD elements after the next NC restart

SDForceByCmd2EachRestart.xml → Repetitive re-initialization of selected SD elements after each NC restart

The permanent system data structure SysSDForce controls this file import. SysSDForce is created using the definition file "SDDef.xml" with the entries from the default setting file "SDDat.xml".

The following structural elements are defined in the "sdforcesd.xsd" schema:

Structural element	Description
SDForceCmd1SnglRestart	Controls the one-time re-initialization of SD elements after the next NC restart
SDForceCmd2EachRestart	Controls the repetitive re-initialization of SD elements after each NC restart

Tab. 32-31: Overview on structural elements "SysSDForce"

One-time re-initialization of SD elements after the next NC restart

One-time re-initialization of SD elements after NC restart		SDFor- ceCmd1SnglRes- tart
Controls the import of the file SDForceByCmd1SnglRestart.xml into the search path "root directory --> /usrfep" for a one-time re-initialization of SD elements after the next NC restart.		
Path	/SysSDForce/SDForceCmd1SnglRestart CPL: SD.SysSDForce.SDForceCmd1SnglRestart	
Value range	0 (FALSE) 1 (TRUE)	
Pre-assignment	1 (TRUE)	
Unit	-	
Description of the variable values	0 (FALSE) The file SDForceByCmd1SnglRestart.xml is not imported (anymore) after the following NC restart. If it is to be imported again (e.g. after a change), it has to be triggered again first (SD.SysSDForce.SDForceCmd1SnglRestart = TRUE) 1 (TRUE) The file SDForceByCmd1SnglRestart.xml is imported after the following NC restart if found. After the file SDForceByCmd1SnglRestart.xml has been imported successfully, this SD element is set to FALSE. This ensures that the named file is only imported once.	
Access rights	MTB: Write Setter: Write User: Read NC prog.: Write PLC prog.: Write	
Activation	System cold start	
Update	Upon system restart if necessary	
Dependency	-	
Data type	Boolean_t (1 byte)	

Repetitive re-initialization of SD elements after the next NC restart

Repetitive re-initialization of SD elements after NC restart		SDFor- ceCmd2EachRes- tart
Controls the import of the file SDForceByCmd2EachRestart.xml into the search path "root directory --> /usrfep" for a repetitive re-initialization of SD elements after each NC restart.		
Path	/SysSDForce/SDForceCmd2EachRestart CPL: SD.SysSDForce.SDForceCmd2EachRestart	
Value range	0 (FALSE) 1 (TRUE)	
Pre-assignment	1 (TRUE)	
Unit	-	

Description of the variable values	0 (FALSE)	The file SDForceByCmd2EachRestart.xml is not imported anymore after the NC restart.
	1 (TRUE)	The file SDForceByCmd2EachRestart.xml is imported again after each NC restart.
Access rights	MTB:	Write
	Setter:	Write
	User:	Read
	NC prog.:	Write
	PLC prog.:	Write
Activation	System cold start	
Update	-	
Dependency	-	
Data type	Boolean_t (1 byte)	

32.2.41 Data of the memory space assignment of system data "/SysSdInfo"

General information

The information on the memory space assignment of system data is stores in the volatile system data structure "SysSdInfo" during control startup.

The following structural elements are defined in the schema "SysSdinfo.xsd".

Structural element	Description
CustPermMax	Maximum memory of permanent customer system data
CustVolMax	Maximum memory of volatile customer system data
CustPermUsed	Current memory of permanent customer system data
CustVolUsed	Current memory of volatile customer system data
CustTmpUsed	Current memory of temporary customer system data
SysPermUsed	Current memory of internal permanent system data
SysVolUsed	Current memory of internal volatile system data
SysTmpUsed	Current memory of internal temporary system data

Tab. 32-32: Overview on the structure element "SysSdInfo"

Maximum memory of permanent customer system data

Maximum memory of permanent customer system data		CustPermMax
Path	/SysSdInfo/CustPermMax CPL: SD.SysSdInfo.CustPermMax	
Value range	-	
Pre-assignment	0	

System data

Unit	Bytes
Description of the variable values	Memory assignment
Access rights	MTB: Read
	Setter: Read
	User: Read
	NC prog.: Read
	PLC prog.: Read
Activation	-
Update	By the NC kernel at startup
Dependency	-
Data type	UnsignedInt_t (4 bytes)

Maximum memory of volatile customer system data

	Maximum memory of volatile customer system data	CustVolMax
Path	/SysSdInfo/CustVolMax CPL: SD.SysSdInfo.CustVolMax	
Value range	-	
Pre-assignment	0	
Unit	Bytes	
Description of the variable values	Memory assignment	
Access rights	MTB: Read	
	Setter: Read	
	User: Read	
	NC prog.: Read	
	PLC prog.: Read	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	UnsignedInt_t (4 bytes)	

Current memory of permanent customer system data

	Current memory of permanent customer system data	CustPermUsed
Path	/SysSdInfo/CustPermUsed CPL: SD.SysSdInfo.CustPermUsed	
Value range	-	
Pre-assignment	0	
Unit	Bytes	
Description of the variable values	Memory assignment	
Access rights	MTB: Read	
	Setter: Read	
	User: Read	
	NC prog.: Read	
	PLC prog.: Read	

Activation	-
Update	By the NC kernel at startup
Dependency	-
Data type	UnsignedInt_t (4 bytes)

Current memory of volatile customer system data

Current memory of volatile customer system data		CustVolUsed
Path	/SysSdInfo/CustVolUsed CPL: SD.SysSdInfo.CustVolUsed	
Value range	-	
Pre-assignment	0	
Unit	Bytes	
Description of the variable values	Memory assignment	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	UnsignedInt_t (4 bytes)	

Current memory of temporary customer system data

Current memory of temporary customer system data		CustTmpUsed
Path	/SysSdInfo/CustTmpUsed CPL: SD.SysSdInfo.CustTmpUsed	
Value range	-	
Pre-assignment	0	
Unit	Bytes	
Description of the variable values	Memory space assignment	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	UnsignedInt_t (4 bytes)	

Current memory of internal permanent system data

Current memory of internal permanent system data	SysPermUsed

System data

Path	/SysSdInfo/SysPermUsed CPL: SD.SysSdInfo.SysPermUsed	
Value range	-	
Pre-assignment	0	
Unit	Bytes	
Description of the variable values	Memory space assignment	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	UnsignedInt_t (4 bytes)	

Current memory of internal volatile system data

Current memory of internal volatile system data		SysVolUsed
Path	/SysSdInfo/SysVolUsed CPL: SD.SysSdInfo.SysVolUsed	
Value range	-	
Pre-assignment	0	
Unit	Bytes	
Description of the variable values	Memory space assignment	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	UnsignedInt_t (4 bytes)	

Current memory of internal temporary system data

Path	/SysSdInfo/SysTmpUsed CPL: SD.SysSdInfo.SysTmpUsed	
Current memory of internal temporary system data		SysTmpUsed
Value range	-	
Pre-assignment	0	
Unit	Bytes	
Description of the variable values	Memory assignment	
Access rights	MTB:	Read

Setter:	Read
User:	Read
NC prog.:	Read
PLC prog.:	Read
Activation	-
Update	By the NC kernel at startup
Dependency	-
Data type	UnsignedInt_t (4 bytes)

32.2.42 Actual data of system spindles "/SysSpActData"

General information

The volatile system data structure SysSpActData is created for each system spindle.

The following structural elements are defined in the "sdspndl.xsd" schema:

Structural element	Description
Speed	Active command spindle speed
LastOriPos	Latest activated orientation position

Tab. 32-33: Overview on structural elements "SysSpActData"

Active command spindle speed

Active command spindle speed	Speed

Delivers the active command spindle speed for the respective system spindle.

Commands of "Jog spindle" are not included.

Path /SysSpActData[<Spindle>]/Speed

CPL: SD.SysSpActData[<Spindle>].Speed

Value range -

Pre-assignment 0

Unit rpm

Description of the variable values Command spindle speed

Access rights **MTB:** Read

Setter: Read

User: Read

NC prog.: Read

PLC prog.: Read

Activation -

Update By the NC kernel

Dependency -

Data type Float_t (4 bytes)

Latest activated orientation position

Latest activated orientation position	LastOriPos

Delivers the last activated orientation position for the respective system spindle.

System data

Path	/SysSpActData[<Spindle>]/LastOriPos CPL: SD.SysSpActData[<Spindle>].LastOriPos	
Value range	-	
Pre-assignment	0	
Unit	°	
Description of the variable values	Orientation position	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel for every spindle orientation (M19)	
Dependency	-	
Data type	Float_t (4 bytes)	

32.2.43 Command data for system spindles "/SysSpCmdData"

General information

The volatile system data structure "SysSpCmdData" is created for each system spindle.

The following structural elements are defined in the "sdspndl.xsd" schema:

Structural element	Description
MaxSpeedLimit	Upper speed limit value of the spindle
MinSpeedLimit	Lower speed limit value of the spindle
SpeedGearRatio	Transmission ratio for programmed spindle speed
Speed	Command spindle speed
OriPos	Orientation position for part program
PlcOriPos	Orientation position for PLC specifications
Tool	Tool-specific limitations
ThreadOverride	Thread cutting G33 with spindle override
ReverseRotation	Reversed direction of the spindle rotation

Tab. 32-34: Overview on structural elements "SysSpCmdData"

Tool limitations (SysSpCmdData[<Spindle>]/Tool):

Structural element	Description
Enable	Release "TOOL" contents for use
MaxSpSpeed	Upper speed limit value
MaxAcc	Maximum acceleration
MaxPosAcc1	Maximum acceleration in position control mode 1
MaxPosAcc2	Maximum acceleration in position control mode 2
MaxSwiSpeed	Max. switching speed for position control mode 2

Structural element	Description
N_Input	Input revolutions
N_Output	Output revolutions

Tab. 32-35: Overview on structural elements SysSpCmdData[]/Tool

Upper speed limit value of the spindle

Upper speed limit value of the spindle		MaxSpeedLimit
	Specifies the upper speed limit value for the system spindle.	
Path	/SysSpCmdData[<Spindle>]/MaxSpeedLimit CPL: SD.SysSpCmdData[<Spindle>].MaxSpeedLimit	
Value range	-	
Pre-assignment	0	
Unit	rpm	
Description of the variable values	Upper speed limit value	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	With the rising edge of interface signal "qSp_SpeedLimit".	
Update	-	
Dependency	-	
Data type	Float_t (4 bytes)	

Lower speed limit value of the spindle

Lower speed limit value of the spindle		MinSpeedLimit
	Specifies the lower speed limit value for the system spindle.	
Path	/SysSpCmdData[<Spindle>]/MinSpeedLimit CPL: SD.SysSpCmdData[<Spindle>].MinSpeedLimit	
Value range	-	
Pre-assignment	0	
Unit	rpm	
Description of the variable values	Lower speed limit value	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	With the rising edge of interface signal "qSp_SpeedLimit".	
Update	-	
Dependency	-	
Data type	Float_t (4 bytes)	

Conversion factor for spindle speed

Conversion value for spindle speed	SpeedGearRatio

Defines the gear ratio for the programmed spindle speed for the respective system spindle. The defined speed specification for the drive is multiplied with the content of SpeedGearRatio.

Before NC-internal processing, the actual speed of the drive is divided by the content of "SpeedGearRatio".



For new applications, please use the function "Tool-specific limitations(/SysSpCmdData[<Spindle>]/Tool)" [tab. 32-35 "Overview on structural elements SysSpCmdData\[\]/Tool" on page 570.](#)



Please refer to the documentation "MTX 15VRS Functional Description - Basics" (DOK-MTX***-NC*F*BA*V15-RERS-EN-P), chapter "Spindle functions".

Path	/SysSpCmdData[<Spindle>]/SpeedGearRatio CPL: SD.SysSpCmdData[<Spindle>].SpeedGearRatio	
Value range	-	
Pre-assignment	1.0	
Unit	-	
Description of the variable values	Transmission ratio	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Immediately	
Update	-	
Dependency	-	
Data type	Float_t (4 bytes)	

Command spindle speed

Command spindle speed	Speed

Specifies the spindle speed for the system spindle.



Please refer to the documentation "MTX 15VRS Functional Description - Basics" (DOK-MTX***-NC*F*BA*V15-RERS-EN-P), chapter "Spindle functions".

Path	/SysSpCmdData[<Spindle>]/Speed CPL: SD.SysSpCmdData[<Spindle>].Speed	
Value range	-	
Pre-assignment	0	
Unit	rpm	
Description of the variable values	Spindle speed	

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	With the rising edge of interface signal "qSp_SValueSD".	
Update	-	
Dependency	-	
Data type	Float_t (4 bytes)	

Orientation position for part program

Orientation position for part program		OriPos
Defines the orientation position for the part program for the respective system spindle.		
Path	/SysSpCmdData[<Spindle>]/OriPos CPL: SD.SysSpCmdData[<Spindle>].OriPos	
Value range	-	
Pre-assignment	0	
Unit	°	
Description of the variable values	Orientation position	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	With the rising edge of interface signal "qSp_SValueSD".	
Update	-	
Dependency	-	
Data type	Float_t (4 bytes)	

Orientation position for PLC specifications

Orientation position for PLC specifications		PlcOriPos
Defines the orientation position for specifications via the PLC bit interface for the respective system spindle.		
Path	/SysSpCmdData[<Spindle>]/PlcOriPos CPL: SD.SysSpCmdData[<Spindle>].PlcOriPos	
Value range	-	
Pre-assignment	0	
Unit	°	
Description of the variable values	Orientation position	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write

System data

	PLC prog.:	Read and write
Activation	With the rising edge of interface signal "qSp_Orientate"	
Update	-	
Dependency	-	
Data type	Float_t (4 bytes)	

Releasing the tool-specific limitations

Releasing the tool-specific limitations	Tool.Enable

Specifies the tool-specific limitations for the affected system spindle.



For more information, refer to the "MTX 15VRS Functional Description - Basics" (DOK-MTX***-NC*F*BA*V15-RERS-EN-P) in the chapter "Spindle functions" under "Tool-specific spindle limitations".

Path	/SysSpCmdData[<Spindle>]/Tool/Enable	
	CPL: SD.SysSpCmdData[<Spindle>].Tool.Enable	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Calling the NC function SToolLim(<ChSp>=1) or SSpToolLim(<Spindle>,1) enables the machine parameters. The transmission ratio of the tool is set to 1:1.
	1	Calling the NC function SToolLim(<ChSp>=1) or SSpToolLim(<Spindle>,1) enables the set values.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Upper speed limit value of the tool

Upper speed limit value of the tool	Tool.MaxSpeed

Specifies the upper speed limit value of the tool for the system spindle.

Path	/SysSpCmdData[<Spindle>]/Tool/MaxSpeed	
	CPL: SD.SysSpCmdData[<Spindle>].Tool.MaxSpeed	
Value range	≥0	
Pre-assignment	0	
Unit	rpm	
Description of the variable values	0	No tool-specific speed limit value.
	>0	Upper speed limit value of the tool.

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	In case of a set ../Tool/Enable (value 1), calling the NC function SToolLim() or SSpToolLim() enables the set value.	
Update	-	
Dependency	-	
Data type	Float_t (4 bytes)	

Maximum acceleration of the tool

Maximum acceleration of the tool	Tool.MaxAcc

Specifies the upper acceleration limit value of the tool for the system spindle.



Acceleration values of drive controlled functions, e.g. "spindle positioning" or "spindle oscillation" are stored in the drive and therefore not affected by this parameter.

Path	/SysSpCmdData[<Spindle>]/Tool/MaxAcc CPL: SD.SysSpCmdData[<Spindle>].Tool.MaxAcc	
Value range	>=0	
Pre-assignment	0	
Unit	rad/sec ²	
Description of the variable values	0	No tool-specific acceleration limit.
	>0	Maximum acceleration of the tool.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	In case of a set ../Tool/Enable (value 1), calling the NC function SToolLim() or SSpToolLim() enables the set value.	
Update	-	
Dependency	-	
Data type	Float_t (4 bytes)	

Maximum acceleration in position control mode 1

Maximum acceleration in position control mode 1	Tool.MaxPosAcc1

Specifies an additional acceleration limit in position control mode 1 for the system spindle.

Path	/SysSpCmdData[<Spindle>]/Tool/MaxPosAcc1 CPL: SD.SysSpCmdData[<Spindle>].Tool.MaxPosAcc1	
Value range	>=0	
Pre-assignment	0	

System data

	Unit	rad/sec ²	
Description of the variable values	0	No additional tool-specific acceleration limit in position control mode 1.	
		The maximum acceleration of the tool is active.	
	>0	Maximum acceleration in position control mode 1.	
Access rights	MTB:	Read and write	
	Setter:	Read and write	
	User:	Read and write	
	NC prog.:	Read and write	
	PLC prog.:	Read and write	
Activation	In case of a set ../Tool/Enable (value 1), calling the NC function SToolLim() or SSpToolLim() enables the set value.		
Update	-		
Dependency	MaxPosAcc1 is internally limited to the active maximum acceleration.		
Data type	Float t (4 bytes)		

Maximum acceleration in position control mode 2

Maximum acceleration in position control mode 2		Tool.MaxPosAcc2
	Specifies an additional acceleration limit in position control mode 2 for the system spindle.	
Path	/SysSpCmdData[<Spindle>]/Tool/MaxPosAcc2 CPL: SD.SysSpCmdData[<Spindle>].Tool.MaxPosAcc2	
Value range	>=0	
Pre-assignment	0	
Unit	rad/sec ²	
Description of the variable values	0	No additional tool-specific acceleration limit in position control mode 2.
	>0	Maximum acceleration in position control mode 1.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	In case of a set ../Tool/Enable (value 1), calling the NC function SToolLim() or SSpToolLim() enables the set value.	
Update	-	
Dependency	MaxPosAcc2 is internally limited to the active maximum acceleration during position control mode 1.	
Data type	Float_t (4 bytes)	

Max. switching speed to position control mode 2

Max. switching speed to position control mode 2	Tool.MaxSwiSpeed
Specifies value 2 for the maximum changeover speed to position control mode for the system spindle.	
Path	/SysSpCmdData[<Spindle>]/Tool/MaxSwiSpeed

	CPL: SD.SysSpCmdData[<Spindle>].Tool.MaxSwiSpeed
Value range	≥ 0
Pre-assignment	0
Unit	rpm
Description of the variable values	0 No tool-specific limit of the switching speed. >0 Max. switching speed to position control mode 2.
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	In case of a set ../Tool/Enable (value 1), calling the NC function SToolLim() or SSpToolLim() enables the set value.
Update	-
Dependency	-
Data type	Float_t (4 bytes)

Input revolutions of the tool

Input revolutions of the tool		Tool.N_Input
	Specifies the tool-specific transmission ratio for the system spindle together with N_Output.	
Path	/SysSpCmdData[<Spindle>]/Tool/N_Input CPL: SD.SysSpCmdData[<Spindle>].Tool.N_Input	
Value range	≥ 0	
Pre-assignment	0	
Unit	-	
Description of the variable values	0 The transformation ratio for N_Output = 0 is 1.0. >0 The transformation ratio for N_Output > 0 is N_Input/N_Output.	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	In case of a set ../Tool/Enable (value 1), calling the NC function SToolLim() or SSpToolLim() enables the set value.	
Update	-	
Dependency	N_Input and N_Output specify the transmission ratio of the tool. N_Input = 0 and N_Output = 0: The transmission ratio is 1.0. Others: The transmission ratio is N_Input/N_Output. The permissible transmission ratio is between 0.0001 and 10000.0.	
Data type	Float_t (4 bytes)	

Output revolutions of the tool

Input revolutions of the tool		Tool.N_Output
	Specifies the tool-specific transmission ratio for the system spindle together with N_input.	
Path	/SysSpCmdData[<Spindle>]/Tool/N_Output CPL: SD.SysSpCmdData[<Spindle>].Tool.N_Output	
Value range	≥0	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	The transformation ratio for N_Input = 0 is 1.0.
	>0	The transformation ratio for N_Input > 0 is N_Input/N_Output.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	In case of a set ../Tool/Enable (value 1), calling the NC function SToolLim() or SSpToolLim() enables the set value.	
Update	-	
Dependency	N_Input and N_Output specify the transmission ratio of the tool.	
	N_Input = 0 and N_Output = 0: The transmission ratio is 1.0.	
	Others: The transmission ratio is N_Input/N_Output. The permissible transmission ratio is between 0.0001 and 10000.0.	
Data type	Float_t (4 bytes)	

Thread cutting spindle override

Thread cutting with spindle override		ThreadOverride
	Enables the spindle override for the affected system spindle if the spindle is used as main spindle during thread cutting.	
	 A fast change of the spindle override can sporadically result in a damage of the thread during machining.	
Path	/SysSpCmdData[<Spindle>]/ThreadOverride CPL: SD.SysSpCmdData[<Spindle>].ThreadOverride	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	The spindle override position does not affect the spindle speed during thread cutting.
	1	The spindle override is active during thread cutting.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write

	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	At the point of block preparation, thread cutting applies the current ThreadOverride content.	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Reversed direction of the spindle rotation

Reversed direction of the spindle rotation	ReverseRotation

Reverses the direction of the drive rotation for the respective system spindle for M3/M4 and G63.



This system date acts like a machine parameter.
The set value is only applied upon control startup.

Path	/SysSpCmdData[<Spindle>]/ReverseRotation CPL: SD.SysSpCmdData[<Spindle>].ReverseRotation	
Value range	0, 1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	The direction of the spindle rotation is not reversed.
	1	The direction of the spindle rotation is reversed.
Access rights	MTB:	Read and write
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	It can only be written to this system date using an SDDat file. The content of the SDDat file is applied upon control startup.	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

32.2.44 Block pre-run data"/SysSRun"

General information

The permanent system data structure "SysSRun" is created for each channel.

The following structural elements are defined in the "sdsrun.xsd" schema:

Structural element	Description
Active	See: SysChSRun.Active
AsupActive	Adjustment program active
LookAheadPreRun	See: SysChSRun.LookAheadPreRun
Reserved01	Reserved

Structural element	Description
TargPos	See: SysChSRun.TargPos
TargBlockSrc	See: SysChSRun.TargBlockSrc
ReEntryIndex	Latest used storage index of abort information
LockTargBlock	See: SysChSRun.LockTargBlock
LockWarning	Note text for blocked target block
NameOfAsup	Name of target block adjustment program (Asup)
NameOfInitAdjPrg	Name of the Init adjustment program
ActionBlock	Action block
Timeout	Runtime monitoring for block pre-run
RetPaStrategy	Approaching strategy after block pre-run
TargetBlkCheck	Backed up point of interruption available
ManWaitForStart	Manual data input after block pre-run
SuppressMsg	Message suppression of messages in block pre-run
ReEntryStore	Backing up aborted information for a later re-entry

Tab. 32-36: Overview on the "SysSRun" structural elements

Current storage index of the program stack file (call chain files)

Current re-entry index of the program stack files (call chain files)	ReEntryIndex

Defines the current storage index of the program stack files (call chain files) in the corresponding channel.

The "ReEntryIndex" labels the latest written program stack file.

Path /SysSRun[<Channel>]/ReEntryIndex

CPL: SD.SysSRun[<Channel>].ReEntryIndex - or -
SD.SysSRun.ReEntryIndex in the individual channel

Value range Integer range

Pre-assignment 0

Unit -

Description of the variable values Current storage index of the program stack file

Access rights

MTB:	Read
Setter:	Read
User:	Read
NC prog.:	Read
PLC prog.:	Read

Activation -

Update Whenever a program stack file is stored

Dependency [chapter "Saving program stack \(callchain\) at reset" on page 586](#)

Data type Int_t (4 bytes)

Note text for locked target block

Note text for blocked target block	LockWarning

For each channel, defines an individual note text if the target block is locked. If nothing is specified here, the general warning in the respective language "This target block is locked." is output. The ID "%s" can be used in the text. It is replaced by the general warning.	
Path	/SysSRun[<Channel>]/LockWarning CPL: SD.SysSRun[<Channel>].LockWarning - or - SD.SysSRun.LockWarning in the individual channel
Value range	Characters from IsoLatin1String
Pre-assignment	<EmptyString>
Unit	-
Description of the variable values	<EmptyString> If the target block is locked, a general warning is output.
	<Note text> If the target block is locked, this note text is output.
Access rights	MTB: Read and write
	Setter: Read and write
	User: Read and write
	NC prog.: Read and write
	PLC prog.: Read and write
Activation	Becomes active immediately in the block preparation.
Update	-
Dependency	-
Data type	IsoLatin1String (80 bytes for text + 3 byte + 1 byte for end identification)

Name of the target block adjustment program

Name of the target block adjustment program		NameOfAsup
Defines the name including the path specification of the program to be called a target block adjustment program. If no program is specified, no target block adjustment program is called before the approaching movement.		
Path	/SysSRun[<Channel>]/NameOfAsup CPL: SD.SysSRun[<Channel>].NameOfAsup SD.SysSRun.NameOfAsup in the individual channel	
Value range	Characters from IsoLatin1String	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	<EmptyString>	No target block adjustment program is called before the approaching movement.
	<invalid program name>	The indicated target block adjustment program is called before the approaching movement.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write

System data

Activation	Becomes active immediately in the block preparation.
Update	-
Dependency	-
Data type	IsoLatin1String (100 bytes for text + 3 bytes + 1 byte for end identification)

Name of init adjustment program

Name of the Init adjustment program		NameOfInitAdjPrg
For each channel, defines the path and name of the subroutine that is to be called as the initialization adjustment program.		
If no subroutine is specified, no Init adjustment program is called before the approaching movement.		
Path	/SysSRun[<Channel>]/NameOfInitAdjPrg CPL: SD.SysSRun[<Channel>].NameOfInitAdjPrg - or - SD.SysSRun.NameOfInitAdjPrg in the individual channel	
Value range	Characters from IsoLatin1String	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	<EmptyString>	No Init adjustment program is called before the approaching movement.
	<invalid program name>	The indicated Init adjustment program is called before the approaching movement.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Becomes active immediately in the block preparation.	
Update	-	
Dependency	-	
Data type	IsoLatin1String (100 bytes for text + 3 bytes + 1 byte for end identification)	

Action block

Action block		ActionBlock
Defines an action block for each channel.		
If an NC block is specified here, it is processed at the start of machining after a block pre-run and before the start of the adjustment program which is also defined.		
Path	/SysSRun[<Channel>]/ActionBlock CPL: SD.SysSRun[<Channel>].ActionBlock - or - SD.SysSRun.ActionBlock in the individual channel	
Value range	Characters from IsoLatin1String	
Pre-assignment	<EmptyString>	
Unit	-	

Description of the variable values	<EmptyString>	There is no action block.
	<Action block>	When machining is started after block pre-run, the defined action block is processed first.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Becomes active immediately in the block preparation.	
Update	-	
Dependency	-	
Data type	IsoLatin1String (512 bytes for text + 3 bytes + 1 byte for end identification)	

Runtime monitoring for block pre-run

Runtime monitoring for block pre-run		Timeout
It allows canceling the block pre-run in the respective channel after monitoring time has elapsed.		
The runtime monitoring can be used if a target block was skipped in a program due to an unfulfilled condition and if the block pre-run is not automatically closed due to an endless loop in the program. Programs containing potential endless loops should set this system data to a sufficient high value at the beginning and subsequently restore it.		
 If the control load is high, the block pre-run might be aborted due to runtime exceedance in the case that the timeout value is only set a bit over the block pre-run runtime of the unloaded control. Set a sufficiently high value or completely disable monitoring.		
Path	/SysSRun[<Channel>]/Timeout CPL: SD.SysSRun[<Channel>].Timeout - or - SD.SysSRun.Timeout in the individual channel	
Value range	-	
Pre-assignment	0	
Unit	ms	
Description of the variable values	0	The runtime monitoring is deactivated.
	>0	The runtime monitoring is active (e.g. 10000 ms).
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	CPL:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	The modification of the value is only applied at next block pre-run.	
Update	-	
Dependency	-	

System data

Data type Int_t (4 bytes)

Approaching strategy after block pre-run

	Approaching strategy after block pre-run	RetPaStrategy
	It defines in the respective channel whether the approaching block should move to the starting point, the interruption or the end point of the target block after block pre-run.	
	This system data is stored permanently.	
Path	/SysSRun[<Channel>]/RetPaStrategy CPL: SD.SysSRun[<Channel>].RetPaStrategy - or - SD.SysSRun.RetPaStrategy in the individual channel	
Value range	0..2	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Approaching the starting point After block pre-run, the approaching block moves to the starting point of the target block.
	1	Approaching the end point After block pre-run, the approaching block moves to the end point of the target block.
	2	After block pre-run, the approaching block moves to the point of interruption. An abort marker has to be used during the block pre-run parameterization.
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	The value valid when starting the block pre-run is now applied. A change during the block pre-run and the adjustment does not influence the approaching movement.	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Backed up point of interruption available

	Backed up freeze block available	TargetBlkCheck
	Enables the check of backed up points of interruption in the part program when the program is selected.	
	The result is stored in the system data /SysChSRun[<Channel>]/TargetBlkAvailable.	
Path	/SysSRun[<Channel>]/TargetBlkCheck CPL: SD.SysSRun[<Channel>].TargetBlkCheck - or - SD.SysSRun.TargetBlkCheck in the individual channel	
Value range	0..1	
Pre-assignment	0	

Unit	-
Description of the variable values	0 No check active 1 Checking of RCB() with backed up mark
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	-
Update	-
Dependency	-
Data type	Int_t (4 bytes)

Manual data input after block pre-run

Manual data input after block pre-run		ManWaitForStart
	Defines in the respective channel whether to wait for a manual data input for a manual adjustment after an Init adjustment program. If the manual data input is activated, the NC waits at the end of the Init adjustment program on the specification of NC blocks that should be processed with an NC start. If no NC blocks are specified, the adjustment is continued normally with an NC start.	
Path	/SysSRun[<Channel>]/ManWaitForStart CPL: SD.SysSRun[<Channel>].ManWaitForStart - or - SD.SysSRun.ManWaitForStart in the individual channel	
Value range	0..1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0 Deactivating manual data input 1 Activating manual data input After an Init adjustment program has been processed, NC blocks can be specified manually for adjustment when the manual data input is activated.	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	It is activated at the end of the Init adjustment program.	
Update	-	
Dependency	The function is used by the user interface to enable the manual data input via an operating function.	
Data type	Byte_t (1 byte)	

Saving program stack (callchain) at reset

Save program stack (call chain) at reset position		ReEntryStore
Defines in the corresponding channel whether the current program stack (call chain) is to be saved at an NC reset.		
The NC stores the current program stack in the file system if the program is active.		
The stored program stack can be subsequently used for a re-entry with block pre-run.		
Path	/SysSRun[<Channel>]/ReEntryStore CPL: SD.SysSRun[<Channel>].ReEntryStore - or - SD.SysSRun.ReEntryStore in the individual channel	
Value range	0..1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Deactivating storage of the program stack at reset
	1	Activating storage of the program stack at reset
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	Immediately	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

Suppressing message output in block pre-run

Suppressing message output in block pre-run		SuppressMsg
Defines whether the message output of MSG, GMSG and PRN should be suppressed.		
If messages are not suppressed, they are output in a fast sequence which might not be intended.		
The system date can be implemented into the part program to specifically suppress or output messages		
Path	/SysSRun[<Channel>]/SuppressMsg CPL: SD.SysSRun[<Channel>].SuppressMsg - or - SD.SysSRun.Timeout in the own channel	
Value range	0..1	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	Messages are output.
	1	Messages are suppressed.
Access rights	MTB:	Read and write

	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		Immediately
Update		-
Dependency		-
Data type		Int_t (4 bytes)

Computation active

Computation active	Active

Outdated and should not be used anymore. For a description, refer to / SysChSRun/Active. The use is still possible due to compatibility reasons.

Block pre-run active in look-ahead range after target block

Block pre-run active in look-ahead range after target block	LookAheadPreRun

Outdated and should not be used anymore. For a description, refer to / SysChSRun/LookAheadPreRun. The use is still possible due to compatibility reasons.

Target position of the approaching block

Target position of the approaching block	TargPos[1..8]

Outdated and should not be used anymore. For the description, refer to / SysChSRun/TargPos. The use is still possible due to compatibility reasons.

Locking target blocks

Lock target blocks	LockTargBlock

Outdated and should not be used anymore. For a description, refer to / SysChSRun/LockTargBlock. The use is still possible due to compatibility reasons.

Asup active

Asup active	AsupActive

Outdated and should not be used anymore. It is recommended to use SysChSRun/TargAdjPrgActive or SysChSRun/InitAdjPrgActive.

32.2.45 System status information "/SysStates"

Component operating time

Module operating time	OpTime

Cyclically prepares the operating time (in seconds) of the module.

If this date is overwritten, the controller restores the correct value in the next update cycle.

Path	/SysStates/BdTime/OpTime
	CPL: SD.SysStates.BdTime.OpTime
Value range	Integer range
Pre-assignment	0

System data

Unit	s
Description of the variable values	Module operating time
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	Cyclic (order of magnitude: seconds) by the NC kernel
Dependency	-
Data type	UnsignedInt_t (4 bytes)

Fan operating time

Fan operating time		FanTime
This parameter specifies the running time of the fan of the module in seconds. If this value approaches the MTBF value of the fan, it might be appropriate to install a new fan in the course of preventive maintenance measures. In this case, the counter should be reset to zero (→ /SysCtrl/SetFanTime). If this date is overwritten, the controller restores the correct value in the next update cycle.		
Path	/SysStates/BdTime/FanTime CPL: SD.SysStates.BdTime.FanTime	
Value range	Integer range	
Pre-assignment	0	
Unit	s	
Description of the variable values	Fan operating time	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By the NC kernel when the overtemperature warning is activated and deactivated.	
Dependency	-	
Data type	UnsignedInt_t (4 bytes)	

Current processor temperature

Current processor temperature	CpuTemp
This parameter specifies the current processor temperature provided the module is equipped with an option permitting measurement of the CPU temperature. If the module does not allow any temperature measurement or if an error is detected during the measurement, a value of -1.0 is entered here. In this case, the data item is not updated by the controller.	

	If this date is overwritten, the controller restores the correct value in the next update cycle.	
Path	/SysStates/BdTemp/CpuTemp CPL: SD.SysStates.BdTemp.CpuTemp	
Value range	Double range	
Pre-assignment	0	
Unit	°C	
Description of the variable values	-1.0	Module has no options for CPU temperature measurement
	Otherwise	Current CPU temperature
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	Cyclic (order of magnitude: seconds) by the NC kernel, but only if the module has temperature measurement	
Dependency	-	
Data type	Double_t (8 bytes)	

32.2.46 Data for cycle Swiveling G230 "/SysSwivel"

General information

The system data structure SysSwivel contains data used by the cycle G230 Swiveling as configuration values (e.g. retract position) as well as data of the active inclined plane (see "IndraMotion MTX Standard NC Cycles", chapter "G230 Swiveling").

As the system date has been created "permanently", the auxiliary cycle RestoreSwiveling can restore the last inclined plane even after an NC restart. In this case, only the placement is reactivated. No swiveling motion takes place.

If an inclined plane is swiveled, the cycle enters the calculated values in the system date. If the inclined plane is disabled again with "G230()", the cycle deletes the values from the system date. Therefore, valid values are only entered in the system date, if the cycle G230 has been swiveled to an inclined plane.

Valid values have to be entered in the substructure "Cfg" before calling G230 for the first time.

The following structural elements are defined in the "sdswivel.xsd" schema:

Structural element	Description
OX, OY, OZ	Zero point of the active incline plane or 0
phi, theta, psi	Euler angle of the active incline plane or 0
AX, AY, AZ	Additional zero point offset in the inclined plane or 0
Cfg	Substructure with configuration data of the G230 cycle

Tab. 32-37: Overview structural elements SysSwivel

The substructure "Cfg" contains the following elements:

Structural element	Description
RetractX, RetractY, RetractZ	Retract position in absolute machine coordinates
RetractF	Retract feed (0 for rapid traverse)
TNR	Default value for parameter TNR (if not programmed during the cycle call)

Tab. 32-38: Overview of structural elements SysSwivel.Cfg

Zero point of the active inclined plane

Zero point of the active inclined plane	OX/OY/OZ

Returns the zero point of the active inclined plane channel-specifically.

Path /SysSwivel[<Channel>]/OX
 /SysSwivel[<Channel>]/OY
 /SysSwivel[<Channel>]/OZ
 CPL: SD.SysSwivel[<Channel>].OX
 CPL: SD.SysSwivel[<Channel>].OY
 CPL: SD.SysSwivel[<Channel>].OZ

Value range -

Pre-assignment 0

Unit mm or inch

Description of the variable values Channel-specifically returns the zero point of the active inclined plane if it has been previously swiveled to the inclined plane with cycle G230.

Access rights **MTB:** Read
Setter: Read
User: Read
NC prog.: Read
PLC prog.: Read

Activation -

Update By NC kernel or cycle G230.

Dependency -

Data type 3 x Double_t (3 x 8 Bytes)

Euler angle of the active inclined plane

Euler angle of the active inclined plane	phi/theta/psi

Returns the Euler angle of the active inclined plane channel-specifically.

Path /SysSwivel[<Channel>]/phi
 /SysSwivel[<Channel>]/theta
 /SysSwivel[<Channel>]/psi
 CPL: SD.SysSwivel[<Channel>].phi
 CPL: SD.SysSwivel[<Channel>].theta
 CPL: SD.SysSwivel[<Channel>].psi

Value range -

Pre-assignment	0
Unit	Degree
Description of the variable values	Channel-specifically returns the Euler angle of the active inclined plane if it has been previously swiveled to the inclined plane with cycle G230.
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	By NC kernel or cycle G230.
Dependency	-
Data type	3 x Double_t (3 x 8 Bytes)

Additional offset of the zero point in the inclined plane

Additional zero point offset in the inclined plane		AX/AY/AZ
Channel-specifically returns the additional zero point offset of the active in- cline plane.		
Path	/SysSwivel[<Channel>]/AX /SysSwivel[<Channel>]/AY /SysSwivel[<Channel>]/AZ CPL: SD.SysSwivel[<Channel>].AX CPL: SD.SysSwivel[<Channel>].AY CPL: SD.SysSwivel[<Channel>].AZ	
Value range	-	
Pre-assignment	0	
Unit	mm or inch	
Description of the variable values	Channel-specifically returns the additional zero point offset of the active in- clined plane if it has been previously swiveled to the inclined plane with cycle G230.	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By NC kernel or cycle G230.	
Dependency	-	
Data type	3 x Double_t (3 x 8 Bytes)	

Retract position in absolute machine coordinates

Retract position in absolute machine coordinates	Cfg.RetractX/ RetractY/RetractZ
Channel-specifically configures the retract position in absolute machine coor- dinates for cycle G230.	

System data

Path	/SysSwivel[<Channel>]/Cfg/RetractX /SysSwivel[<Channel>]/Cfg/RetractY /SysSwivel[<Channel>]/Cfg/RetractZ CPL: SD.SysSwivel[<Channel>].Cfg.RetractX CPL: SD.SysSwivel[<Channel>].Cfg.RetractY CPL: SD.SysSwivel[<Channel>].Cfg.RetractZ
Value range	-
Pre-assignment	0
Unit	mm or inch
Description of the variable values	Channel-specifically configures the retract position in absolute machine coordinates for cycle G230. This position is approached with active G76 during retracting of cycle G230.
Access rights	MTB: Read, write Setter: Read, write User: Read NC prog.: Read, write PLC prog.: Read, write
Activation	-
Update	-
Dependency	-
Data type	3 x Double_t (3 x 8 Bytes)

Retract feed (0 for rapid traverse)

Retract feed (0 for rapid traverse)		Cfg.RetractF
Channel-specifically configures the retract position for the cycle G230. If the value 0 has been selected, retracting takes place in rapid traverse.		
Path	/SysSwivel[<Channel>]/Cfg/RetractF CPL: SD.SysSwivel[<Channel>].Cfg.RetractF	
Value range	≥0	
Pre-assignment	0	
Unit	-	
Description of the variable values	Channel-specifically configures the retract position for the cycle G230. The configured feed is used to travel to the retract position.	
Access rights	MTB: Read, write Setter: Read, write User: Read NC prog.: Read, write PLC prog.: Read, write	
Activation	-	
Update	-	
Dependency	-	
Data type	Double_t (8 bytes)	

Default value for parameter TNR

Default value for parameter TNR		Cfg.TNR
	Channel-specifically configures the default value for parameter TNR if the parameter has not been programmed in cycle G230.	
Path	/SysSwivel[<Channel>]/Cfg/TNR CPL: SD.SysSwivel[<Channel>].Cfg.TNR	
Value range	0-20	
Pre-assignment	0	
Unit	-	
Description of the variable values	When calling cycle G230, specify the path of the kinematic data of the machine. Program parameter TNR, linking to the axis transformation data. As usually the same axis transformation number is used at a machine, the number can also be configured in the system date. If the cycle G230 is called without the parameters TNR and TSD, the kinematic data referring to the system date is used.	
Access rights	MTB: Read, write Setter: Read, write User: Read NC prog.: Read, write PLC prog.: Read, write	
Activation	-	
Update	-	
Dependency	-	
Data type	Byte_t (1 byte)	

32.2.47 Target information "/SysTargetInfo"

General information

The following structural elements are defined in the "sys_sd_opdta.xsd" schema:

Structural element	Description
Eth1/IpAddr	IP address of the first Ethernet interface
Eth1/SubNetMask	Network mask of the first Ethernet interface
Eth1/GatewayAddr	IP address of the gateway of the first Ethernet interface
Eth1/MACAddr	MAC address of the first Ethernet interface
Eth1/Speed	Velocity setting of the first Ethernet interface
Eth2/...	Same data for the second Ethernet interface
FW/SysRel	Firmware string of the control
FW/KnsRel	KNS version string
FW/KnbRel	KNB version string
HW/Type	Control type
HW/LpNmb	Circuit board number
HW/Version	Hardware version

Structural element	Description
HW/SerNmb	Serial number
HW/BoardFPGA	Board FPGA version
Sercos1/Version	Version Sercos connection 1
Sercos1/Type	Type Sercos connection 1
Sercos1/FPGA	FPGA version Sercos connection 1
Sercos2/...	Same data for Sercos connection 2 (optional function module)

Tab. 32-39: Overview on structural elements "SysTargetInfo"

Configuration of the first or second Ethernet interface

IP address	Eth(1,2)/IpAddr

Returns the IP address of the first or second Ethernet interface.

Path /SysTargetInfo/Eth(1,2)/IpAddr

Value range -

Pre-assignment <EmptyString>

Unit -

Description of parameter values The IP address of the first or second Ethernet interface.

Access rights

MTB:	Read (read-only)
Setter:	Read (read-only)
Operator:	Read (read-only)
NC prog.:	Read (read-only)
PLC prog.:	Read (read-only)

Activation -

Update By the NC kernel at startup

Dependency -

Data type IsoLatin1String (40 bytes)

Configuration of the network mask of the first or second Ethernet interface

SubNetMask	Eth(1,2)/SubNet-Mask

Returns the network mask of the first or second Ethernet interface.

Path /SysTargetInfo/Eth(1,2)/SubNetMask

Value range -

Pre-assignment <EmptyString>

Unit -

Description of parameter values The network mask of the first or second Ethernet interface.

Access rights

MTB:	Read (read-only)
Setter:	Read (read-only)
Operator:	Read (read-only)
NC prog.:	Read (read-only)
PLC prog.:	Read (read-only)

Activation	-
Update	By the NC kernel at startup
Dependency	-
Data type	ISO Latin1String (40 bytes)

Configuration of the gateway of the first or second Ethernet interface

GatewayAddr		Eth(1,2)/GatewayAddr
Returns the IP address of the gateway of the first or second Ethernet interface.		
Path	/SysTargetInfo/Eth(1,2)/GatewayAddr	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	The IP address of the gateway of the first or second Ethernet interface.	
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	ISO Latin1String (40 bytes)	

Configuration of the MAC address of the first or second Ethernet interface

MACAddr		Eth(1,2)/MACAddr
Returns the MAC address of the first or second Ethernet interface.		
Path	/SysTargetInfo/Ethx/MACAddr	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	The MAC address of the first or second Ethernet interface.	
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	ISO Latin1String (40 bytes)	

Configuration of the velocity of the first or second Ethernet interface

Speed		Eth(1,2)/Speed
Returns the velocity of the first or second Ethernet interface.		
Path	/SysTargetInfo/Eth(1,2)/Speed	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Velocity of the first or second Ethernet interface.	
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	IsoLatin1String (40 bytes)	

Active firmware release

SysRel		FW/SysRel
Returns the firmware string of the control.		
Path	/SysTargetInfo/FW/SysRel	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Firmware string of the control.	
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	IsoLatin1String (64 bytes)	

Active KNS version string

KnsRel		FW/KnsRel
Returns the KNS version string.		
Path	/SysTargetInfo/FW/KnsRel	
Value range	-	
Pre-assignment	<EmptyString>	

Unit	-
Description of parameter values	KNS version string.
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)
Activation	-
Update	By the NC kernel at startup
Dependency	-
Data type	IsoLatin1String (64 bytes)

Active KNB version string

KnRel		FW/KnbRel
Returns the KNB version string.		
Path	/SysTargetInfo/FW/KnbRel	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	KNB version string.	
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	IsoLatin1String (64 bytes)	

Control type

Type		HW/Type
Returns the control hardware type.		
Path	/SysTargetInfo/HW/Type	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Control type.	
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)	

System data

Activation	-
Update	By the NC kernel at startup
Dependency	-
Data type	IsoLatin1String (16 bytes)

Printed circuit board number

LpNmb		HW/LpNmb
It provides the printed circuit board number of the control hardware.		
Path	/SysTargetInfo/HW/LpNmb	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Printed circuit board number.	
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	IsoLatin1String (16 bytes)	

Hardware version

Version		HW/Version
Returns the control hardware version.		
Path	/SysTargetInfo/HW/Version	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Hardware version.	
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	IsoLatin1String (16 bytes)	

Serial number of the control hardware

SerNmb	HW/SerNmb
Returns the serial number of the control hardware.	

Path	/SysTargetInfo/HW/SerNmb
Value range	-
Pre-assignment	<EmptyString>
Unit	-
Description of parameter values	Serial number of the control hardware.
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)
Activation	-
Update	By the NC kernel at startup
Dependency	-
Data type	IsoLatin1String (16 bytes)

Board FPGA version

BoardFPGA		HW/BoardFPGA
Returns the version of the board FPGA.		
Path	/SysTargetInfo/HW/BoardFPGA	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Board FPGA version.	
Access rights	MTB: Read (read-only) Setter: Read (read-only) Operator: Read (read-only) NC prog.: Read (read-only) PLC prog.: Read (read-only)	
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	IsoLatin1String (16 bytes)	

Version of the Sercos connection 1 or 2

Version		Sercos(1;2)/Version
Returns the version of the Sercos connection 1 or 2.		
Path	/SysTargetInfo/Sercos(1,2)/Version	
Value range	No Value, Sercos	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Version of the Sercos connection.	
Access rights	MTB: Read (read-only) Setter: Read (read-only)	

System data

	Operator:	Read (read-only)
	NC prog.:	Read (read-only)
	PLC prog.:	Read (read-only)
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	IsoLatin1String (16 bytes)	

Type of the Sercos connection 1 or 2

Type		Sercos(1;2)/Type
Returns the type of the Sercos connection 1 or 2.		
Path	/SysTargetInfo/Sercos(1,2)/Type	
Value range	No value, master, slave	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	Type of the Sercos connection.	
Access rights	MTB:	Read (read-only)
	Setter:	Read (read-only)
	Operator:	Read (read-only)
	NC prog.:	Read (read-only)
	PLC prog.:	Read (read-only)
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	IsoLatin1String (16 bytes)	

FPGA version of the Sercos connection 1 or 2

FPGA		Sercos(1;2)/FPGA
Returns the FPGA version of the Sercos connection 1 or 2.		
Path	/SysTargetInfo/Sercos(1,2)/FPGA	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of parameter values	FGPA version of the Sercos connection.	
Access rights	MTB:	Read (read-only)
	Setter:	Read (read-only)
	Operator:	Read (read-only)
	NC prog.:	Read (read-only)
	PLC prog.:	Read (read-only)
Activation	-	
Update	By the NC kernel at startup	
Dependency	-	
Data type	IsoLatin1String (16 bytes)	

32.2.48 Tool correction tool edge numbers"/SysTlActMaxEd"

Preselected tool edge number

	Preselected tool edge number	ActEd
	Provides for each channel the number of the preselected edge in the preselected tool. Write access occurs only with the NC kernel and is triggered by programming NC command ED. The preselected tool becomes actually active only when the NC block that is being prepared starts to be processed.	
Path	/SysTlActMaxEd[<Channel>]/ActEd	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Preselected tool edge number in the channel	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel when processing the NC command ED	
Dependency	-	
Data type	Int_t (4 bytes)	

Maximum tool edge number

	Maximum tool edge number	MaxEd
	Provides for each channel the maximum number of tool edges for the tool corrections in the system (/SysTlEd[n]), where the tool edge register display is to come.	
Path	/SysTlActMaxEd[<Channel>]/MaxEd	
Value range	-	
Pre-assignment	2	
Unit	-	
Description of the variable values	Maximum number of tool edges for the tool edge register display	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	Int_t (4 bytes)	

32.2.49 Active and preselected tool "/SysTool"

General information

This permanent and channel-specific system data is provided for the tool management. Apart for the element "ActEd", the NC core neither writes nor reads the system data.

Element "ActEd" allows read-only access on the number of the active edge. Write access occurs only with the NC kernel and is triggered by programming NC command ED.

The active "edge" is the edge active for the block preparation. This edge becomes active only if the NC block currently in preparation is being processed.

The structural elements described are defined in the "sdtlcorr.xsd" schema.

Structural element	Description
ActToolValid	Validity of the active tool
ActEd	Number of the active edge
ActTool.K1	Tool data element K1 of active tool
ActTool.K2	Tool data element K2 of active tool
ActTool.SKQ	Tool data element SKQ of the active tool
ActTool.IKQ1	Tool data element IKQ1 of the active tool
ActTool.IKQ2	Tool data element IKQ2 of the active tool
ActTool.IKQ3	Tool data element IKQ3 of the active tool
ActTool.IQ1	Tool data element IQ1 of the active tool
ActTool.IQ2	Tool data element IQ2 of the active tool
ActTool.IQ3	Tool data element IQ3 of the active tool
ActTool.BQ1	Tool data element BQ1 of the active tool
ActTool.BQ2	Tool data element BQ2 of the active tool
ActTool.BQ3	Tool data element BQ3 of the active tool
PreTool.K1	Tool data element K1 of the prepared tool
PreTool.K2	Tool data element K2 of the prepared tool
PreTool.SKQ	Tool data element SKQ of the prepared tool
PreTool.IKQ1	Tool data element IKQ1 of the prepared tool
PreTool.IKQ2	Tool data element IKQ2 of the prepared tool
PreTool.IKQ3	Tool data element IKQ3 of the prepared tool
PreTool.IQ1	Tool data element IQ1 of the prepared tool
PreTool.IQ2	Tool data element IQ2 of the prepared tool
PreTool.IQ3	Tool data element IQ3 of the prepared tool
PreTool.BQ1	Tool data element BQ1 of the prepared tool
PreTool.BQ2	Tool data element BQ2 of the prepared tool
PreTool.BQ3	Tool data element BQ3 of the prepared tool

Tab. 32-40: Overview on structural elements "SysTool"

Validity of the active tool

Validity of the active tool		ActToolValid
	Indicates whether the active tool is valid.	
Path	/SysTool[<Channel>]/ActToolValid CPL: SD.SysTool[<Channel>].ActToolValid	
Value range	0 (FALSE) 1 (TRUE)	
Pre-assignment	0 (FALSE)	
Unit	-	
Description of the variable values	TRUE: Active tool is valid	
Access rights	MTB: Read and write	
	Setter: Read and write	
	User: Read and write	
	NC prog.: Read and write	
	PLC prog.: Read and write	
Activation	-	
Update	-	
Dependency	-	
Data type	Boolean_t (1 byte)	

Number of the active edge

Number of the active edge		ActEd
	Number of the active edge.	
Path	/SysTool[<Channel>]/ActEd CPL: SD.SysTool[<Channel>].ActEd	
Value range	0..16	
Pre-assignment	0	
Unit	-	
Description of the variable values	0 No edge active	
	1..16 Number of the active edge	
Access rights	MTB: Read	
	Setter: Read	
	User: Read	
	NC prog.: Read	
	PLC prog.: Read	
Activation	-	
Update	By the NC kernel	
Dependency	-	
Data type	Int_t (4 bytes)	

ActTool.K1

ActTool.K1	ActTool.K1
Tool data element K1 of active tool	

System data

Path	/SysTool[<Channel>]/ActTool/K1	
	CPL: SD.SysTool[<Channel>]/ActTool.K1	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	-	
Dependency	-	
Data type	xs:short (2 bytes)	

ActTool.K2

ActTool.K2		ActTool.K2
Tool data element K2 of active tool		
Path	/SysTool[<Channel>]/ActTool/K2	
	CPL: SD.SysTool[<Channel>].ActTool.K2	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	-	
Dependency	-	
Data type	xs:short (2 bytes)	

ActTool.SKQ

ActTool.SKQ		ActTool.SKQ
Tool data element SKQ of the active tool		
Path	/SysTool[<Channel>]/ActTool/SKQ	
	CPL: SD.SysTool[<Channel>].ActTool.SKQ	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	

Description of the variable values

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	isoStr31_t (31 bytes + 1 byte for end identification)	

ActTool.IKQ1

ActTool.IKQ1	ActTool.IKQ1

Tool data element IKQ1 of the active tool

Path /SysTool[<Channel>]/ActTool/IKQ1
CPL: SD.SysTool[<Channel>].ActTool.IKQ1

Value range -
Pre-assignment 0
Unit -

Description of the variable values

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

ActTool.IKQ2

ActTool.IKQ2	ActTool.IKQ2

Tool data element IKQ2 of the active tool

Path /SysTool[<Channel>]/ActTool/IKQ2
CPL: SD.SysTool[<Channel>].ActTool.IKQ2

Value range -
Pre-assignment 0
Unit -

Description of the variable values

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write

System data

	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

ActTool.IKQ3

ActTool.IKQ3	ActTool.IKQ3

Tool data element IKQ3 of the active tool

Path /SysTool[<Channel>]/ActTool/IKQ3

CPL: SD.SysTool[<Channel>].ActTool.IKQ3

Value range -**Pre-assignment** 0**Unit** -

Description of the variable values

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write

Activation -**Update** -**Dependency** -**Data type** xs:int (4 bytes)

ActTool.IQ1

ActTool.IQ1	ActTool.IQ1

Tool data element IQ1 of the active tool

Path /SysTool[<Channel>]/ActTool/IQ1

CPL: SD.SysTool[<Channel>].ActTool.IQ1

Value range -**Pre-assignment** 0**Unit** -

Description of the variable values

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write

Activation -**Update** -**Dependency** -**Data type** xs:int (4 bytes)

ActTool.IQ2

ActTool.IQ2		ActTool.IQ2
		Tool data element IQ2 of the active tool
Path		/SysTool[<Channel>]/ActTool/IQ2 CPL: SD.SysTool[<Channel>].ActTool.IQ2
Value range		-
Pre-assignment		0
Unit		-
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		-
Update		-
Dependency		-
Data type		xs:int (4 bytes)

ActTool.IQ3

ActTool.IQ3		ActTool.IQ3
		Tool data element IQ3 of the active tool
Path		/SysTool[<Channel>]/ActTool/IQ3 CPL: SD.SysTool[<Channel>].ActTool.IQ3
Value range		-
Pre-assignment		0
Unit		-
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		-
Update		-
Dependency		-
Data type		xs:int (4 bytes)

ActTool.BQ1

ActTool.BQ1		ActTool.BQ1
		Tool data element BQ1 of the active tool
Path		/SysTool[<Channel>]/ActTool/BQ1 CPL: SD.SysTool[<Channel>]/ActTool/BQ1

System data

Value range	-
Pre-assignment	0
Unit	-
Description of the variable values	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	-
Update	-
Dependency	-
Data type	xs:int (4 bytes)

ActTool.BQ2

ActTool.BQ2		ActTool.BQ2
Tool data element BQ2 of the active tool		
Path	/SysTool[<Channel>]/ActTool/BQ2 CPL: SD.SysTool[<Channel>].ActTool.BQ2	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

ActTool.BQ3

ActTool.BQ3		ActTool.BQ3
Tool data element BQ3 of the active tool		
Path	/SysTool[<Channel>]/ActTool/BQ3 CPL: SD.SysTool[<Channel>].ActTool.BQ3	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read and write

	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

PreTool.K1

PreTool.K1		PreTool.K1
	Tool data element K1 of the prepared tool	
Path	/SysTool[<Channel>]/PreTool/K1 CPL: SD.SysTool[<Channel>].PreTool.K1	
Value range	-	
Pre-assignment	0	
Unit	-	
e variable values		
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	-	
Dependency	-	
Data type	xs:short (2 bytes)	

PreTool.K2

PreTool.K2		PreTool.K2
	Tool data element K2 of the prepared tool	
Path	/SysTool[<Channel>]/PreTool/K2 CPL: SD.SysTool[<Channel>].PreTool.K2	
Value range	-	
Pre-assignment	0	
Unit	-	
e variable values		
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	

System data

Update -
 Dependency -
 Data type xs:short (2 bytes)

PreTool.SKQ

PreTool.SKQ	PreTool.SKQ

Tool data element SKQ of the prepared tool

Path /SysTool[<Channel>]/PreTool/SKQ
 CPL: SD.SysTool[<Channel>].PreTool.SKQ

Value range -
 Pre-assignment <EmptyString>
 Unit -

Description of the variable values

Access rights MTB: Read and write
 Setter: Read and write
 User: Read and write
 NC prog.: Read and write
 PLC prog.: Read and write

Activation -
 Update -
 Dependency -
 Data type isoStr31_t (31 bytes + 1 byte for end identification)

PreTool.IKQ1

PreTool.IKQ1	PreTool.IKQ1

Tool data element IKQ1 of the prepared tool

Path /SysTool[<Channel>]/PreTool/IKQ1
 CPL: SD.SysTool[<Channel>]/PreTool/IKQ1

Value range -
 Pre-assignment 0
 Unit -

Description of the variable values

Access rights MTB: Read and write
 Setter: Read and write
 User: Read and write
 NC prog.: Read and write
 PLC prog.: Read and write

Activation -
 Update -
 Dependency -
 Data type xs:int (4 bytes)

PreTool.IKQ2

PreTool.IKQ2		PreTool.IKQ2
		Tool data element IKQ2 of the prepared tool
Path		/SysTool[<Channel>]/PreTool/IKQ2 CPL: SD.SysTool[<Channel>].PreTool.IKQ2
Value range		-
Pre-assignment		0
Unit		-
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		-
Update		-
Dependency		-
Data type		xs:int (4 bytes)

PreTool.IKQ3

PreTool.IKQ3		PreTool.IKQ3
		Tool data element IKQ3 of the prepared tool
Path		/SysTool[<Channel>]/PreTool/IKQ3 CPL: SD.SysTool[<Channel>].PreTool.IKQ3
Value range		-
Pre-assignment		0
Unit		-
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		-
Update		-
Dependency		-
Data type		xs:int (4 bytes)

PreTool.IQ1

PreTool.IQ1	PreTool.IQ1
	Tool data element IQ1 of the prepared tool
Path	/SysTool[<Channel>]/PreTool/IQ1 CPL: SD.SysTool[<Channel>].PreTool.IQ1

System data

Value range	-
Pre-assignment	0
Unit	-
Description of the variable values	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	-
Update	-
Dependency	-
Data type	xs:int (4 bytes)

PreTool.IQ2

PreTool.IQ2		PreTool.IQ2
Tool data element IQ2 of the prepared tool		
Path	/SysTool[<Channel>]/PreTool/IQ2 CPL: SD.SysTool[<Channel>].PreTool.IQ2	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

PreTool.IQ3

PreTool.IQ3		PreTool.IQ3
Tool data element IQ3 of the prepared tool		
Path	/SysTool[<Channel>]/PreTool/IQ3 CPL: SD.SysTool[<Channel>].PreTool.IQ3	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read

Setter: Read
User: Read
NC prog.: Read
PLC prog.: Read

Activation -
Update -
Dependency -
Data type xs:int (4 bytes)

PreTool.BQ1

PreTool.BQ1	PreTool.BQ1

Tool data element BQ1 of the prepared tool

Path /SysTool[<Channel>]/PreTool/BQ1

CPL: SD.SysTool[<Channel>].PreTool.BQ1

Value range -

Pre-assignment 0

Unit -

Description of the variable values

Access rights **MTB:** Read and write
Setter: Read and write
User: Read and write
NC prog.: Read and write
PLC prog.: Read and write

Activation -

Update -

Dependency -

Data type xs:int (4 bytes)

PreTool.BQ2

PreTool.BQ2	PreTool.BQ2

Tool data element BQ2 of the prepared tool

Path /SysTool[<Channel>]/PreTool/BQ2

CPL: SD.SysTool[<Channel>].PreTool.BQ2

Value range -

Pre-assignment 0

Unit -

Description of the variable values

Access rights **MTB:** Read and write
Setter: Read and write
User: Read and write
NC prog.: Read and write
PLC prog.: Read and write

Activation -

Update -
 Dependency -
 Data type xs:int (4 bytes)

PreTool.BQ3

	PreTool.BQ3	PreTool.BQ3
	Tool data element BQ3 of the prepared tool	
Path	/SysTool[<Channel>]/PreTool/BQ3 CPL: SD.SysTool[<Channel>].PreTool.BQ3	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

32.2.50 Tool Corrections per edge "/SysToolEd"

General information

The system date SysToolEd(<i>) permits read access to the tool correction data of edge <i> of the active tool in the current channel.

The system date SysToolEd(<i>,<Channel>) permits read access to the tool correction data of edge <i> of the active tool in the channel <Channel>.

Write access occurs only with the NC kernel and only when tool correction data is written by the PLC or using CPL command "DCT".

The active "tool" is the tool active for the block preparation. This tool becomes active only if the NC block currently in preparation is being processed.

The structural elements described here are stored in the "sdtlcorr.xsd" schema.

Structural element	Description
L1	L1 correction length
L2	L2 correction length
L3	L3 correction length
R	Edge radius
Ori	Edge position
Phi	Euler angle φ

Structural element	Description
Theta	Euler angle θ
Psi	Euler angle ψ

Tab. 32-41: Overview on structural elements "SysToolEd"

Length correction L1, L2, L3

Length correction L1, L2, L3	L1
	L2
	L3

Provides for each channel the desired length correction of the preselected tool for the respective edge number. Write access occurs only with the NC kernel and only when tool correction data is written by the PLC or using CPL command "DCT".

The preselected tool becomes actually active only when the NC block that is being prepared starts to be processed.

Path /SysToolEd[1..16,<Channel>]/L1
 /SysToolEd[1..16,<Channel>]/L2
 /SysToolEd[1..16,<Channel>]/L3
 CPL: SD.SysToolEd[1..16,<Channel>].L1
 CPL: SD.SysToolEd[1..16,<Channel>].L2
 CPL: SD.SysToolEd[1..16,<Channel>].L3

Value range -

Pre-assignment 0

Unit mm

Description of the variable values Length correction

Access rights **MTB:** Read
Setter: Read
User: Read
NC prog.: Read
PLC prog.: Read

Activation -

Update By NC kernel after changing the tool correction data via the PLC or the CPL command "DCT".

Dependency -

Data type Float_t (4 bytes)

Edge radius correction R

Edge radius correction R	R

Provides for each channel the edge radius correction of the preselected tool for the respective edge number. Write access occurs only with the NC kernel and only when tool correction data is written by the PLC or using CPL command "DCT".

The preselected tool becomes actually active only when the NC block that is being prepared starts to be processed.

System data

Path	/SysTIEd[1..16,<Channel>]/R CPL: SD.SysToolEd[1..16,<Channel>].R
Value range	-
Pre-assignment	0
Unit	mm
Description of the variable values	Edge radius correction
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	-
Update	By NC kernel after changing the tool correction data via the PLC or the CPL command "DCT".
Dependency	-
Data type	Float_t (4 bytes)

Tool edge position Ori

Edge position Ori		Ori
Provides for each channel the edge position of the preselected tool for the respective edge number. Write access occurs only with the NC kernel and only when tool correction data is written by the PLC or using CPL- command DCT. The preselected tool becomes actually active only when the NC block that is being prepared starts to be processed.		
Path	/SysTIEd[1..16,<Channel>]/Ori CPL: SD.SysToolEd[1..16,<Channel>].Ori	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Edge position	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	By NC kernel after changing the tool correction data via the PLC or the CPL command "DCT".	
Dependency	-	
Data type	Int_t (4 bytes)	

Euler angle Phi, Theta, Psi

Euler angle φ , θ , ψ	Phi Theta Psi

It provides for each channel the desired Euler angle (φ , θ , ψ) of the preselected tool for the respective edge number. Write access occurs only with the NC kernel and only when tool correction data is written by the PLC or using CPL- command DCT.

The preselected tool becomes actually active only when the NC block that is being prepared starts to be processed.

Path /SysTIEd[1..16,<Channel>]/Phi
 /SysTIEd[1..16,<Channel>]/Theta
 /SysTIEd[1..16,<Channel>]/Psi
 CPL: SD.SysToolEd[1..16,<Channel>].Phi
 CPL: SD.SysToolEd[1..16,<Channel>].Theta
 CPL: SD.SysToolEd[1..16,<Channel>].Psi

Value range -

Pre-assignment 0

Unit °

Description of the variable values Euler Angles

Access rights **MTB:** Read
Setter: Read
User: Read
NC prog.: Read
PLC prog.: Read

Activation -

Update By NC kernel after changing the tool correction data via the PLC or the CPL command "DCT".

Dependency -

Data type Float_t (4 bytes)

32.2.51 SysToolInsert "/SysToolInsert"

General information

This volatile system data is provided for the tool management.

The structural elements described are defined in the schemas "dbt1sd.xsd" and "dbt1ud.xsd".

Structural element	Description
Hd	Tool data head of type DBT1Hd_t (see scheme "dbt1sd.xsd") The data structure below the "Hd" node is the same as the one of the system variable "/SysTool" below the path /ActTool.
UD	User data of the tool data of type DBT1UD_t (see scheme "dbt1ud.xsd")
UD.TI	Data structure of type "TI_V09_t" (see scheme "dbt1ud.xsd")
UD.Ed	Data structure of type "Ed_V09_t" (see scheme "dbt1ud.xsd")

Tab. 32-42: Overview on structural elements "SysToolInsert"

32.2.52 Active and preselected tool at block pre-run "/SysToolSRun"

General information

This volatile and channel-specific system data is provided for the tool management at block pre-run. Using the data, the respective actual and command values of the tool management can be adjusted in an adjustment program.

The elements "ActTool" and "PreTool" are used for block pre-run in the tool management. They have to be used by the respective tool management and are thus exclusively available for the user program. The actual tool change should store the information on the active or preselected tool in the user program.

The elements "TargEd" or "TargDnr" allow read-only access on the number of the active edge or the number of the D-correction in the target block and are written after successful computation by the NC kernel.

The element "TCorrNotAdj" allows read-only access on the adjustment status of the tool correction and is written after successful computation by the NC kernel or updated in the respective adjustment process.

The structural elements described are defined in the "sdtlcorr.xsd" schema.

Structure element	Description
ActTool.K1	Tool data element K1 of the active tool
ActTool.K2	Tool data element K2 of the active tool
ActTool.SKQ	Tool data element SKQ of the active tool
ActTool.IKQ1	Tool data element IKQ1 of the active tool
ActTool.IKQ2	Tool data element IKQ2 of the active tool
ActTool.IKQ3	Tool data element IKQ3 of the active tool
ActTool.IQ1	Tool data element IQ1 of the active tool
ActTool.IQ2	Tool data element IQ2 of the active tool
ActTool.IQ3	Tool data element IQ3 of the active tool
ActTool.BQ1	Tool data element BQ1 of the active tool
ActTool.BQ2	Tool data element BQ2 of the active tool

Structure element	Description
ActTool.BQ3	Tool data element BQ3 of the active tool
PreTool.K1	Tool data element K1 of the prepared tool
PreTool.K2	Tool data element K2 of the prepared tool
PreTool.SKQ	Tool data element SKQ of the prepared tool
PreTool.IKQ1	Tool data element IKQ1 of the prepared tool
PreTool.IKQ2	Tool data element IKQ2 of the prepared tool
PreTool.IKQ3	Tool data element IKQ3 of the prepared tool
PreTool.IQ1	Tool data element IQ1 of the prepared tool
PreTool.IQ2	Tool data element IQ2 of the prepared tool
PreTool.IQ3	Tool data element IQ3 of the prepared tool
PreTool.BQ1	Tool data element BQ1 of the prepared tool
PreTool.BQ2	Tool data element BQ2 of the prepared tool
PreTool.BQ3	Tool data element BQ3 of the prepared tool
TargEd	Number of the active edge in the target block
TargDNr	Number of the active D correction in the target block
TCorrNotAdj	Tool correction not adjusted

Tab. 32-43: Overview on the structure elements "SysToolSRun"

ActTool.K1

ActTool.K1	ActTool.K1

Tool data element K1 of active tool

Path /SysToolSRun[<Channel>]/ActTool/K1

CPL: SD.SysToolSRun[<Channel>]/ActTool.K1

Value range -**Pre-assignment** 0**Unit** -**Description of the variable values****Access rights** **MTB:** Read**Setter:** Read**User:** Read**NC prog.:** Read**PLC prog.:** Read**Activation** -**Update** -**Dependency** -**Data type** xs:short (2 bytes)**ActTool.K2**

ActTool.K2	ActTool.K2

Tool data element K2 of active tool

System data

Path	/SysToolSRun[<Channel>]/ActTool/K2 CPL: SD.SysToolSRun[<Channel>].ActTool.K2		
Value range	-		
Pre-assignment	0		
Unit	-		
Description of the variable values			
Access rights	MTB:	Read	
	Setter:	Read	
	User:	Read	
	NC prog.:	Read	
	PLC prog.:	Read	
Activation	-		
Update	-		
Dependency	-		
Data type	xs:short (2 bytes)		

ActTool.SKQ

ActTool.SKQ		ActTool.SKQ
Tool data element SKQ of the active tool		
Path	/SysToolSRun[<Channel>]/ActTool/SKQ CPL: SD.SysToolSRun[<Channel>].ActTool.SKQ	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	isoStr31_t (31 bytes + 1 byte for end identification)	

ActTool.IKQ1

ActTool.IKQ1		ActTool.IKQ1
Tool data element IKQ1 of the active tool		
Path	/SysToolSRun[<Channel>]/ActTool/IKQ1 CPL: SD.SysToolSRun[<Channel>].ActTool.IKQ1	
Value range	-	
Pre-assignment	0	
Unit	-	

Description of the variable values

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

ActTool.IKQ2

ActTool.IKQ2	ActTool.IKQ2

Tool data element IKQ2 of the active tool

Path	/SysToolSRun[<Channel>]/ActTool/IKQ2
	CPL: SD.SysToolSRun[<Channel>].ActTool.IKQ2
Value range	-
Pre-assignment	0
Unit	-

Description of the variable values

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

ActTool.IKQ3

ActTool.IKQ3	ActTool.IKQ3

Tool data element IKQ3 of the active tool

Path	/SysToolSRun[<Channel>]/ActTool/IKQ3
	CPL: SD.SysToolSRun[<Channel>].ActTool.IKQ3
Value range	-
Pre-assignment	0
Unit	-

Description of the variable values

Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write

System data

	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

ActTool.IQ1

ActTool.IQ1	ActTool.IQ1

	Tool data element IQ1 of the active tool	
Path	/SysToolSRun[<Channel>]/ActTool/IQ1	
	CPL: SD.SysToolSRun[<Channel>].ActTool.IQ1	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

ActTool.IQ2

ActTool.IQ2	ActTool.IQ2

	Tool data element IQ2 of the active tool	
Path	/SysToolSRun[<Channel>]/ActTool/IQ2	
	CPL: SD.SysToolSRun[<Channel>].ActTool.IQ2	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

ActTool.IQ3

ActTool.IQ3		ActTool.IQ3
		Tool data element IQ3 of the active tool
Path		/SysToolSRun[<Channel>]/ActTool/IQ3 CPL: SD.SysToolSRun[<Channel>].ActTool.IQ3
Value range		-
Pre-assignment		0
Unit		-
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		-
Update		-
Dependency		-
Data type		xs:int (4 bytes)

ActTool.BQ1

ActTool.BQ1		ActTool.BQ1
		Tool data element BQ1 of the active tool
Path		/SysToolSRun[<Channel>]/ActTool/BQ1 CPL: SD.SysToolSRun[<Channel>]/ActTool/BQ1
Value range		-
Pre-assignment		0
Unit		-
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		-
Update		-
Dependency		-
Data type		xs:int (4 bytes)

ActTool.BQ2

ActTool.BQ2		ActTool.BQ2
		Tool data element BQ2 of the active tool
Path		/SysToolSRun[<Channel>]/ActTool/BQ2 CPL: SD.SysToolSRun[<Channel>].ActTool.BQ2

System data

Value range	-
Pre-assignment	0
Unit	-
Description of the variable values	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	-
Update	-
Dependency	-
Data type	xs:int (4 bytes)

ActTool.BQ3

ActTool.BQ3		ActTool.BQ3
Tool data element BQ3 of the active tool		
Path	/SysToolSRun[<Channel>]/ActTool/BQ3 CPL: SD.SysToolSRun[<Channel>].ActTool.BQ3	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

PreTool.K1

PreTool.K1		PreTool.K1
Tool data element K1 of the prepared tool		
Path	/SysToolSRun[<Channel>]/PreTool/K1 CPL: SD.SysToolSRun[<Channel>].PreTool.K1	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read

	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	-	
Dependency	-	
Data type	xs:short (2 bytes)	

PreTool.K2

PreTool.K2		PreTool.K2
	Tool data element K2 of the prepared tool	
Path	/SysToolSRun[<Channel>]/PreTool/K2 CPL: SD.SysToolSRun[<Channel>].PreTool.K2	
Value range	-	
Pre-assignment	0	
Unit	-	
e variable values		
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	-	
Dependency	-	
Data type	xs:short (2 bytes)	

PreTool.SKQ

PreTool.SKQ		PreTool.SKQ
	Tool data element SKQ of the prepared tool	
Path	/SysToolSRun[<Channel>]/PreTool/SKQ CPL: SD.SysToolSRun[<Channel>].PreTool.SKQ	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
e variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	

System data

Update -
 Dependency -
 Data type isoStr31_t (31 bytes + 1 byte for end identification)

PreTool.IKQ1

PreTool.IKQ1	PreTool.IKQ1

Tool data element IKQ1 of the prepared tool

Path /SysToolSRun[<Channel>]/PreTool/IKQ1
 CPL: SD.SysToolSRun[<Channel>]/PreTool/IKQ1

Value range -

Pre-assignment 0

Unit -

Description of the variable values

Access rights MTB: Read and write
 Setter: Read and write
 User: Read and write
 NC prog.: Read and write
 PLC prog.: Read and write

Activation -

Update -

Dependency -

Data type xs:int (4 bytes)

PreTool.IKQ2

PreTool.IKQ2	PreTool.IKQ2

Tool data element IKQ2 of the prepared tool

Path /SysToolSRun[<Channel>]/PreTool/IKQ2
 CPL: SD.SysToolSRun[<Channel>].PreTool.IKQ2

Value range -

Pre-assignment 0

Unit -

Description of the variable values

Access rights MTB: Read and write
 Setter: Read and write
 User: Read and write
 NC prog.: Read and write
 PLC prog.: Read and write

Activation -

Update -

Dependency -

Data type xs:int (4 bytes)

PreTool.IKQ3

PreTool.IKQ3		PreTool.IKQ3
		Tool data element IKQ3 of the prepared tool
Path		/SysToolSRun[<Channel>]/PreTool/IKQ3 CPL: SD.SysToolSRun[<Channel>].PreTool.IKQ3
Value range		-
Pre-assignment		0
Unit		-
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		-
Update		-
Dependency		-
Data type		xs:int (4 bytes)

PreTool.IQ1

PreTool.IQ1		PreTool.IQ1
		Tool data element IQ1 of the prepared tool
Path		/SysToolSRun[<Channel>]/PreTool/IQ1 CPL: SD.SysToolSRun[<Channel>].PreTool.IQ1
Value range		-
Pre-assignment		0
Unit		-
Description of the variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation		-
Update		-
Dependency		-
Data type		xs:int (4 bytes)

PreTool.IQ2

PreTool.IQ2		PreTool.IQ2
		Tool data element IQ2 of the prepared tool
Path		/SysToolSRun[<Channel>]/PreTool/IQ2 CPL: SD.SysToolSRun[<Channel>].PreTool.IQ2

System data

Value range	-
Pre-assignment	0
Unit	-
Description of the variable values	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	-
Update	-
Dependency	-
Data type	xs:int (4 bytes)

PreTool.IQ3

PreTool.IQ3		PreTool.IQ3
Tool data element IQ3 of the prepared tool		
Path	/SysToolSRun[<Channel>]/PreTool/IQ3 CPL: SD.SysToolSRun[<Channel>].PreTool.IQ3	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read PLC prog.: Read	
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

PreTool.BQ1

PreTool.BQ1		PreTool.BQ1
Tool data element BQ1 of the prepared tool		
Path	/SysToolSRun[<Channel>]/PreTool/BQ1 CPL: SD.SysToolSRun[<Channel>].PreTool.BQ1	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values		
Access rights	MTB:	Read and write

	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

PreTool.BQ2

PreTool.BQ2		PreTool.BQ2
	Tool data element BQ2 of the prepared tool	
Path	/SysToolSRun[<Channel>]/PreTool/BQ2 CPL: SD.SysToolSRun[<Channel>].PreTool.BQ2	
Value range	-	
Pre-assignment	0	
Unit	-	
e variable values		
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	-	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

PreTool.BQ3

PreTool.BQ3		PreTool.BQ3
	Tool data element BQ3 of the prepared tool	
Path	/SysToolSRun[<Channel>]/PreTool/BQ3 CPL: SD.SysToolSRun[<Channel>].PreTool.BQ3	
Value range	-	
Pre-assignment	0	
Unit	-	
e variable values		
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	

System data

Update -
 Dependency -
 Data type xs:int (4 bytes)

Number of the active edge in the target block

Number of the active edge in the target block		TargEd
Number of the active edge in the target block.		
Path	/SysToolSRun[<Channel>]/TargEd CPL: SD.SysToolSRun[<Channel>].TargEd	
Value range	0..16	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No edge active
	1..16	Number of the active edge in the target block
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel when reaching the target block in the block pre-run	
Dependency	-	
Data type	Int_t (4 bytes)	

Number of the D-correction active in the target block

Number of the active D-correction in the target block		TargDNr
Number of the active D-correction in the target block.		
Path	/SysToolSRun[<Channel>]/TargDNr CPL: SD.SysToolSRun[<Channel>].TargDNr	
Value range	0..99	
Pre-assignment	0	
Unit	-	
Description of the variable values	0	No correction active
	1..99	Number of the active D correction in the target block
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By the NC kernel when reaching the target block in the block pre-run	
Dependency	-	
Data type	Int_t (4 bytes)	

Adjustment status of the tool corrections

Adjustment status of the tool correction		TCorrNotAdj
Adjustment status of the tool correction		
Path	/SysToolSRun[<Channel>]/TCorrNotAdj CPL: SD.SysToolSRun[<Channel>].TCorrNotAdj	
Value range	0..1	
Pre-assignment	1	
Unit	-	
Description of the variable values	0	Tool correction adjusted
	1	Tool correction not adjusted
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	-	
Update	By NC kernel when reaching the target block in the block pre-run and at active adjustment	
Dependency	-	
Data type	Bytes_t (1 byte)	

32.2.53 Tool type data of tools in the tool catalog "/SysToolType"

General information

The volatile system data consists of an array of 1...n and contains the contains the value pairs for IKQ2 and BQ3 for all tools in the tool catalog. Upon control startup, the array size "n" is initialized with the value specified in the "SDDefTICat.xml" file (preassignment: 67). The SD.SysToolType system data is initialized with the values in the "SDDatTICat.xml" file upon the start-up.

If the user creates user-defined tools in the tool catalog, the tools also have to be entered in the system data.

For more information, refer to the "IndraMotion MTX Functional Description" documentation.

The structural elements described are defined in the "sdtlcat.xsd" schema.

Structural element	Description
TypeNo	Type number (tool data element IKQ2)
Edge_variable	Variable number of edges
BQ3_Basic	Tool data element BQ3

Tab. 32-44: Overview on structural elements SysToolType

Type number in the tool catalog

TypeNo	TypeNo
Contains the type number (tool data element IKQ2) of the tool in the tool catalog.	

System data

Path	/SysToolType[1..n]/TypeNo CPL: SD.SysToolType[1..n].TypeNo
Value range	-
Pre-assignment	Different (see SDDatTICat.xml file)
Unit	-
Description of the variable values	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	-
Update	-
Dependency	-
Data type	Int_t (xs:int, 4 bytes)

Variable number of edges in the tool catalog

Variable number of edges		Edge_variable
Signifies if the number of edges of the tool in the tool catalog is fixed or can be edited.		
Path	/SysToolType[1..n]/Edge_variable CPL: SD.SysToolType[1..n].Edge_variable	
Value range	0 (false) 1 (true)	
Pre-assignment	Different (see SDDatTICat.xml file)	
Unit	-	
Description of the variable values	false: Number of edges is fixed. true: Edge number is variable.	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	-	
Update	-	
Dependency	-	
Data type	Boolean_t (xs:boolean, 1 Byte)	

Tool data element BQ3 in the tool catalog

BQ3_Basic		BQ3_Basic
Contains the tool data element BQ3 of the tool in the tool catalog.		
Path	/SysToolType[1..n]/BQ3_Basic CPL: SD.SysToolType[1..n].BQ3_Basic	
Value range	-	

Pre-assignment	Different (see SDDatTICat.xml file)
Unit	-
Description of the variable values	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	-
Update	-
Dependency	-
Data type	Int_t (xs:int, 4 bytes)

32.2.54 Variables for the tool change cycle TCH "/SysTch"

General information

This permanent and channel-specific system date is provided to set up the tool change cycle TCH and to design the program processes in the user programs TCHPICK and TCHPLACE.

For a detailed description of the mechanisms, refer to the manual "IndraMotion MTX Commissioning", chapter "Tool management with standard tool change".

The structural elements described are defined in the "sdtch.xsd" schema.

Structural element	Description
SETUP.MODE	Tool changer type
SETUP.SPINDLES	Number of spindle tools
SETUP.SPOFFSET	Position offset multiple-spindle
SETUP.SPISEC	Sector number of spindle tool
SETUP.SPIPLA	Position number of spindle tool
SETUP.MAGSEC	Sector number magazine
SETUP.MAGFIX	Hard disk-coded magazine
SETUP.SEARCH1	Search condition 1
SETUP.SEARCH2	Search condition 2
SETUP.SEARCH3	Search condition 3
SETUP.PREPRG	Program name of tool header (optional)
SETUP.POSTPRG	Program name of tool trailer (optional)
SETUP.ADJM6PRG	Program name "Block pre-run" (optional)
INTERNAL.*	Control-internally used data (not described in detail)
EXTERNAL.TOOL_FOL	Follow-on tool programmed
EXTERNAL.SP_EMPTY	Spindle empty

Tab. 32-45: Overview on structural elements "SysTch"

Tool changer type

Tool changer type		SETUP.MODE
	Specifies the type of tool changer.	
Path	/SysTch[<Channel>]/SETUP/MODE CPL: SD.SysTch[<Channel>].SETUP.MODE	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	1: Pickup automatic tool changer (milling) 2: Turret changer (turning)	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	The tool change cycle uses the currently set value.	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

Number of spindle tools

Number of spindle tools		SETUP.SPINDLES
	Specifies the number of spindle tools.	
Path	/SysTch[<Channel>]/SETUP/SPINDLES CPL: SD.SysTch[<Channel>].SETUP.SPINDLES	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Number of spindles	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	The tool change cycle uses the currently set value.	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

Position offset multiple-spindle	SETUP.SPOFFSET
Defines the position offset if more than one spindle tool is used.	

Path	/SysTch[<Channel>]/SETUP/SPOFFSET CPL: SD.SysTch[<Channel>].SETUP.SPOFFSET	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Position offset spindles	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	The tool change cycle uses the currently set value.	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

Sector number of spindle tool

Sector number of spindle tool		SETUP.SPISEC
Specifies the sector number of the spindle tool in the database.		
Path	/SysTch[<Channel>]/SETUP/SPISEC CPL: SD.SysTch[<Channel>].SETUP.SPISEC	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Sector number spindle	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	The tool change cycle uses the currently set value.	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

Position number of spindle tool

Position number of spindle tool		SETUP.SPIPLA
Specifies the position number of the spindle tool in the database.		
Path	/SysTch[<Channel>]/SETUP/SPIPLA CPL: SD.SysTch[<Channel>].SETUP.SPIPLA	
Value range	-	
Pre-assignment	0	
Unit	-	

System data

Description of the variable values	Place number of the spindle
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write
Activation	The tool change cycle uses the currently set value.
Update	-
Dependency	-
Data type	xs:int (4 bytes)

Sector number of magazine

Sector number magazine		SETUP.MAGSEC
	Specifies the sector number of the tool magazine in the database.	
Path	/SysTch[<Channel>]/SETUP/MAGSEC CPL: SD.SysTch[<Channel>].SETUP.MAGSEC	
Value range	-	
Pre-assignment	0	
Unit	-	
Description of the variable values	Sector number magazine	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	The tool change cycle uses the currently set value.	
Update	-	
Dependency	-	
Data type	xs:int (4 bytes)	

Hard disk-coded magazine

Hard disk-coded magazine		SETUP.MAGFIX
	Specifies, whether the tool magazine is a hard disk-coded magazine (FPC).	
Path	/SysTch[<Channel>]/SETUP/MAGFIX CPL: SD.SysTch[<Channel>].SETUP.MAGFIX	
Value range	0 (false) 1 (true)	
Pre-assignment	0 (false)	
Unit	-	
Description of the variable values	0 (false): No FPC magazine 1 (true): FPC magazine	
Access rights	MTB: Read and write Setter: Read and write	

User:	Read and write
NC prog.:	Read and write
PLC prog.:	Read and write
Activation	The tool change cycle uses the currently set value.
Update	-
Dependency	-
Data type	xs:boolean (1 byte)

Tool search condition 1

Tool search condition 1		SETUP.SEARCH1
	Specifies search condition 1 for the tool search always consisting of three search runs.	
Path	/SysTch[<Channel>]/SETUP/SEARCH1 CPL: SD.SysTch[<Channel>].SETUP.SEARCH1	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	Specified syntax: "<DecimalNumber 1>:<DecimalNumber 2>" The search condition is defined using the tool status bit BQ2. This is done using two decimal numbers separated by a colon (e.g. "16:208"). <Decimal number 2>: Total number of significances of all tool status bits to be considered in a search condition (e.g. tool worn 2^4 + tool inhibit 2^6 + tool broken 2^7 = 208) <Decimal number 1>: Mask for the requested states of the tool status bits (e.g. tool worn = TRUE, tool inhibit = FALSE, tool broken = FALSE selected in <DecimalNumber 2> $2^4 + 0 + 0 = 16$)	
Access rights	MTB: Read and write Setter: Read and write User: Read and write NC prog.: Read and write PLC prog.: Read and write	
Activation	The tool change cycle uses the currently set value.	
Update	-	
Dependency	All three search conditions SEARCH1, SEARCH2 und SEARCH3 have to be specified, i.e. no search condition may be empty.	
Data type	Up to 21 characters from IsoLatin1String	

Tool search condition 2

Tool search condition 2	SETUP.SEARCH2
Specifies search condition 2 for the tool search always consisting of three search runs (refer to the tool search condition 1).	

Tool search condition 3

Tool search condition 3	SETUP.SEARCH3

Specifies search condition 3 for the tool search always consisting of three search runs (refer to the tool search condition 1).

Program name of tool header (optional)

Program name of tool header (optional)	SETUP.PREPRG

Allows the user to integrate a subroutine as preload of the tool change (e.g. TCHPRE).

Path /SysTch[<Channel>]/SETUP/PREPRG

CPL: SD.SysTch[<Channel>].SETUP.PREPRG

Value range -

Pre-assignment <EmptyString>

Unit -

Description of the variable values Program name for tool header

Access rights **MTB:** Read and write

Setter: Read and write

User: Read and write

NC prog.: Read and write

PLC prog.: Read and write

Activation The tool change cycle uses the currently set value.

Update -

Dependency -

Data type Up to 16 characters from IsoLatin1String

Program name of tool trailer (optional)

Program name of tool trailer (optional)	SETUP.POSTPRG

Allows the user to integrate a subroutine to retighten the tool change (e.g. TCHPOST).

Path /SysTch[<Channel>]/SETUP/POSTPRG

CPL: SD.SysTch[<Channel>].SETUP.POSTPRG

Value range -

Pre-assignment <EmptyString>

Unit -

Description of the variable values Program name of tool trailer

Access rights **MTB:** Read and write

Setter: Read and write

User: Read and write

NC prog.: Read and write

PLC prog.: Read and write

Activation The tool change cycle uses the currently set value.

Update -

Dependency -

Data type Up to 16 characters from IsoLatin1String

Program name "Block pre-run" (optional)

Program name "Block pre-run" (optional)		SETUP. ADJM6PRG
	Allows the user to integrate a subroutine for the block pre-run before the tool is changed (e.g. TCHPRE).	
Path	/SysTch[<Channel>]/SETUP/ADJM6PRG CPL: SD.SysTch[<Channel>].SETUP.ADJM6PRG	
Value range	-	
Pre-assignment	<EmptyString>	
Unit	-	
Description of the variable values	Program name of tool trailer	
Access rights	MTB:	Read and write
	Setter:	Read and write
	User:	Read and write
	NC prog.:	Read and write
	PLC prog.:	Read and write
Activation	The tool change cycle uses the currently set value.	
Update	-	
Dependency	-	
Data type	Up to 16 characters from IsoLatin1String	

Follow-on tool programmed

Follow-on tool programmed		EXTERNAL. TOOL_FOL
	The tool changer program TCH enters into this variable whether a follow-on tool is empty.	
Path	/SysTch[<Channel>]/EXTERNAL/TOOL_FOL CPL: SD.SysTch[<Channel>].EXTERNAL.TOOL_FOL	
Value range	0 (false) 1 (true)	
Pre-assignment	0 (false)	
Unit	-	
Description of the variable values	0 (false):	Non follow-on tool programmed
	1 (true):	Follow-on tool is programmed
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read (and write by TCH)
	PLC prog.:	Read
Activation	-	
Update	By the tool change program TCH	
Dependency	-	
Data type	xs:boolean (1 byte)	

Spindle empty

Spindle empty		EXTERNAL. SP_EMPTY
The tool changer program TCH enters into this variable whether the spindle is empty.		
Path	/SysTch[<Channel>]/EXTERNAL/SP_EMPTY CPL: SD.SysTch[<Channel>].EXTERNAL.SP_EMPTY	
Value range	0 (false) 1 (true)	
Pre-assignment	0 (false)	
Unit	-	
Description of the variable values	0 (false): Spindle contains tool 1 (true): Spindle is empty	
Access rights	MTB: Read Setter: Read User: Read NC prog.: Read (and write by TCH) PLC prog.: Read	
Activation	-	
Update	By the tool change program TCH	
Dependency	-	
Data type	xs:boolean (1 byte)	

32.2.55 Displaying the SRAM assignment "/SysSRamMemInfo"

General information

Information on the SRAM assignment is stored in the volatile system data structure "SysSRamMemInfo".

The following structural elements are defined in the schema "sd_opdta.xsd".

All structural elements are of the data type UnsignedInt_t(4Bytes).

Structural element	Description
CplVarMax	Maximum available memory for permanent PLC variables in the system..
CplVarUsed	Currently used system memory by permanent CPL variables.
SpsVarMax	Maximum available memory for PLC retaining data in the system.
SpsVarUsed	System memory currently required for the PLC retain data.
SdDataMax	Maximum available system memory for permanent system data.
SdDataUsed	System memory currently required for permanent system data.

Tab. 32-46: Overview on the structural elements "SysSRamMemInfo"

32.2.56 Checksum for Sercos configuration"/SysSercCfgInfo"

General information

In the permanent system data structure SysSercCfgInfo, the checksum for the Sercos configuration and a flag for the Sercos configuration change status is stored.

The following structural elements are defined in the schema "sdcfginfo.xsd".

Structural element	Description
SercConfig-Checksum	Data type: Double_t Sercos configuration file (/cf_user/Sercos_config.scs) checksum. Information about the Sercos configuration of the control is entered in this file upon control startup: Checksum for the control configuration data (MACODA). Checksum for the SCSP configuration parameters. Checksum for all configured drives (S-0-0531). Name of all used SCS files and the last time they have been accessed. Upon the control startup, the new checksum is calculated and stored in this variable.
SercConf-Changed	Data type: Boolean_t Upon each control startup, the old checksum is compared to the new checksum. If the checksums differ, the flag is set to True, otherwise, it is set to False.

Tab. 32-47: Overview of structural elements SysSercCfgInfo

32.2.57 Data of the zero point offsets (global) "/SysZOData"

General information

The system data SysZOData (in connection with SysZODataC and SysZODataA) is used to configure the basic zero point offset as well as for switching between the zero point offset banks and the cycle G500.

The global information is stored in this structure.

The structural elements described here are stored in the "sdzo.xsd" schema.

Structural element	Description
Enabled	Is the functionality enabled (0 or 1)? A user interface can check if zero point offsets G54.1-G59.5 or G500 (1)-G500 (100), including the basic zero point offsets are to be displayed.
Unit	In which unit are the offset values specified? 0=mm; 1=inch

Tab. 32-48: Overview of structural elements SysZOData

State of the ZO functionality

State of the ZO functionality	Enabled

System data

Configures if zero point offsets are to be displayed as banks (G54.1-G59.5) or as cycle (G500(1)-G500(100)), including the basic zero point offsets) are to be displayed. The value zero stands for "banks" and the value one stands for "Cycle G500".

This system data is not available by default and has to be permanently created by the commissioning engineer. Use the "/feprom/SDDefZO.xml_" file as template.

Path	/SysZOData/Enabled CPL: SD.SysZOData.Enabled
Value range	"0" or "1" as integer number (Int)
Pre-assignment	0
Unit	-
Description of the variable values	0 - Show zero point offsets as banks (G54.1-G59.5), no basic ZO 1 - Show zero point offsets as cycle (G500(1)-G500(100)), including ZO
Access rights	MTB: Read, write Setter: Read User: Read NC prog.: Read PLC prog.: Read
Activation	Copying the file "/feprom/schemas/sdzo.xsd_" to "/usrfep/schemas/sdzo.xsd", copying the file "/feprom/SDDefZO.xml_" to "/usrfep/SDDefZO.xml" and re-start the NC.
Update	-
Dependency	-
Data type	Positive integer number (UnsignedByte)

Units of the basic ZO values

Units of the basic ZO values		Unit
Configures if values of the basic zero point offset are to be interpreted in millimeters (0) or inch (1).		
This system data is not available by default and has to be permanently created by the commissioning engineer. Use the "/feprom/SDDefZO.xml_" file as template.		
Path	/SysZOData/Unit CPL: SD.SysZOData.Unit	
Value range	"0" or "1"	
Pre-assignment	0	
Unit	-	
Description of the variable values	"0" for millimeters "1" for inch	
Access rights	MTB: Read, write Setter: Read, write User: Read NC prog.: Read, write PLC prog.: Read, write	

Activation	Copying the file "/feprom/schemas/sdzo.xsd_" to "/usrfep/schemas/sdzo.xsd", copying the file "/feprom/SDDefZO.xml_" to "/usrfep/SDDefZO.xml" and re-start the NC.
Update	-
Dependency	-
Data type	Positive integer number (UnsignedByte)

32.2.58 Data of the zero point offsets (axis-related) "/SysZODataA"

General information

The system date SysZODataA (in connection with SysZOData and SysZODataC) is used to configure the basic zero point offset as well as for switching between the zero point offset banks and the cycle G500.

The axis-related information is stored in this structure.

The structural elements described here are stored in the "sdzo.xsd" schema.

Structural element	Description
AxName	System axis name The axis name for the basic zero point offset is configured.
Offset	Offset The zero point offset of this axis is configured for the basic zero point offset..

Tab. 32-49: Overview of structural elements SysZODataA

System axis name for the basic zero point offset

System axis name	AxName

In this system name, an axis for the basic zero point offset is configured. Enter the system axis name. If the entry is not required, leave the system date blank.

The functionality basic zero point offset is only active if the macro for G53 or the cycle STP_BaseZO including the matching init programs is active. The values are applied (if the basic zero point offset has been changed) upon the next call of G53 (time of preparation).

This system date is not available by default and has to be permanently created by the commissioning engineer. Use the "/feprom/SDDefZO.xml_" file as template.

Path	/SysZODataA[1]/AxName CPL: SD.SysZODataA[1].AxName
Value range	System axis names as IsoLatin1String or empty string
Pre-assignment	"" (empty string)
Unit	-
Description of the variable values	System axis name for the basic zero point offset
Access rights	MTB: Read, write Setter: Read, write User: Read NC prog.: Read, write PLC prog.: Read, write

System data

Activation	Copying the file "/feprom/schemas/sdzo.xsd_" to "/usrfep/schemas/sdzo.xsd", copying the file "/feprom/SDDefZO.xml_" to "/usrfep/SDDefZO.xml" and re-start the NC.
Update	-
Dependency	-
Data type	isoLatin1String of 8 characters

System axis offset for the basic zero point offset

System axis offset for the basic zero point offset	Offset

In this system date, the axis offset for the basic zero point offset is configured. The value is only taken into consideration if a valid system axis name is entered in the substructure `AxName` ([chapter "System axis name for the basic zero point offset" on page 643](#)).

The functionality basic zero point offset is only active if the macro for G53 or the cycle `STP_BaseZO` including the matching init programs is active. The values are applied (if the basic zero point offset has been changed) upon the next call of G53 (time of preparation).

This system date is not available by default and has to be permanently created by the commissioning engineer. Use the "/feprom/SDDefZO.xml_" file as template.

Path	/SysZODataA[1..x]/Offset CPL: SD.SysZODataA[1..x].Offset
Value range	-
Pre-assignment	0
Unit	mm/inch depending on the configuration (see chapter "Units of the basic ZO values" on page 642)
Description of the variable values	Offset for the basic zero point offset
Access rights	MTB: Read, write Setter: Read, write User: Read NC prog.: Read, write PLC prog.: Read, write
Activation	Copying the file "/feprom/schemas/sdzo.xsd_" to "/usrfep/schemas/sdzo.xsd", copying the file "/feprom/SDDefZO.xml_" to "/usrfep/SDDefZO.xml" and re-start the NC.
Update	-
Dependency	-
Data type	Double

32.2.59 Data of the zero point offsets (channel-related) "/SysZODataC"

General information

The system date SysZODataC (in connection with SysZOData and SysZODataA) is used to configure the basic zero point offset as well as for switching between the zero point offset banks and the cycle G500.

The channel-related information is stored in this structure.

The structural elements described here are stored in the "sdzo.xsd" schema.

Structural element	Description
Current	Number of the currently active zero point offset table in the channel This date is only refreshed (and is only then valid) if the zero point offsets are switched by the cycle G500.

Tab. 32-50: Overview of structural elements SysZODataC

Number of the currently active zero point offset table in the channel

Number of the currently active zero point offset table in the channel	Current

In this system data, the control stores information about which ZO table is currently active when switching by means of the G500 cycle. Valid values: 0 to 100. The value 0 indicates that the ZO has been disabled.

This system data is not available by default and has to be permanently created by the commissioning engineer. Use the "/feprom/SDDefZO.xml_" file as template.

Path	/SysZODataC/Current CPL: SD.SysZODataC.Current	
Value range	0-100 as integer	
Pre-assignment	0	
Unit	-	
Description of the variable values	Number of the currently active zero point offset table in the channel	
Access rights	MTB:	Read
	Setter:	Read
	User:	Read
	NC prog.:	Read
	PLC prog.:	Read
Activation	Copying the file "/feprom/schemas/sdzo.xsd_" to "/usrfep/schemas/sdzo.xsd", copying the file "/feprom/SDDefZO.xml_" to "/usrfep/SDDefZO.xml" and re-start the NC.	
Update	By the control if the ZO is switched by means of cycle G500.	
Dependency	-	
Data type	Positive integer number (UnsignedByte)	

32.2.60 Interaction between simulation and NC program

The system data structure SysSimProg allows to define and manage graphics/geometries for the simulation from the NC program or the manual data input (MDI). The structure created in the permanent memory is channel-related. It is used/described by the following instructions for the simulation referring to the workpieces, clamping devices and tools:

- STOCK
- STOCKIMP
- STOCKEXP
- STOCKSEL
- STOCKADD
- TOOL

The following structural elements are defined in the "sdadvdistsig.xsd_" schema:

Structural element		Description
DwellTime		Response time of the simulation in ms (0..1000)
ExchgInfo		Data exchange between the NC program and the simulation: Bit 1 - Debug mode active Bit 2 - Acknowledgement bit (synchronization) Bits 3 and 4: Touch probe number (1 or 2)
ProcessParam		Data from the NC program for the simulation: Bits 1 and 2: Axis functionality of the measuring axis Bit 3 and higher: System axis index of the measurement axis
Stock	/...	Data for workpiece geometries
	Counter	Counter for workpiece geometry instructions STOCK. Incrementing generates the change event for the simulation.
	NodeName	Name of the workpiece node (32 characters)
	Add	0 - Replace 1 - Add
Tool	/...	Data for tool descriptions
	Counter	Counter for tool descriptions TOOL. Incrementing generates the change event for the simulation.

Tab. 32-51: Overview on the structural elements "SysSimProg"

Data of the SysSimProg structure is reserved for internal use in the instructions for the simulation and the basic measuring cycle STP_PrBase. Direct user access is not provided. The DwellTime element is an exception.



The instructions for the simulation are not reported block-synchronously to the simulation, thus possibly causing problems in case of multiple and subsequent instructions following one another quickly. To solve them, increase the DwellTime or use dwell times or a program stop.

33 Sercos initialization

33.1 General information

Apart from the parameters in the data group "Sercos Communication (SCS)" of the NC Configurator, other parameters are defined under "Sercos Files" in MTX. With these files, you can parameterize all connected drives.

Sercos files can be identified by the file extension ".scs". They can be edited with the NC editor, since they are pure ASCII files.

In the FEPROM file system of MTX is an example set of Sercos files for IndraDrive drives under the path "root/feeprom/scsindra".

Copy these files into the root directory of the file system, adjust them according to your requirements and then move them into the User-FEPROM file system ("root/usrfep").

⚠ CAUTION

Unauthorized changes to parameters or files may lead to personal injury or damage to equipment!

Parameters or files may therefore only be changed by properly trained expert personnel! This personnel must be familiar with the user interface and the directory structure of MTX and must have thorough knowledge of the Sercos interface. This personnel has to be able to detect the risks involved in changing parameters and the inherent general risks of mechanical, electrical or electronic equipment.

Rexroth is not liable for consequential damages resulting from incorrectly programmed, calculated or optimized parameters or limit data that has not been complied with!

33.2 Sercos initialization sequence

The sequence for establishing communication between the MTX and the connected drives cannot be changed.

In general, a distinction is made between 5 different states (**Sercos phases**) that run sequentially at startup (see figure below):

Sercos initialization

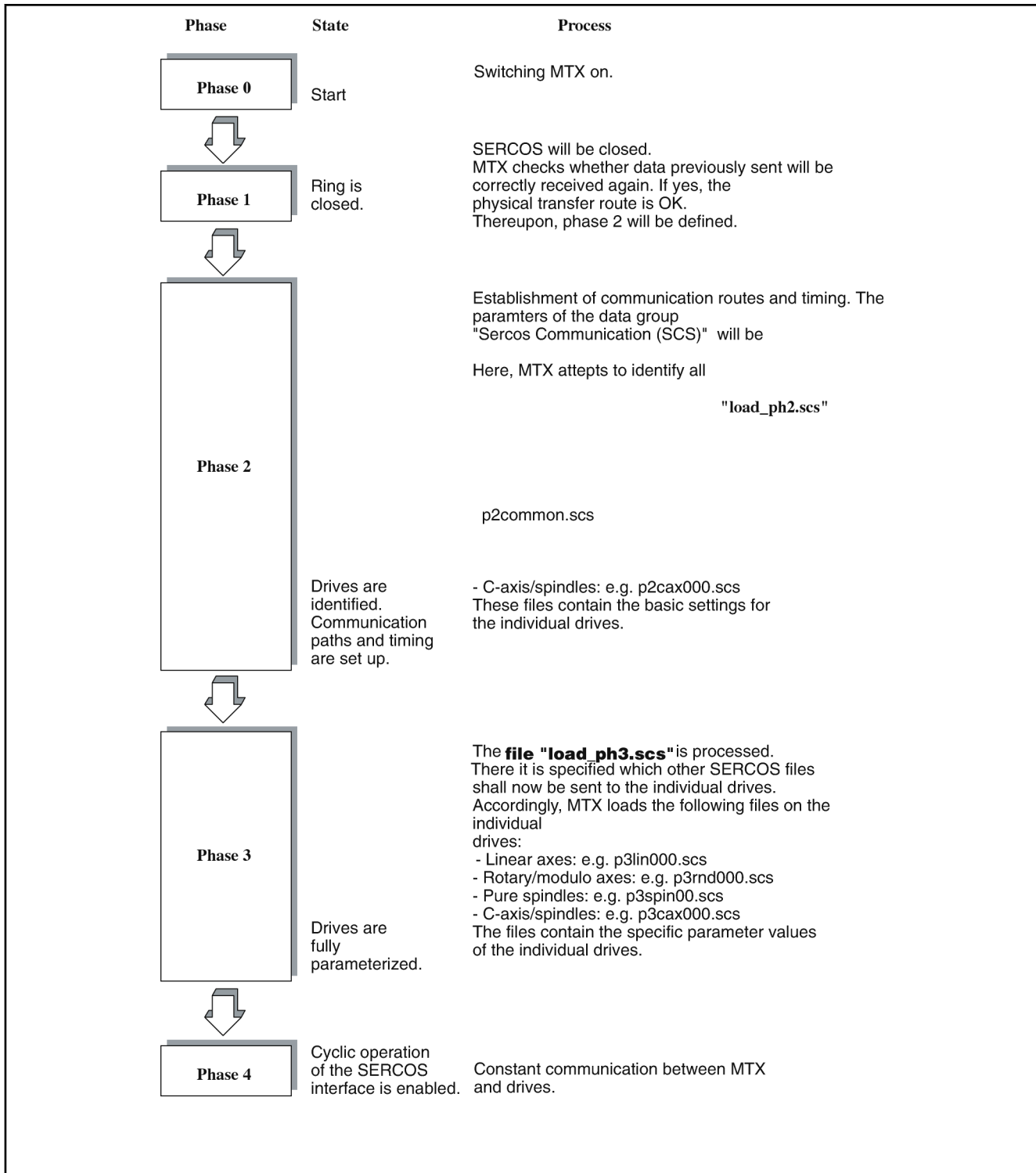


Fig. 33-1: Sercos initialization (phases)

Logging Sercos parameters transmitted at startup

The Sercos parameters transmitted by the controller at startup can be logged for diagnostics purposes.

To achieve this, delete the ";" character at the beginning of the line in the following Sercos files.

- Start logging for Phase 2:

In the file "load_ph2.scs", delete the character ";" at the beginning of the line

```
"dbg p2dbg.scs".
```

- **Start logging for Phase 3:**

In the file "load_ph3.scs", delete the character ";" at the beginning of the line

```
"dbg p3dbg.scs".
```

The log results are then found in the files "p2dbg.scs" and/or " p3dbg.scs" after startup.

To disable logging, simply reinsert the ";" character into the original locations.

Check Sercos timing at startup



Use this function for commissioning only!

Sercos timing, as defined in the data group "Sercos Communication (SCS)" of NC configurator and in the *.scs files of phase 2, can be checked during Phase 3 of Sercos startup:

To achieve this, delete the ";" character at the beginning of the line reading "opt -m time.scs" in the Sercos file "load_ph3.scs".

Upon completion of the startup, Sercos timing is documented in the file "time.scs".

To disable logging, simply reinsert the ";" character into the original location.

34 Appendix

List of all parameters

Path	Parameter ID	Macoda No.	Description
/MAIN/	NofCh	9040 00001	Number of channels
/MAIN/	NofDr	1001 00001	Drive function type
/MAIN/Dr[i]/	EnablDr		
/MAIN/Dr[i]/	AxFunc		
/MAIN/Dr[i]/	SpFunc		
/MAIN/Dr[i]/SpFunc/	SpInd		
/MAIN/	MinNofDrNodes	6001 00001	Minimum number of configured drives
/MAIN/	NofDrNodes	-	Number of configured drives
/MAIN/	MaxNofCh	-	Maximum number of channels (startup.xml)
/MAIN/	MaxNofDr	-	Maximum number of NC-contr. drives (start-up.xml)
/MAIN/	MaxNofSp	-	Maximum number of NC-contr. spindles (startup.xml)
/MAIN/Dr[i]/	EnablVirtMode	1001 00010	Virtual drive
/MAIN/Dr[i]/	SupprCmdValOut	1001 00011	Drive without command value input
/MAIN/Dr[i]/	DelayedErrorReaction	1001 00012	Delayed reaction in case of non-fatal drive errors
/MAIN/Dr[i]/	SysDrName	1003 00001	System drive names
/MAIN/Dr[i]/AxFunc/	DefaultCh	1003 00002	Channel assignment
/MAIN/Dr[i]/AxFunc/	SysAxType	1003 00004	Axis movement type
/MAIN/Dr[i]/Scs/	DrType	1050 00001	Drive type
/MAIN/Dr[i]/	EnablScsCom	1050 00002	Drive exists in the ring
/MAIN/Dr[i]/Scs/	ScsAddr	1050 00004	Assignment of drives to Sercos addresses
/CHAN/Ch[i]/Coord/Acs/ChAx[i]/	ChAxName	7010 00010	Names of channel axes
/CHAN/Ch[i]/Coord/Acs/OptAx-Name/	OptChAxName	7010 00020	Optional axis designation
/CHAN/Ch[i]/Coord/Acs/ChAx[i]/	ChAxClass	7010 00030	Coordinate meaning
/CHAN/Ch[i]/Couple/	AxG96	7010 00110	Channel-axis index of the reference axis for constant cutting velocity (G96)
/CHAN/Ch[i]/Couple/	CoordG96	7010 00120	Reference coordinate system for G96 reference axis
/CHAN/Ch[i]Couple/	MainSp	7020 00010	Main spindle
/CHAN/Ch[i]/Path/Ovrd/ChOvrd-Step[i]	AxOvrdStep	7030 00010	Feed override for channel
/AX/AsynchrAxOvrdIf/			

Appendix

Path	Parameter ID	Macoda No.	Description
/CHAN/Ch[i]/Path/Ovrd/ /AX/AsynchrAxOvrdIf/	MaxChOvrd MaxAxOvrd	7030 00012	Maximum feed override
/CHAN/Ch[i]/Path/RapTravOvrd/	EnablRapTravOvrd	7030 00014	Enable rapid traverse override
/CHAN/Ch[i]/Path/Vel/	RedChVel	7030 00110	Value of limited rapid traverse velocity
/CHAN/Ch[i]/Path/Vel/	MaxChVel	7030 00120	Maximum path velocity
/CHAN/Ch[i]/Path/Acc/	MaxChAcc	7030 00210	Path acceleration
/CHAN/Ch[i]/Path/Acc/	MaxChDec	7030 00220	Path deceleration
/CHAN/Ch[i]/Path/Jump/	MinAngVelJump	7030 00310	Minimum angle to consider the axis jump capability
/CHAN/Ch[i]/Path/KinLim/	KinLimLevel	7030 00410	Kinematic limitation level
/CHAN/Ch[i]/Path/KinLim/	KinLimWeight	7030 00420	Kinematic limitation scaling
/CHAN/Ch[i]/Path/KinLim/	KinLimFilter	7030 00430	Kinematic limitation filter
/CHAN/Ch[i]/Path/Shape/	ChShapeG8	7050 00320	SHAPE order for jerk-limited path operation (G8)
/CHAN/Ch[i]/Path/Shape/	MinChShape	7050 00330	Minimum SHAPE order for path jerk limitation (G8, G9)
/CHAN/Ch[i]/Path/Shape/	ChShapeG9	7050 00340	SHAPE order for jerk-limited positioning mode (G9)
/CHAN/Ch[i]/Ini/	DelZpo	7050 00510	SetPos Delete offset with basic setting
/CHAN/Ch[i]/Ini/	NCRresetState	7060 00010	State after startup (switched-on state)
/CHAN/Ch[i]/Ini/	ChResetState	7060 00020	State after channel reset
/CHAN/Ch[i]/Ini/	ChIniProgFile	7060 00030	File name of the channel initialization program
/CHAN/Ch[i]/Ini/	AutoProgSel	7060 00410	Automatic program reselection
/CHAN/Ch[i]/Coord/Wcs/	LinCoord[1..3]	7080 00010 [i] [1..3]	Designation of the linear coordinate (tool coordinates)
/CHAN/Ch[i]/Coord/Wcs/	OriCoord[1..3]	7080 00010 [i] [4..6]	Designation of the rotary coordinate (tool coordinates)
/CHAN/Ch[i]/Coord/Mcs/	CartCoord[1..6]	7080 00020	Coordinate designation (machine coordinates)
/AX/Dr[i]/PosLog/	EnablModCalc	1003 00004	Axis movement type
/AX/Dr[i]/PosLog/	ModePosLog	1003 00005	Positioning logic
/AX/Dr[i]/Shape/	MinAxShape	1003 00007	Minimum SHAPE order for axis
/AX/Dr[i]/Shape/	AxShape	1003 00008	SHAPE order for axis
/AX/Dr[i]/FeedFwd/	AxFeedFwd	1003 00009	Function "Zero following error movement" (feed forward) can be enabled
/AX/Dr[i]/TorqRed/	RedMaxTorq	1003 00010	Presetting for reduced maximum axis torque
/AX/Dr[i]/Vel/	FeedCalc	1003 00020	Axis not taken into account for feed calculation with "FeedAd" (FAD)

Path	Parameter ID	Macoda No.	Description
/AX/Dr[i]/PosLog/	EnablChgPosLog	1003 00050	Permit switching the positioning logic of end-less axes using "PosMode" (PMD)
/AX/Dr[i]/LocProg/Part/	NofLoc	1003 00055	Places per circle for Hirth axis
/AX/Dr[i]/LocProg/	ModeLocProg	1003 00056	Place programming for Hirth axis
/AX/Dr[i]/LocProg/AutoCorr/	EnablAutoCorr	1003 00057	End point correction for Hirth axes
/AX/Dr[i]/TorqRed/	MaxStandstTorq	1003 00070	Max. static torque of a single axis in percent
/AX/Dr[i]/Ovrd/	EnablAxAsynchrOvrd	1003 00100	Override for auxiliary axis
/AX/Dr[i]/Vel/	MaxVel	1005 00002	Maximum axis velocity and maximum rapid traverse velocity
/AX/Dr[i]/Jog/Vel/	SlowAxVel	1005 00003	Manual feed, slow
/AX/Dr[i]/Jog/Vel/	MediumAxVel	1005 00004	Manual feed, medium
/AX/Dr[i]/Jog/Vel/	FastAxVel	1005 00005	Manual feed, fast
/AX/Dr[i]/Jog/Vel/	RapidAxVel	1005 00006	Manual feed, rapid traverse
/AX/Dr[i]/Jog/Vel/	DefStepAxVel	1005 00007	Manual feed for defined steps
/AX/Dr[i]/Jog/Vel/	VarDistAxVel	1005 00008	Manual feed for variable steps
/AX/Dr[i]/Acc/	FeedAxAcc	1010 00001	Max. axis acceleration in feed mode
/AX/Dr[i]/Acc/	RapidAxAcc	1010 00003	Max. axis acceleration in rapid traverse
/AX/Dr[i]/Acc/	AxAccReserve	1010 00012	Axis acceleration reserve
/AX/Dr[i]/Jerk/	FeedAxJerk	1010 00020	Max. axis jerk in feed mode
/AX/Dr[i]/Jerk/	RapidAxJerk	1010 00022	Max. axis jerk in rapid traverse
/AX/Dr[i]/Jerk/	AxJerkReserve	1010 00013	Axis jerk reserve
/AX/Dr[i]/Jog/	AccAxJog	1010 00002	Jog acceleration
/AX/Dr[i]/Jog/	JerkAxJog	1010 00021	Jog jerk
/AX/Dr[i]/Jog/	IncrAxJog	1015 00001	Axis resolution
/AX/Dr[i]/Jog/	VarDistAxJog	1015 00002	Variable increment step
/AX/Dr[i]/Pos/TrvLim[1]/ /AX/Dr[i]/Pos/TrvLim[2]/ /AX/Dr[i]/Pos/TrvLim[1]/	EnablTrvLim EnablTrvLim PoTrvLim	1020 00001	Software limit switch, positive
/AX/Dr[i]/Pos/TrvLim[1]/	NeTrvLim	1020 00002	Software limit switch, negative
/AX/Dr[i]/Pos/TrvLim[2]/	PoTrvLim	1020 00003	Second software limit switch, positive
/AX/Dr[i]/Pos/TrvLim[2]/	NeTrvLim	1020 00004	Second software limit switch, negative
/CHAN/Ch[i]/Path/Ovrd/ChOvrd- Step[1] /AX/AsynchrAxOvrdIf/	AxOvrdStep[1]	7030 00010	Feed override for channel
/AX/AsynchrAxOvrdIf/	MaxAxOvrd	7030 00012	Maximum feed override
/SP/SpFunc/Sp[i]/Base/	SpType	1040 00001	Selection of spindle type

Appendix

Path	Parameter ID	Macoda No.	Description
/SP/SpAdmin/Sp/SpGrAlloc/Sp[i]/	SpGrInd	1040 00002	Assignment of a physical spindle to a spindle group
/SP/SpFunc/Sp[i]/Gear/	GearChgMethod	1040 00008	Gear change method
/SP/SpFunc/Sp[i]/Gear/ScsParSet/ /SP/SpFunc/Sp[i]/Gear/ScsParSet/ /SP/SpFunc/Sp[i]/Gear/ScsParSet/	NofScsParSets PosCtrlSpSet PosCtrlCAxSet	1040 00009	Drive parameter blocks per G-level (only for IndraDrive)
/SP/SpFunc/Sp[i]/Gear/	NofGears	1040 00010	Number of gear ranges
/SP/SpFunc/Sp[i]/Gear/Gear-Step[1]/	MinSpSpeed	1040 00011	Minimum spindle speed in each gear stage
/SP/SpFunc/Sp[i]/Gear/Gear-Step[1]/	MaxSpSpeed	1040 00012	Max. spindle speed of the gear stage in rpm
/SP/SpFunc/Sp[i]/Jog/Speed/	JogSpeed[i]	1040 00015	Jog spindle revolution speed
/SP/SpFunc/Sp[i]/SpeedWin/	AbsWin	1040 00020	Speed reached window
/SP/SpFunc/Sp[i]/SpeedWin/	PercWin	1040 00021	Speed reached window
/SP/SpFunc/Sp[i]/SpeedWin/	StandstillWin	1040 00022	Standstill window
/SP/SpFunc/Sp[i]/Gear/Gear-Step[1]/	MaxSpAcc	1040 00031	Maximum acceleration rate
/SP/SpFunc/Sp[i]/Ovrd/	SpOvrdStep[i]	1040 00041	Spindle override in 32 steps
/SP/SpFunc/Sp[i]/Ovrd/	MaxSpOvrd	1040 00042	Maximum spindle override
/SP/SpFunc/Sp[i]/Gear/Gear-Step[1]/	MaxSpAccPosCtrl[1]	1040 00051	Position interface: First acceleration in rad/s^2
/SP/SpFunc/Sp[i]/Gear/Gear-Step[1]/	MaxSpAccPosCtrl[2]	1040 00052	Position interface: Second acceleration in rad/s^2
/SP/SpFunc/Sp[i]/Gear/Gear-Step[1]/	SwiSpeed	1040 00053	Position interface: Switching speed for second acceleration
/SP/SpFunc/Sp[i]/Gear/Gear-Step[1]/	MaxSpJerkPosCtrl	1040 00055	Max. spindle jerk in position control mode
/SP/SpFunc/Sp[i]/SpTurrAx/	MeasSysSpTurrAx	1040 00060	C-axis: tool turret
/SP/SpFunc/Sp[i]/SpCtrl/Ch[1]/	EnablChCtrl	1040 00070	Spindle management
/SP/SpFunc/Sp[i]/Encoder/	DrIndSpEnco	1040 00081	Drive number of the external spindle encoder
/SP/SpFunc/Sp[i]/Encoder/	ScsParSpEnco	1040 00082	Sercos parameter of the external spindle encoder
/SP/SpFunc/SpSyn/Syn[1]/	SpCW	1040 00101	Spindle syntax: Clockwise rotation
/SP/SpFunc/SpSyn/Syn[1]/	SpCWCool	1040 00102	Spindle syntax: clockwise rotation - coolant ON
/SP/SpFunc/SpSyn/Syn[1]/	SpCCW	1040 00103	Spindle syntax: Counterclockwise rotation
SP/SpFunc/SpSyn/Syn[1]/	SpCCWCool	1040 00104	Spindle syntax: counterclockwise rotation - coolant ON
/SP/SpFunc/SpSyn/Syn[1]/	SpStop	1040 00105	Spindle syntax: Stop

Path	Parameter ID	Macoda No.	Description
/SP/SpFunc/SpSyn/Syn[1]/	SpPos	1040 00106	Spindle syntax: orient
/SP/SpFunc/SpSyn/Syn[1]/	SpGearAutoSel	1040 00107	Spindle syntax: automatic gear stage selection
/SP/SpFunc/SpSyn/Syn[1]/	SpGear1Sel	1040 00108	Spindle syntax: Manual gear stage selection, GTS 1
/SP/SpFunc/SpSyn/Syn[1]/	SpGear2Sel	1040 00109	Spindle syntax: Manual gear stage selection, GTS 2
/SP/SpFunc/SpSyn/Syn[1]/	SpGear3Sel	1040 00110	Spindle syntax: Manual gear stage selection, GTS 3
/SP/SpFunc/SpSyn/Syn[1]/	SpGear4Sel	1040 00111	Spindle syntax: Manual gear stage selection, GTS 4
/SP/SpFunc/SpSyn/Syn[1]/	SpGearIdleSel	1040 00112	Spindle syntax: Idle state - Disengage gear stage
/SP/SpGrFunc/SpGrSyn[1]/	SpGrCW	1040 00201	Spindle group syntax: Clockwise rotation
/SP/SpGrFunc/SpGrSyn[1]/	SpGrCWCool	1040 00202	Spindle group syntax: clockwise rotation - coolant ON
/SP/SpGrFunc/SpGrSyn[1]/	SpGrCCW	1040 00203	Spindle group syntax: Counterclockwise rotation
/SP/SpGrFunc/SpGrSyn[1]/	SpGrCCWCool	1040 00204	Spindle group syntax: counterclockwise rotation - coolant ON
/SP/SpGrFunc/SpGrSyn[1]/	SpGrStop	1040 00205	Spindle group syntax: Stop
/SP/SpGrFunc/SpGrSyn[1]/	SpGrPos	1040 00206	Spindle group syntax: orient
/SP/SpGrFunc/SpGrSyn[1]/	SpGrGearAutoSel	1040 00207	Spindle group syntax: automatic gear stage selection
/SP/SpGrFunc/SpGrSyn[1]/	SpGrGear1Sel	1040 00208	Spindle group syntax: Manual gear stage selection, GTS 1
/SP/SpGrFunc/SpGrSyn[1]/	SpGrGear2Sel	1040 00209	Spindle group syntax: Manual gear stage selection, GTS 2
/SP/SpGrFunc/SpGrSyn[1]/	SpGrGear3Sel	1040 00210	Spindle group syntax: Manual gear stage selection, GTS 3
/SP/SpGrFunc/SpGrSyn[1]/	SpGrGear4Sel	1040 00211	Spindle group syntax: Manual gear stage selection, GTS 4
/SP/SpGrFunc/SpGrSyn[1]/	SpGrGearIdleSel	1040 00212	Spindle group syntax: Idle state - Disengage gear stage
/SP/SpAdmin/BlkSwiCtrl/Waiting-Points/VarStrategy2/	Strat2WaitSpSpeed Strat2WaitSpStop Strat2WaitSpOri Strat2WaitSpGear	1040 00300	Waiting points in time - Strategy 2: Speed reached - Strategy 2: Spindle standstill - Strategy 2: Spindle oriented - Strategy 2: Gear range changed

Appendix

Path	Parameter ID	Macoda No.	Description
/SP/SpAdmin/BlkSwiCtrl/Waiting-Points/VarStrategy3/	Strat3WaitSpSpeed Strat3WaitSpStop Strat3WaitSpOri Strat3WaitSpGear	1040 00301	Waiting points in time - Strategy 3: Speed reached - Strategy 3: Spindle standstill - Strategy 3: Spindle oriented - Strategy 3: Gear range changed
/SP/SpAdmin/	ChSp	1040 00999	Enable channel spindles
/SP/SpAdmin/	Sp	1040 00999	Switch-on system spindles
/SP/SpAdmin/ChSp/ChSpGrAlloc/Ch[1]/ChSp[1]/	SpIndChSp	7020 00020	Channel spindles: Assignment from Channel Spindles to System Spindles
/SP/SpAdmin/BlkSwiCtrl/Ch[1]/Sp_ChSp[1]/	SelBlkSwiStrategy	7020 00030	Selection of block switching strategy
/SP/SpAdmin/BlkSwiCtrl/Ch[1]/Sp_ChSp[1]/	EnablBlkSwiStrategy	7020 00040	Enable block switching strategy
/SCS/VirtMode/Dr[i]/	RefPos	1015 00110	Reference point position of virtual axes
/SCS/TestMode/	EnablTestMode	1050 00020	Controller startup in Test mode
/SCS/ScsIf[1]/	ScsTimeOut	1050 00021	Time monitoring of Sercos startup
/SCS/DataHandl/	ModeScsFiles	1050 00022	SCS files - allow suppressing of phase 3 download
/SCS/DataHandl/Dr[i]/	ScsIdCmdVal	1050 00050	Sercos ID for NC command Values
/SCS/ScsIf[1]/	ScsCycTime	9030 00001	NC cycle time
/SCSP/			Data group SCSP
SCSP/Dr[i]/Gen/	StorMode	omitted, Sercos Id: S-0-0269	Storage mode
SCSP/Dr[i]/Gen/	Enco	Not applicable	Encoder selection
SCSP/Dr[i]/Gen/	MpxCh	Not applicable	Multiplex channel
SCSP/Dr[i]/Lim/	PosLoopKv	omitted, Sercos Id: S-0-0104	Servo gain factor (Kv) of position controller
SCSP/Dr[i]/Lim/	TrqLim	omitted, Sercos Id: S-0-0092	Torque limit value
SCSP/Dr[i]/Teleg/	DispUsrPara	Not applicable	User parameter display

Path	Parameter ID	Macoda No.	Description
SCSP/Dr[i]/Teleg/MDT	MDTConf[1..16]	omitted, Sercos Id: S-0-1050.0.6	Master data telegram entry
SCSP/Dr[i]/Teleg/AT	ATConf[1..16]	omitted, Sercos Id: S-0-1050.1.6	Drive telegram entry
SCSP/Dr[i]/Teleg/SigStatW/Conf	SigStatWConf[1..16]	omitted, Sercos Id: S-0-0026	Configuration signal status word
SCSP/Dr[i]/Teleg/SigStatW/Assign	SigStatWAssign[1..16]	omitted, Sercos Id: S-0-0328	Bit assignment signal status word
SCSP/Dr[i]/Teleg/SigCtrlW/Conf	SigCtrlWConf[1..16]	omitted, Sercos Id: S-0-0027	Configuration signal status word
SCSP/Dr[i]/Teleg/SigCtrlW/Assign	SigCtrlWAssign[1..16]	omitted, Sercos Id: S-0-0329	Bit assignment signal control word
/PLC/MachStatDiag/	MachStatDiagCycTime	2060 00090	Cycle time of machine status display (MZA)
/PLC/Plc/	IL	2060 00200	PLC selection
/PLC/Plc/IL/	PlcSymbFilePath	2060 00209	Path of the PLC application
/PLC/Plc/IL/	PlcMemIL	2060 00210	Minimum size of the PLC user program memory
/PLC/NcFuncBitIf/	NcFunc[1]	3020 00001	iChActFunc[1..24] NC function assignment \n(max. 8 characters)
/PLC/NcFuncBitIf/	NcFunc[1]	3020 00002	iChActFunc[1..24] NC function assignment \n(max. 8 additional characters)
/PLC/DigIO/	HSDigIOSynchrMode	4075 00100	Synchronization type of the high-speed I/Os
/PLC/DigIO/FuncAllocIn/	DigIOAllocIn[1]	4075 00101	Assignment of digital inputs
/PLC/DigIO/FuncAllocOut/	DigIOAllocOut[1]	4075 00102	Assignment of digital outputs
/PLC/AnalO/FuncAllocIn/	AnalOAllocIn[1]		
/PLC/AnalO/FuncAllocOut/	AnalOAllocOut[1]	4075 00104	Assignment of the analog outputs
/PLC/DigIO/	DigIOStartAddrIn	4075 00105	Initial address of digital inputs
/PLC/DigIO/	DigIOStartAddrOut	4075 00106	Initial address of digital outputs
/PLC/AnalO/	AnalOStartAddrIn		
/PLC/AnalO/	AnalOStartAddrOut		
/PLC/DigIO/	DigIOHw	4075 00200	High-speed I/O via Profibus DP or hardware
/PLC/AnalO/ConvOffs/	DigAnaConvOffs[1]	4075 00201	Offset of the D/A converter (analog outputs)
/PLC/AnalO/ConvGain/	DigAnaConvGain[1]	4075 00202	Gain of the D/A converter (analog outputs)

Appendix

Path	Parameter ID	Macoda No.	Description
/PLC/AnalO/ConvType/	DigAnaConvType[1]	4075 00203	Format of the D/A converter (analog outputs)
/PLC/Plc/ResetCpllf/Ch[i]/	CpllfStartBit	7070 00020	CPL customer signals in the bit interface
/PLC/Plc/ResetCpllf/Ch[i]/	CpllfEndBit		Signals that can be reset by the control reset.
/AUXF/Plclf/ChIndep/	AddrChIndepBcd32	2060 00003	Operand address (general)
/AUXF/Plclf/ChDep/Chlf[1]/	AddrChDepBit	2060 00009	Addresses of the channel-specific bit-coded auxiliary functions
/AUXF/Plclf/ChDep/Chlf[1]/	AddrChDepBcd32	2060 00010	Addresses of the channel-specific 32 bit BCD-coded auxiliary functions
/AUXF/Plclf/ChDep/Chlf[1]/	AddrChDepInt	2060 00012	Addresses of the channel-specific INTEGER-coded auxiliary functions
/AUXF/Plclf/ChDep/Chlf[1]/	AddrChDepFloat	2060 00013	Addresses of the channel-specific FLOAT-coded auxiliary functions
/AUXF/Plclf/ChDep/Chlf[1]/	AddrChDepDouble	2060 00014	Addresses of the channel-specific DOUBLE-coded auxiliary functions
/AUXF/Plclf/ChIndep/	AddrChIndepInt	2060 00020	Operand addresses: INTEGER, FLOAT, DOUBLE
/AUXF/Plclf/ChIndep/	AddrChIndepFloat		
/AUXF/Plclf/ChIndep/	AddrChIndepDouble		
[1] /AUXF/NofAuxFunc/	UsabAuxFunc	3010 00010	Number of auxiliary functions
[2] /AUXF/NofAuxFunc/ChDep/	ChDepBit		
[3] /AUXF/NofAuxFunc/ChIndep/	ChIndepBcd32		
[5] /AUXF/NofAuxFunc/ChDep/	ChDepBcd32		
[6] /AUXF/NofAuxFunc/ChIndep/	ChIndepInt		
[8] /AUXF/NofAuxFunc/ChIndep/	ChIndepFloat		
[9] /AUXF/NofAuxFunc/ChIndep/	ChIndepDouble		
[10] /AUXF/NofAuxFunc/ChDep/	ChDepInt		
[11] /AUXF/NofAuxFunc/ChDep/	ChDepFloat		
[12] /AUXF/NofAuxFunc/ChDep/	ChDepDouble		
/AUXF/OutBehav/	OutBehavMode	3010 00015	Auxiliary function output behavior

Path	Parameter ID	Macoda No.	Description
/AUXF/BitAckn/Byte000_015/	Byte000	3010 00020	Acknowledgement requirement for bit-coded auxiliary functions
/AUXF/BitAckn/Byte000_015/	Byte001		
/AUXF/BitAckn/Byte000_015/	Byte002		
/AUXF/BitAckn/Byte000_015/	Byte003		
/AUXF/BitAckn/Byte000_015/	Byte004		
/AUXF/BitAckn/Byte000_015/	Byte005		
/AUXF/BitAckn/Byte000_015/	Byte006		
/AUXF/BitAckn/Byte000_015/	Byte007		
/AUXF/BitAckn/Byte000_015/	Byte008		
/AUXF/BitAckn/Byte000_015/	Byte009		
/AUXF/BitAckn/Byte000_015/	Byte010		
/AUXF/BitAckn/Byte000_015/	Byte011		
/AUXF/BitAckn/Byte000_015/	Byte012		
/AUXF/BitAckn/Byte000_015/	Byte013		
/AUXF/BitAckn/Byte000_015/	Byte014		
/AUXF/BitAckn/Byte000_015/	Byte015		
/AUXF/BitAckn/Byte016_031/	Byte016		
/AUXF/BitAckn/Byte016_031/	Byte017		
/AUXF/BitAckn/Byte016_031/	Byte018		
/AUXF/BitAckn/Byte016_031/	Byte019		
/AUXF/BitAckn/Byte016_031/	Byte020		
/AUXF/BitAckn/Byte016_031/	Byte021		
/AUXF/BitAckn/Byte016_031/	Byte022		
/AUXF/BitAckn/Byte016_031/	Byte023		
/AUXF/BitAckn/Byte016_031/	Byte024		
/AUXF/BitAckn/Byte016_031/	Byte025		
/AUXF/BitAckn/Byte016_031/	Byte026		
/AUXF/BitAckn/Byte016_031/	Byte027		
/AUXF/BitAckn/Byte016_031/	Byte028		
/AUXF/BitAckn/Byte016_031/	Byte029		
/AUXF/BitAckn/Byte016_031/	Byte030		
/AUXF/BitAckn/Byte016_031/	Byte031		
/AUXF/BitAckn/Byte032_047/	Byte032		
/AUXF/BitAckn/Byte032_047/	Byte033		
/AUXF/BitAckn/Byte032_047/	Byte034		
/AUXF/BitAckn/Byte032_047/	Byte035		
/AUXF/BitAckn/Byte032_047/	Byte036		
/AUXF/BitAckn/Byte032_047/	Byte037		
/AUXF/BitAckn/Byte032_047/	Byte038		
/AUXF/BitAckn/Byte032_047/	Byte039		
/AUXF/BitAckn/Byte032_047/	Byte040		

Appendix

Path	Parameter ID	Macoda No.	Description
/AUXF/BitAckn/Byte032_047/	Byte041		
/AUXF/BitAckn/Byte032_047/	Byte042		
/AUXF/BitAckn/Byte032_047/	Byte043		
/AUXF/BitAckn/Byte032_047/	Byte044		
/AUXF/BitAckn/Byte032_047/	Byte045		
/AUXF/BitAckn/Byte032_047/	Byte046		
/AUXF/BitAckn/Byte032_047/	Byte047		
/AUXF/BitAckn/Byte048_063/	Byte048		
/AUXF/BitAckn/Byte048_063/	Byte049		
/AUXF/BitAckn/Byte048_063/	Byte050		
/AUXF/BitAckn/Byte048_063/	Byte051		
/AUXF/BitAckn/Byte048_063/	Byte052		
/AUXF/BitAckn/Byte048_063/	Byte053		
/AUXF/BitAckn/Byte048_063/	Byte054		
/AUXF/BitAckn/Byte048_063/	Byte055		
/AUXF/BitAckn/Byte048_063/	Byte056		
/AUXF/BitAckn/Byte048_063/	Byte057		
/AUXF/BitAckn/Byte048_063/	Byte058		
/AUXF/BitAckn/Byte048_063/	Byte059		
/AUXF/BitAckn/Byte048_063/	Byte060		
/AUXF/BitAckn/Byte048_063/	Byte061		
/AUXF/BitAckn/Byte048_063/	Byte062		
/AUXF/BitAckn/Byte048_063/	Byte063		
/AUXF/BitAckn/Byte064_079/	Byte064		
/AUXF/BitAckn/Byte064_079/	Byte065		
/AUXF/BitAckn/Byte064_079/	Byte066		
/AUXF/BitAckn/Byte064_079/	Byte067		
/AUXF/BitAckn/Byte064_079/	Byte068		
/AUXF/BitAckn/Byte064_079/	Byte069		
/AUXF/BitAckn/Byte064_079/	Byte070		
/AUXF/BitAckn/Byte064_079/	Byte071		
/AUXF/BitAckn/Byte064_079/	Byte072		
/AUXF/BitAckn/Byte064_079/	Byte073		
/AUXF/BitAckn/Byte064_079/	Byte074		
/AUXF/BitAckn/Byte064_079/	Byte075		
/AUXF/BitAckn/Byte064_079/	Byte076		
/AUXF/BitAckn/Byte064_079/	Byte077		
/AUXF/BitAckn/Byte064_079/	Byte078		
/AUXF/BitAckn/Byte064_079/	Byte079		
/AUXF/BitAckn/Byte080_095/	Byte080		

Path	Parameter ID	Macoda No.	Description
/AUXF/BitAckn/Byte080_095/	Byte081		
/AUXF/BitAckn/Byte080_095/	Byte082		
/AUXF/BitAckn/Byte080_095/	Byte083		
/AUXF/BitAckn/Byte080_095/	Byte084		
/AUXF/BitAckn/Byte080_095/	Byte085		
/AUXF/BitAckn/Byte080_095/	Byte086		
/AUXF/BitAckn/Byte080_095/	Byte087		
/AUXF/BitAckn/Byte080_095/	Byte088		
/AUXF/BitAckn/Byte080_095/	Byte089		
/AUXF/BitAckn/Byte080_095/	Byte090		
/AUXF/BitAckn/Byte080_095/	Byte091		
/AUXF/BitAckn/Byte080_095/	Byte092		
/AUXF/BitAckn/Byte080_095/	Byte093		
/AUXF/BitAckn/Byte080_095/	Byte094		
/AUXF/BitAckn/Byte080_095/	Byte095		
/AUXF/BitAckn/Byte096_111/	Byte096		
/AUXF/BitAckn/Byte096_111/	Byte097		
/AUXF/BitAckn/Byte096_111/	Byte098		
/AUXF/BitAckn/Byte096_111/	Byte099		
/AUXF/BitAckn/Byte096_111/	Byte100		
/AUXF/BitAckn/Byte096_111/	Byte101		
/AUXF/BitAckn/Byte096_111/	Byte102		
/AUXF/BitAckn/Byte096_111/	Byte103		
/AUXF/BitAckn/Byte096_111/	Byte104		
/AUXF/BitAckn/Byte096_111/	Byte105		
/AUXF/BitAckn/Byte096_111/	Byte106		
/AUXF/BitAckn/Byte096_111/	Byte107		
/AUXF/BitAckn/Byte096_111/	Byte108		
/AUXF/BitAckn/Byte096_111/	Byte109		
/AUXF/BitAckn/Byte096_111/	Byte110		
/AUXF/BitAckn/Byte096_111/	Byte111		
/AUXF/BitAckn/Byte112_127/	Byte112		
/AUXF/BitAckn/Byte112_127/	Byte113		
/AUXF/BitAckn/Byte112_127/	Byte114		
/AUXF/BitAckn/Byte112_127/	Byte115		
/AUXF/BitAckn/Byte112_127/	Byte116		
/AUXF/BitAckn/Byte112_127/	Byte117		
/AUXF/BitAckn/Byte112_127/	Byte118		
/AUXF/BitAckn/Byte112_127/	Byte119		
/AUXF/BitAckn/Byte112_127/	Byte120		

Appendix

Path	Parameter ID	Macoda No.	Description
/AUXF/BitAckn/Byte112_127/	Byte121		
/AUXF/BitAckn/Byte112_127/	Byte122		
/AUXF/BitAckn/Byte112_127/	Byte123		
/AUXF/BitAckn/Byte112_127/	Byte124		
/AUXF/BitAckn/Byte112_127/	Byte125		
/AUXF/BitAckn/Byte112_127/	Byte126		
/AUXF/BitAckn/Byte112_127/	Byte127		
/AUXF/BitAckn/Byte128_143/	Byte128		
/AUXF/BitAckn/Byte128_143/	Byte129		
/AUXF/BitAckn/Byte128_143/	Byte130		
/AUXF/BitAckn/Byte128_143/	Byte131		
/AUXF/BitAckn/Byte128_143/	Byte132		
/AUXF/BitAckn/Byte128_143/	Byte133		
/AUXF/BitAckn/Byte128_143/	Byte134		
/AUXF/BitAckn/Byte128_143/	Byte135		
/AUXF/BitAckn/Byte128_143/	Byte136		
/AUXF/BitAckn/Byte128_143/	Byte137		
/AUXF/BitAckn/Byte128_143/	Byte138		
/AUXF/BitAckn/Byte128_143/	Byte139		
/AUXF/BitAckn/Byte128_143/	Byte140		
/AUXF/BitAckn/Byte128_143/	Byte141		
/AUXF/BitAckn/Byte128_143/	Byte142		
/AUXF/BitAckn/Byte128_143/	Byte143		
/AUXF/BitAckn/Byte144_159/	Byte144		
/AUXF/BitAckn/Byte144_159/	Byte145		
/AUXF/BitAckn/Byte144_159/	Byte146		
/AUXF/BitAckn/Byte144_159/	Byte147		
/AUXF/BitAckn/Byte144_159/	Byte148		
/AUXF/BitAckn/Byte144_159/	Byte149		
/AUXF/BitAckn/Byte144_159/	Byte150		
/AUXF/BitAckn/Byte144_159/	Byte151		
/AUXF/BitAckn/Byte144_159/	Byte152		
/AUXF/BitAckn/Byte144_159/	Byte153		
/AUXF/BitAckn/Byte144_159/	Byte154		
/AUXF/BitAckn/Byte144_159/	Byte155		
/AUXF/BitAckn/Byte144_159/	Byte156		
/AUXF/BitAckn/Byte144_159/	Byte157		
/AUXF/BitAckn/Byte144_159/	Byte158		
/AUXF/BitAckn/Byte144_159/	Byte159		
/AUXF/BitAckn/Byte160_175/	Byte160		

Path	Parameter ID	Macoda No.	Description
/AUXF/BitAckn/Byte160_175/	Byte161		
/AUXF/BitAckn/Byte160_175/	Byte162		
/AUXF/BitAckn/Byte160_175/	Byte163		
/AUXF/BitAckn/Byte160_175/	Byte164		
/AUXF/BitAckn/Byte160_175/	Byte165		
/AUXF/BitAckn/Byte160_175/	Byte166		
/AUXF/BitAckn/Byte160_175/	Byte167		
/AUXF/BitAckn/Byte160_175/	Byte168		
/AUXF/BitAckn/Byte160_175/	Byte169		
/AUXF/BitAckn/Byte160_175/	Byte170		
/AUXF/BitAckn/Byte160_175/	Byte171		
/AUXF/BitAckn/Byte160_175/	Byte172		
/AUXF/BitAckn/Byte160_175/	Byte173		
/AUXF/BitAckn/Byte160_175/	Byte174		
/AUXF/BitAckn/Byte160_175/	Byte175		
/AUXF/BitAckn/Byte176_191/	Byte176		
/AUXF/BitAckn/Byte176_191/	Byte177		
/AUXF/BitAckn/Byte176_191/	Byte178		
/AUXF/BitAckn/Byte176_191/	Byte179		
/AUXF/BitAckn/Byte176_191/	Byte180		
/AUXF/BitAckn/Byte176_191/	Byte181		
/AUXF/BitAckn/Byte176_191/	Byte182		
/AUXF/BitAckn/Byte176_191/	Byte183		
/AUXF/BitAckn/Byte176_191/	Byte184		
/AUXF/BitAckn/Byte176_191/	Byte185		
/AUXF/BitAckn/Byte176_191/	Byte186		
/AUXF/BitAckn/Byte176_191/	Byte187		
/AUXF/BitAckn/Byte176_191/	Byte188		
/AUXF/BitAckn/Byte176_191/	Byte189		
/AUXF/BitAckn/Byte176_191/	Byte190		
/AUXF/BitAckn/Byte176_191/	Byte191		
/AUXF/ClassDef/Class[1]/	AuxFuncName	3010 00030	Names of auxiliary functions
/AUXF/ClassDef/Class[1]/	EnablAuxFunc	3010 00040	Classes of auxiliary functions
/AUXF/ClassDef/Class[1]/	AuxFuncChDep		
/AUXF/ClassDef/Class[1]/	Bit		
/AUXF/ClassDef/Class[1]/	Bcd32		
/AUXF/ClassDef/Class[1]/	Int		
/AUXF/ClassDef/Class[1]/	Float		
/AUXF/ClassDef/Class[1]/	Double		

Appendix

Path	Parameter ID	Macoda No.	Description
/AUXF/ClassDef/Class[1]/Bit/ /AUXF/ClassDef/Class[1]/Bcd32/	StartBitAddr PreIPlaceBcd32	3010 00060	BCD format integer places, start bit address or bit address for individually defined Auxiliary functions
/AUXF/ClassDef/Class[1]/Bit/ /AUXF/ClassDef/Class[1]/Bcd32/	EndBitAddr DeciPlaceBcd32	3010 00070	BCD format of decimal positions, end bit addresses or channel numbers
/AUXF/ClassDef/Class[1]/Bit/ /AUXF/ClassDef/Class[1]/Bcd32/ /AUXF/ClassDef/Class[1]/Int/ /AUXF/ClassDef/Class[1]/Float/ /AUXF/ClassDef/Class[1]/Double/	StartCode IndBcd32 IndInt IndFloat IndDouble	3010 00080	Index of non-bit-coded auxiliary functions, starting code for bit-coded auxiliary functions
/AUXF/ClassDef/Class[1]/Bit/ /AUXF/ClassDef/Class[1]/Bcd32/ /AUXF/ClassDef/Class[1]/Int/ /AUXF/ClassDef/Class[1]/Float/ /AUXF/ClassDef/Class[1]/Double/	EndCode AcknBcd32 AcknInt AcknFloat AcknDouble	3010 00090	Acknowledgement requirement for non-bit-coded auxiliary functions or end code for bit-coded auxiliary functions
[1] /AUXF/GrDef/Gr[1]/ [2-16] /AUXF/GrDef/Gr[1]/ ... [1] /AUXF/GrDef/Gr[64]/ [2..16] /AUXF/GrDef/Gr[64]/	GrChDep GrAuxFunc[1..15] GrChDep GrAuxFunc[1..15]	3010 01000 3010 01001 3010 01002 3010 01003 ... 63	Auxiliary function group 1 Auxiliary function group 2 Auxiliary function group 3 Auxiliary function group 4 ... 64
/GUI/ChDisp/	ActCh	6001 00010	Active channel displayed after switch-on
/GUI/ChDisp/Suppr/Ch[1]/	SupprChDisp	6005 00010	Suppress display of channel
/GUI/PosDisp/	UpdateTimePosDisp	9030 00002	Updating actual workpiece coordinate values
/GUI/PosDisp/Ord/Ch[1]/AddWin/ AddAx/Dr[1]/ /GUI/PosDisp/Ord/Ch[1]/AddWin/ AddAx/Dr[1]/	ModeAddAx PrioAddAx	6005 00020	Position of the axis in the total sequence
/GUI/PosDisp/Ord/Ch[1]/AddWin/ ChCoordPack/Dr[1]/ /GUI/PosDisp/Ord/Ch[1]/AddWin/ ChCoordPack/Dr[1]/	ModeChPack PrioChPack	6005 00021	Position of the CCP in the total sequence
/GUI/PosDisp/Ord/Ch[1]/ChWin/ Dr[1]/ /GUI/PosDisp/Ord/Ch[1]/ChWin/ Dr[1]/	ModeAxChWin PrioAxChWin	6005 00022	Position of the axis in the channel sequence
/GUI/PosDisp/Ord/Ch[1]/	NofAxPosDisp	6005 00023	Maximum number of axes in axis display
/GUI/TechDisp/Ch[1]/ SpSpGrDisp[1]/SpDisp/ /GUI/TechDisp/Ch[1]/ SpSpGrDisp[1]/SpGrDisp/	SpIndSpDisp SpGrIndSpGrDisp	6005 00030	Configuration of spindle display
/GUI/NcCodeDisp/Ch[1]/	NcCodeGr[1]	6005 00040	Display filter for modal NC functions

Path	Parameter ID	Macoda No.	Description
/GUI/AuxFuncDisp/Ch[1]/	AuxCodeGr[1]	6005 00050	Display filter for modal auxiliary functions
/GUI/ZotDisp/Ch[1]/	NcCodeZotSet[1]	6005 00060	Configuration of the zero offset shift display
/GUI/ZotDisp/Ch[1]/	NofZotSets	6005 00061	Number of banks used for zero point offset
/GUI/PmtDisp/Ch[1]/	NcCodePmtSet[1]	6005 00070	Configuration of placement display
/GUI/PmtDisp/Ch[1]/	NofPmtSets	6005 00071	Number of banks used for placements
/GUI/PosDisp/CoordSys/Ch[1]/	WcsPosDisp	6005 00100	Coordinate systems allowed for display
/GUI/PosDisp/CoordSys/Ch[1]/	McsPosDisp		
/GUI/PosDisp/CoordSys/Ch[1]/	BcsPosDisp		
/GUI/PosDisp/CoordSys/Ch[1]/	AcsPosDisp		
/GUI/TechDisp/FeedDisp/	PrecFeedMetr	6020 00001	Number of decimal places for feed display (metric)
/GUI/TechDisp/FeedDisp/	PrecFeedInch	6020 00002	Number of decimal places for feed display (inch)
/GUI/PosDisp/Prec/Dr[1]/	PrecPosMetr	6020 00011	Number of decimal places for axis display (metric)
/GUI/PosDisp/Prec/Dr[1]/	PrecPosInch	6020 00012	Number of decimal places for axis display (inch)
/GUI/TabDisp/	PrecLinMetr	6020 00021	Number of decimal places for correction data (metric)
/GUI/TabDisp/	PrecRotMetr		
/GUI/TabDisp/	PrecLinInch	6020 00022	Number of decimal places for correction data (inch)
/GUI/TabDisp/	PrecRotInch		
/GUI/DispUnit		6020 00030	Measuring unit for the display
/NCP/SubProg/NonModalGFunc/ GFunc[1]/	GFuncCode	3090 00001	Non-modal G-functions (G-codes)
/NCP/SubProg/NonModalGFunc/ GFunc[1]/	GFuncSubProg	3090 00002	Non-modal G-functions (subroutine names)
/NCP/SubProg/NonModalMFunc/ MFunc[1]/	MFuncCode	3090 00003	Non-modal M-functions (M-codes)
/NCP/SubProg/NonModalMFunc/ MFunc[1]/	MFuncSubProg	3090 00004	Non-modal M-functions (subroutine names)
/NCP/SubProg/ModalSubProg/ SwiOff/ /NCP/SubProg/ModalSubProg/ SwiOn[1]/	SwiOffSynSubProg SwiOnSynSubProg	3090 00005	Modal subroutines (NC syntax)
/NCP/SubProg/ModalSubProg/ SwiOn[1]/	NameSubProg	3090 00006	Modal subroutines (subroutine names)
/NCP/SubProg/ModalSubProg/ SwiOn[1]/	NofParSubProg	3090 00007	Modal subroutines (number of parameters)
/NCP/DiaProg/Ch[1]/	DiaCoord[1]	7010 00210	Diameter coordinates

Appendix

Path	Parameter ID	Macoda No.	Description
/NCP/CornRound/Ch[1]/	MaxDev	7050 00110	Corner rounding function: maximum permissible deviation from programmed contour
/NCP/CornRound/Ch[1]/	MinLen	7050 00120	Corner rounding function: Minimum block length for corner rounding
/NCP/CornRound/Ch[1]/	MaxAng	7050 00130	Corner rounding function: maximum angle for quasi-continuous block transitions
/NCP/PolarCoord/	PolarCoordAngle1	8005 00001	NC syntax for polar angle 1
/NCP/PolarCoord/	PolarCoordAngle2	8005 00002	NC syntax for polar angle 2
/NCP/CutComp/	CutCompActDeact-Mode	8009 00001	Starting and exiting correction build-up strategy for tool path correction
/NCP/DiaProg/	IncrSetPointsDiaProg	8010 00001	Effect of incremental specifications when programming the diameter
/NCO/InPosWin/Dr[1]/	InPosRough	1015 00100	In-position window
/NCO/FileOrg/	SrchPathSubProg	3080 00001	Search path for subroutines
/NCO/Mem/SysMem/	BuffSizeNfs	3080 00002	Buffer size for processing via NFS
/NCO/FileOrg/	SrchPathLinkTab	3080 00004	Directory name for link tables
/NCO/FileOrg/	EnablEqProgName-sLinkOpti	3080 00010	Enable link process optimization (same project names)
/NCO/WebServ/	EnablWebServ	4090 00001	Enable web server
/NCO/CommServ/	EnablCommServ	4090 00100	Enable communication server
/NCO/PathVel/Ch[1]/	KeepPathVelNoIpoAct	7030 00500	Keep velocity in motion-neutral NC blocks
/NCO/VelScaleFact/Ch[1]/	FactVel	7040 00010	Unit of feed programmed in metric
/NCO/VelScaleFact/Ch[1]/	FactVelPerRev	7040 00020	Unit for programmed feed per revolution
[1] /NCO/VelScaleFact/Ch[1]/ [2] /NCO/VelScaleFact/Ch[1]/	FactRotInch FactRotMetr	7040 00110	Scaling of velocity of rotary axis with G70 or G71
/NCO/CircleAccur/Ch[1]/CentProg/	RadTolCentProg	7050 00020	Radius tolerance with center point programming
/NCO/CircleAccur/Ch[1]/RadProg/	RadTolRadProg	7050 00030	Radius tolerance with radius programming
/NCO/InPosEval/Ch[1]/	InPosEvalMode	7050 00500	Mode of the In-Position evaluation
/NCO/LookAh/Ch[1]/	NofBlkPrep	7060 00110	Number of all blocks for look-ahead including interpolation
[1] /NCO/LookAh/Ch[1]/MaxBlk/ [2] /NCO/LookAh/Ch[1]/MaxBlk/ [3] /NCO/LookAh/Ch[1]/MaxBlk/ [4] /NCO/LookAh/Ch[1]/MaxBlk/ [5] /NCO/LookAh/Ch[1]/MaxBlk/	MaxBlkVelCalc MaxBlkBuffered MaxBlkSpline MaxBlkMiscFunc MaxBlkTCorr3D	7060 00120	Maximum number of blocks block look-ahead for different NC functions

Path	Parameter ID	Macoda No.	Description
[1]/NCO/LookAh/Ch[1]/PercSplit/ [2] /NCO/LookAh/Ch[1]/PercSplit/ [3] /NCO/LookAh/Ch[1]/PercSplit/ [4] /NCO/LookAh/Ch[1]/PercSplit/ [5] /NCO/LookAh/Ch[1]/PercSplit/	PercVelCalc PercBuffered PercSpline PercMiscFunc PercTCorr3D	7060 00130	Number of blocks in percentage for look ahead for different NC functions
/NCO/FileOrg/ChFileOrg/Ch[1]/	MaxOpenFile	7060 00210	Number of open files per channel
/NCO/Mem/ChMem/Ch[1]/	BuffSizeExternBlk	7060 00510	Size of buffer for NC block selection
/NCO/Mem/ChMem/Ch[1]/	NofBlkRetrace	7060 00610	Number of blocks retrace mode
/NCO/Mem/ChMem/Ch[1]/ /NCO/Mem/ChMem/Ch[1]/	CplStack CplLinkBuff	7070 00010	Data areas for CPL programs
/NCO/FileOrg/ChFileOrg/Ch[1]/	KindOfLinking	7070 00030	Linking type
/NCO/PrecProg/	ContDevTol	8003 00001	Contour deviation tolerance with precision programming
/NCO/BlkTransferLock/	StandstModeBlkTransferLock	8011 00001	Standstill type at block end for block transfer inhibit
/NCO/CorrUnit/	SysCorrUnit	9020 00010	Dimensions of offsets and corrections (tables and external)
/NCO/CorrUnit/	AsynchrAxUnit	9020 00020	Default measuring unit for asynchronous linear axes
/NCO/Perf/	CpuTimeBlkPrep	9040 00010	CPU time for block processing
/NCO/DateTime/Winter/ /NCO/DateTime/Winter/ /NCO/DateTime/Winter/ /NCO/DateTime/Summer/ /NCO/DateTime/Summer/ /NCO/DateTime/Summer/ /NCO/DateTime/SwiWinter/ /NCO/DateTime/SwiWinter/ /NCO/DateTime/SwiSummer/ /NCO/DateTime/SwiSummer/	WinterZoneName WinterLoc WinterDiffUtc SummerZoneName SummerLoc SummerDiffUtc SwiWinterDate SwiWinterTime SwiSummerDate SwiSummerTime	9050 00001	Time zone
/NCO/DbTables/	DbTabEventBuffSize	9050 00010	Buffer size for change event of the DB tables
/OPF/ReEntry/	ReEntryFuncMode	3080 00300	Function mode for re-entry data
/OPF/ReEntry/	ReEntryDataPath	3080 00301	Directory for re-entry data
/OPF/WcsJog/Ch[1]/Vel/WcsJog/ /OPF/WcsJog/Ch[1]/Vel/WcsJog/ /OPF/WcsJog/Ch[1]/Vel/WcsJog/ /OPF/WcsJog/Ch[1]/Vel/WcsJog/ /OPF/WcsJog/Ch[1]/Vel/WcsJog/ /OPF/WcsJog/Ch[1]/Vel/WcsJog/	SlowVelWcs MedVelWcs FastVelWcs RapidVelWcs DefStepVelWcs VarStepVelWcs	7050 01000	Jog WCS: Manual feeds

Appendix

Path	Parameter ID	Macoda No.	Description
/OPF/WcsJog/Ch[1]/Vel/	WcsJogIf	7050 01010	Jog WCS: Feed and incremental step, system axis number
/OPF/WcsJog/Ch[1]/	IncrWcsJog	7050 01020	Jog WCS: resolution of an increment
/OPF/WcsJog/Ch[1]/	DistWcsJog	7050 01030	Jog WCS: Variable increment step
/OPF/WcsJog/Ch[1]/	TrafoProgName	7050 01100	Jog WCS: Program name for default configuration
/OPF/WcsJog/Ch[1]/	CoordProgName	7050 01110	Jog WCS: program name for coordinate selection
/OPF/ReturnPath/Ch[1]/	ReturnMode	7060 00310	"Return to contour" operation mode
/OPF/ReturnPath/Ch[1]/	ReturnPoint	7060 00320	Approach point: Return to contour
/OPF/ReturnPath/Ch[1]/	NofJogSteps	7060 00330	Maximum records for return to path in blocks
/OPF/Handw/Wh[1]/	IncrPerRev	9060 00001	Number of increments per revolution
/SFU/AxDistCtrl/Dr[1]/ /SFU/AxDistCtrl/Dr[1]/	EnablAxDistCtrl DistEndPoint	1015 00010	Distance To End Point
/SFU/WayPoint/Point[1]/ /SFU/WayPoint/Point[1]/	DrIndWayPoint MonTypeWayPoint	2010 00100	Monitored axis
/SFU/WayPoint/Point[1]/	PosWayPoint	2010 00110	Position of the point
/SFU/WayPoint/Point[1]/	IfSigWayPoint	2010 00120	Interface signal of the point
/SFU/WayPoint/Point[1]/	MonTypeWayPoint	2010 00130	Cyclical monitoring of rotary axes
/SFU/AreaCtrl/Area[1]/Dim[1]/	DrIndAreaCtrl	8002 00001	Area monitoring: System axis number of the first area dimension
/SFU/AreaCtrl/Area[1]/Dim[2]/	DrIndAreaCtrl	8002 00002	Area monitoring: System axis number of the second area dimension
/SFU/AreaCtrl/Area[1]/Dim[1]/	CentPointAreaCtrl	8002 00011	Area monitoring: Center of the area, first dimension
/SFU/AreaCtrl/Area[1]/Dim[2]/	CentPointAreaCtrl	8002 00012	Area monitoring: Center of the area, second dimension
/SFU/AreaCtrl/Area[1]/Dim[1]/	ExpandAreaCtrl	8002 00021	Area monitoring: Expansion of the area, first dimension
/SFU/AreaCtrl/Area[1]/Dim[2]/	ExpandAreaCtrl	8002 00022	Area monitoring: Expansion of the area, second dimension
/SFU/AreaCtrl/Area[1]/	AreaTypeAreaCtrl	8002 00031	Area monitoring: monitoring area type
/SFU/AreaCtrl/Area[1]/	ProgModeAreaCtrl	8002 00032	Area monitoring: area is programmable
/SFU/CollMon/Gr[1]/	EnablCollMon	8008 00001	One-dimensional collision monitoring: Activating
/SFU/CollMon/Gr[1]/	DrIndMonAx1	8008 00002	One-dimensional collision monitoring: Axis 1
/SFU/CollMon/Gr[1]/	DrIndMonAx2	8008 00003	One-dimensional collision monitoring: Axis 2
/SFU/CollMon/Gr[1]/	DirCollMon	8008 00004	One-dimensional collision monitoring: Traveling direction

Path	Parameter ID	Macoda No.	Description
/SFU/CollMon/Gr[1]/	ActUpLim	8008 00005	One-dimensional collision monitoring: monitoring the upper limit value
/SFU/CollMon/Gr[1]/	ActLowLim	8008 00006	One-dimensional collision monitoring: monitoring the lower limit value
/SFU/CollMon/Gr[1]/	PosUpLim	8008 00007	One-dimensional collision monitoring: Upper limit value
/SFU/CollMon/Gr[1]/	PosLowLim	8008 00008	One-dimensional collision monitoring: Lower limit value
/SAFE/Dr[1]/	EnablSafeTech	1001 00002	Enabling of safety systems possible
/SAFE/Dr[1]/	SupprSafeTechNc	1001 00003	Hide safety engineering in NC
/SAFE/Dr[1]/	VelWeightFact	1001 00004	Safety technology: safe speed scaling factor
/COUP/AxCouple/Dr[1]/	MaxFollErr	1003 00060	Max. following error for coupled axes
/COUP/AxCouple/Dr[1]/Gantry/	MastInd	1003 00061	Index of the gantry master axis
/COUP/AxCouple/Dr[1]/Gantry/	MaxGrTorq	1003 00062	Max. total static torque of a gantry group
/COUP/AxCouple/Dr[1]/Gantry/	EnablSynchrMon	1003 00063	Synchronous run monitoring when referencing gantry
/COUP/AxCouple/Dr[1]/Gantry/	SynchrMonWin	1003 00064	Synchronous operation monitoring window when referencing gantry
/COUP/AxCouple/Dr[1]/Gantry/	EnablRefSwiMon	1003 00065	Reference switch monitoring when referencing gantry
/COUP/AxCouple/Dr[1]/Gantry/	CmdOffslpoMode	1003 00066	Compensate for gantry command value offset
/COUP/AxCouple/Dr[1]/Gantry/	MaxRedCmdOffs	1003 00067	Compensate for gantry command value offset: Upper limit [mm]
/COUP/AxCouple/Dr[1]/SupplAx/	MastIndSupplAx	1003 00080	Additional axes: System axis number of the master
/DIG/	NofDigBuff	8006 00001	Number of digitizing buffers (buffer size 512 bytes)
/DIG/	NofReservBuff	8006 00002	Number of reserve buffers
/MEAS/Dr[1]/	Probe1Edge	1003 00011	Edge of touch probe 1
/MEAS/Dr[1]/	EnablProbeFunc	1003 00012	Touch probe function available
/MEAS/Dr[1]/	Probe2Edge	1003 00013	Edge of touch probe 2
/PUN/	EnablPunch	8001 00010	Application of punching and nibbling
/PUN/StrokeRelTime/Dr[1]/	TimeOffs	8001 00020	Axis-specific stroke release times
/PUN/StrokeRelTime/Dr[1]/	TimeRef	8001 00021	Referencing stroke release times
/PUN/	StrokeMonTime	8001 00030	Stroke check time

Appendix

Path	Parameter ID	Macoda No.	Description
/ONL/McsOnlCorr/Ch[1]/	EnablMcsCorr	7050 00700	Axis distance control exists
/ONL/McsOnlCorr/Ch[1]/Base/ /ONL/McsOnlCorr/Ch[1]/Base/	ChAxMcsCorr DirMcsCorr	7050 00702	Axis distance control: number of the channel axis and correction direction
/ONL/McsOnlCorr/Ch[1]/Base/	ChIfSigMcsCorr	7050 00708	Axis distance control: Selectable interface signal
/ONL/McsOnlCorr/Ch[1]/Device/ /ONL/McsOnlCorr/Ch[1]/Device/	Dr CplVar	7050 00710	Axis distance control: Sensor selection, device group
/ONL/McsOnlCorr/Ch[1]/Device/Dr/	DrIndMcsCorr	7050 00712	Axis distance control: sensor selection, device number
/ONL/McsOnlCorr/Ch[1]/Device/Dr/ /ONL/McsOnlCorr/Ch[1]/Device/ CplVar/	ScsParMcsCorr CplVarMcsCorr	7050 00714	Axis distance control: sensor selection, special identification
/ONL/McsOnlCorr/Ch[1]/IncrScale/	FactMcsCorr	7050 00720	Axis distance control: sensor increments, factor
/ONL/McsOnlCorr/Ch[1]/IncrScale/	DivMcsCorr	7050 00721	Axis distance control: sensor increments, divisor
/ONL/McsOnlCorr/Ch[1]/Lim/	FilterMcsCorr	7050 00730	Axis distance control: filter, smoothing time
/ONL/McsOnlCorr/Ch[1]/Lim/	MaxVelMcsCorr	7050 00740	Axis distance control: limiter, speed
/ONL/McsOnlCorr/Ch[1]/Lim/	MaxAccMcsCorr	7050 00741	Axis distance control: limiter, acceleration
/ONL/McsOnlCorr/Ch[1]/Det/	CollDet	7050 00750	Axis distance control: monitoring collision detection
/ONL/McsOnlCorr/Ch[1]/Det/	HoleDet	7050 00752	Axis distance control: monitoring of hole detection
/ONL/WcsOnlCorr/Ch[1]/	EnablWcsCorr	7050 00900	Online correction in workpiece coordinates (WCS) available
/ONL/WcsOnlCorr/Ch[1]/Device/ /ONL/WcsOnlCorr/Ch[1]/Device/ /ONL/WcsOnlCorr/Ch[1]/Device/	Dr CplVar Handw	7050 00910	Online correction WCS: Sensor selection, device group
/ONL/WcsOnlCorr/Ch[1]/Device/Dr/ /ONL/WcsOnlCorr/Ch[1]/Device/ Handw/	DrIndWcsCorr HandwWcsCorr	7050 00912	Online correction WCS: sensor selection, device number
[1] /ONL/WcsOnlCorr/Ch[1]/ Device/Dr/ [2] /ONL/WcsOnlCorr/Ch[1]/Device/ CplVar/	ScsParWcsCorr CplVarWcsCorr	7050 00914	Online correction WCS: Sensor selection, identification
/ONL/WcsOnlCorr/Ch[1]/IncrScale/	FactWcsCorr	7050 00920	Online correction WCS: sensor increments, factor
/ONL/WcsOnlCorr/Ch[1]/IncrScale/	DivWcsCorr	7050 00921	Online correction WCS: sensor increments, divisor
/ONL/WcsOnlCorr/Ch[1]/ IncrSource/	DrIndIncrSource	7050 00926	Online correction WCS: Incremental step, system axis number

Path	Parameter ID	Macoda No.	Description
/ONL/WcsOnlCorr/Ch[1]/Lim/	MaxVelWcsCorr	7050 00940	Online correction WCS: max. velocity
/ONL/WcsOnlCorr/Ch[1]/Lim/	MaxAccWcsCorr	7050 00941	Online correction WCS: max. acceleration
/MPS/Dr[1]/	EnablMovePoStop	1003 00030	Positive stop can be enabled
/MPS/Dr[1]/	MaxTorqPoStop	1003 00031	Torque limit positive stop
/MPS/Dr[1]/	MaxDevPoStop	1003 00032	Monitoring window for positive stop
/MPS/Dr[1]/	MaxVelPoStop	1005 00030	Maximum feed, move to positive stop
/MPS/Dr[1]/	MaxAccPoStop	1010 00030	Maximum acceleration, move to positive stop
/SPL/	MaxPosDevSpline	8007 00010	Maximum Allowed Starting Point/Endpoint Deviation
/SPL/	MaxVelDevSpline	8007 00020	Maximum Allowable Deviation for the Path Velocity
[1] /SPL/ [2] /SPL/ [3] /SPL/	SplineType NofOverlapBlk VectOriSpline	8007 00099	Spline configuration
/SPL/	BSplineApproxTol	8007 00030	Dimensional tolerance for B-spline approximation (preset)
/TTL/Ch[1]/	RotAxTangT	7050 00210	Tangential tool control: Rotary axis number
/TTL/Ch[1]/ /TTL/Ch[1]/	NofEdgeTangT DirTangT	7050 00220	Tangential tool control: Tool symmetry
/TTL/Ch[1]/	BlkAngTangT	7050 00230	Tangential tool control: Intermediate block angle
/TTL/Ch[1]/	OffAngTangT	7050 00240	Tangential tool control: Setting angle
/TTL/Ch[1]/	StartAngTangT	7050 00250	Tangential tool control: Initialization of the setting angle
/TTL/Ch[1]/	NcPlcComTangT	7050 00260	Tangential tool control: NC-PLC communication for the intermediate block
/THRD/Ch[1]/ModeSel/ /THRD/Ch[1]/ModeSel/	AxModeThrd SpModeThrd	7050 00600	Thread cutting: position-controlled spindle, programming type
/THRD/Ch[1]/Vel/	VelJumpThrd	7050 00610	Thread cutting: dynamics, jump velocity
/THRD/Ch[1]/Acc/ /THRD/Ch[1]/Acc/	AccThrd AccJumpThrd	7050 00615	Thread cutting: dynamics, acceleration
/THRD/Ch[1]/Dec/ /THRD/Ch[1]/Dec/	DecThrd DecJumpThrd	7050 00620	Thread cutting: dynamics, deceleration
/THRD/Ch[1]/Retr/	EnablRetrThrd	7050 00640	Thread cutting: fast retract, enable

Appendix

Path	Parameter ID	Macoda No.	Description
/THRD/Ch[1]/Retr/ /THRD/Ch[1]/Retr/	DrIndRetrAx DrIndSeconAx	7050 00645	Thread cutting: fast retract, system axis numbers
/THRD/Ch[1]/Retr/ /THRD/Ch[1]/Retr/	RetrVectRetrAx RetrVectSeconAx	7050 00650	Thread cutting: fast retract, distance and direction
/TLWP/TCorr/Ch[1]/ /TLWP/TCorr/Ch[1]/ /TLWP/TCorr/Ch[1]/	WcsFixed WcsActPlane Tcs	7050 01300	Tool correction: Mode
/TLWP/TCorr/Ch[1]/WcsFixed/ TLen[1]/ /TLWP/TCorr/Ch[1]/WcsFixed/ TLen[1]/ /TLWP/TCorr/Ch[1]/Tcs/TLen[1]/ /TLWP/TCorr/Ch[1]/Tcs/TLen[1]/	ChAxTCorrWcs DirTCorrWcs CoordTCorrTcs DirTCorrTcs	7050 01310	Tool correction: assignment
/TRA/AxTrafo[1]/	TrafoType	1030 00110	Axis transformation type
/TRA/AxTrafo[1]/FwdInCoord/	FwdInCoordIndTrafo[1]	1030 00120	System axes/coordinates of transformation (-6, -5, -4, -3, -2, -1, 1 .. 64)
/TRA/AxTrafo[1]/FwdInCoordClass/ FwdInCoord[1]/ /TRA/AxTrafo[1]/FwdInCoordClass/ FwdInCoord[1]/	CoordClass CoordDir	1030 00125	Axis classification of transformation axes
/TRA/AxTrafo[1]/RefPos/	RefPosTrafo[1]	1030 00130	Axis positions of the reference position (zero position of coordinates)
/TRA/AxTrafo[1]/JointPar/	JointParTrafo[1]	1030 00140	Length and angle parameters
/TRA/AxTrafo[1]/TcsRefOri/TcsX/ /TRA/AxTrafo[1]/TcsRefOri/TcsX/ /TRA/AxTrafo[1]/TcsRefOri/TcsY/ /TRA/AxTrafo[1]/TcsRefOri/TcsY/ /TRA/AxTrafo[1]/TcsRefOri/TcsZ/ /TRA/AxTrafo[1]/TcsRefOri/TcsZ/	CoordTcsXBcs DirTcsXBcs CoordTcsYBcs DirTcsYBcs CoordTcsZBcs DirTcsZBcs	1030 00150	Reference position of tool coordinate system
/TRA/AxTrafo[1]/	EpsAxTrafo	1030 00160	Epsilon environment
/VDO/Ch[1]/Out/ /VDO/Ch[1]/Out/	AnaOutUpVelLim AnaOutLowVelLim	7050 00810	Laser power control: power limit values
/VDO/Ch[1]/Vel/ /VDO/Ch[1]/Vel/	UpVelLim LowVelLim	7050 00815	Laser power control: speed limit values
/VDO/Ch[1]/	CoordSel	7050 00820	Laser power control: Coordinate selection

Tab. 34-1: List of all parameters, with reference to MACODA No.

35 Service and support

Our worldwide service network provides an optimized and efficient support. Our experts offer you advice and assistance should you have any queries. You can contact us **24/7**.

Service Germany

Our technology-oriented Competence Center in Lohr, Germany, is responsible for all your service-related queries for electric drive and controls.

Contact the **Service Hotline** and **Service Helpdesk** under:

Phone: **+49 9352 40 5060**
Fax: **+49 9352 18 4941**
E-mail: service.svc@boschrexroth.de
Internet: <http://www.boschrexroth.com>

Additional information on service, repair (e.g. delivery addresses) and training can be found on our internet sites.

Service worldwide

Outside Germany, please contact your local service office first. For hotline numbers, refer to the sales office addresses on the internet.

Preparing information

To be able to help you more quickly and efficiently, please have the following information ready:

- Detailed description of malfunction and circumstances
- Type plate specifications of the affected products, in particular type codes and serial numbers
- Your contact data (phone and fax number as well as your e-mail address)

Index

Symbols

/SysAdvDistSigA.....	429
/SysAdvDistSigE.....	428

0 ... 9

6.6 Time interval to the programmed Reset mark ADSLAB.....	425
6.6 Time interval to the programmed Set mark ADSLAB.....	424
32-bit-BCD-coded auxiliary functions.....	200
1001 00001.....	51, 57, 62
1003 00001.....	56
1001 00002.....	341
1003 00002.....	61
1005 00002.....	93
1001 00003.....	342
1005 00003.....	103
1001 00004.....	342
1003 00004.....	60, 106
1005 00004.....	103
1003 00005.....	107
1005 00005.....	104
1005 00006.....	104
1003 00007.....	96
1005 00007.....	104
1003 00008.....	95
1005 00008.....	105
1003 00009.....	97
1001 00010.....	58
1003 00010.....	111
1001 00011.....	58
1003 00011.....	354
1001 00012.....	59
1003 00012.....	354
1003 00013.....	355
1003 00020.....	94
1003 00030.....	375
1005 00030.....	376
1003 00031.....	377
1003 00032.....	377
1003 00033.....	378
1003 00050.....	109
1003 00055.....	110
1003 00056.....	110
1003 00057.....	111
1003 00060.....	343
1003 00061.....	344
1003 00062.....	344
1003 00063.....	346
1003 00064.....	347
1003 00065.....	348
1003 00066.....	345
1003 00067.....	346
1003 00070.....	112
1003 00080.....	348

1003 00100.....	94
1010 00001.....	97
1010 00002.....	102
1010 00003.....	98
1010 00012.....	99
1010 00013.....	100
1010 00020.....	99
1010 00021.....	102
1010 00022.....	100
1010 00030.....	376
1015 00001.....	101
1015 00002.....	101
1015 00010.....	335, 336
1015 00100.....	298
1015 00110.....	168
1020 00001.....	91
1020 00002.....	92
1020 00003.....	91
1020 00004.....	92
1030 00110.....	403
1030 00120.....	404
1030 00125.....	406, 407
1030 00130.....	405
1030 00140.....	404
1030 00150.....	408, 409, 410, 411
1030 00160.....	407
1040 00001.....	120
1040 00002.....	154
1040 00008.....	121
1040 00009 [1].....	127
1040 00009 [2].....	127
1040 00009 [3].....	128
1040 00010.....	120
1040 00011.....	122
1040 00012.....	122
1040 00015.....	133
1040 00020.....	130
1040 00021.....	131
1040 00022.....	131
1040 00031.....	123
1040 00041.....	129
1040 00042.....	128
1040 00051.....	124
1040 00052.....	124
1040 00053.....	125
1040 00055.....	125
1040 00060.....	135
1040 00070.....	135
1040 00081.....	132
1040 00082.....	133
1040 00101.....	136
1040 00102.....	137
1040 00103.....	138
1040 00104.....	138
1040 00105.....	139

Index

1040 00106.....	140	3010 00020.....	214, 216
1040 00107.....	140	3010 00030.....	203
1040 00108.....	141	3010 00040.....	202, 203, 205
1040 00109.....	142	3010 00060.....	207, 208
1040 00110.....	142	3010 00070.....	207, 209
1040 00111.....	143	3010 00080.....	206, 210, 211, 212, 213
1040 00112.....	144	3010 00090.....	206, 208, 210, 211, 212
1040 00201.....	145	3010 01000 [1] .. 3010 01063 [1].....	216
1040 00202.....	145	3010 01000 [2..16] .. 3010 01063 [2..16].....	217
1040 00203.....	146	3020 00001.....	190
1040 00204.....	147	3020 00002.....	190
1040 00205.....	147	3080 00001.....	292
1040 00206.....	148	3080 00002.....	289
1040 00207.....	149	3080 00004.....	293
1040 00208.....	149	3080 00010.....	294
1040 00209.....	150	3080 00300.....	321
1040 00210.....	151	3080 00301.....	322
1040 00211.....	151	3090 00001.....	263
1040 00212.....	152	3090 00002.....	266
1040 00300[1].....	157, 160	3090 00003.....	267
1040 00300[2].....	158, 160	3090 00004.....	268
1040 00300[3].....	158, 161	3090 00005.....	268, 270
1040 00300[4].....	159, 162	3090 00006.....	271
1040 00401.....	118	3090 00007.....	273
1040 00999.....	152	4075 00100.....	183
1050 00001.....	63	4075 00101.....	184
1050 00002.....	57	4075 00102.....	185
1050 00004.....	63	4075 00104.....	187
1050 00020.....	169	4075 00105.....	183
1050 00021.....	166	4075 00106.....	184
1050 00022.....	167	4075 00108.....	186
1050 00050.....	167	4075 00200.....	182
2010 00100.....	330	4075 00201.....	187
2010 00110.....	330	4075 00202.....	188
2010 00120.....	332	4075 00203.....	189
2010 00130.....	331	4090 00001.....	316
2060 00003.....	227	4090 00100.....	316
2060 00009.....	224	6001 00001.....	51
2060 00010.....	224	6001 00010.....	240
2060 00012.....	225	6005 00010.....	240
2060 00013.....	226	6005 00020.....	249, 250
2060 00014.....	226	6005 00021.....	251
2060 00020.....	227, 228	6005 00022.....	248, 249
2060 00090.....	197	6005 00023.....	248
2060 00200.....	179	6005 00030.....	253, 255, 256
2060 00209.....	180	6005 00040.....	256
2060 00210.....	180	6005 00050.....	257
3010 00010 [1].....	219	6005 00061.....	257
3010 00010 [2].....	219	6005 00071.....	258
3010 00010 [3].....	221	6005 00100.....	242, 243, 244
3010 00010 [5].....	220	6020 00001.....	254
3010 00010 [7].....	222	6020 00002.....	254
3010 00010 [8].....	222	6020 00005.....	254
3010 00010 [9].....	223	6020 00011.....	252
3010 00010 [10].....	220	6020 00012.....	253
3010 00010 [11].....	221	6020 00021.....	258, 259
3010 00010 [12].....	221	6020 00022.....	259, 260
3010 00015.....	218	6020 00030.....	239

7010 00010.....	80	7050 00730.....	367
7010 00020.....	82	7050 00740.....	366
7010 00030.....	81	7050 00741.....	367
7010 00110.....	86	7050 00750.....	368
7010 00120.....	87	7050 00752.....	369
7010 00210.....	276	7050 00810 [1].....	415
7020 00010.....	86	7050 00815 [1].....	414
7020 00020.....	153	7050 00815 [2].....	413
7020 00030.....	163	7050 00820.....	415
7020 00040.....	163	7050 00900.....	369
7030 00010.....	71, 114	7050 00910.....	370
7030 00012.....	70, 113	7050 00912.....	370, 372
7030 00014.....	73	7050 00914.....	371
7030 00110.....	70	7050 00920.....	372
7030 00120.....	69	7050 00921.....	373
7030 00210.....	75	7050 00926.....	374
7030 00220.....	76	7050 00940.....	373
7030 00310.....	77	7050 00941.....	374
7030 00410.....	77	7050 01000 [1].....	325
7030 00420.....	78	7050 01000 [2].....	326
7030 00430.....	79	7050 01000 [3].....	326
7030 00500.....	296	7050 01000 [4].....	327
7040 00010.....	308	7050 01000 [5].....	327
7040 00020.....	308	7050 01000 [6].....	327
7040 00110 [1].....	311	7050 01010.....	328
7040 00110 [2].....	312	7050 01020.....	323
7050 00020.....	307	7050 01030.....	324
7050 00030.....	307	7050 01100.....	324
7050 00110.....	274	7050 01110.....	325
7050 00120.....	275	7050 01300.....	397
7050 00130.....	275	7050 01310.....	398, 399, 400
7050 00210.....	383	7060 00010.....	65
7050 00220.....	384, 385	7060 00020.....	66
7050 00230.....	386	7060 00030.....	67
7050 00240.....	387	7060 00110.....	280
7050 00250.....	387	7060 00120 [1].....	281
7050 00260.....	388	7060 00120 [2].....	282
7050 00320.....	73	7060 00120 [3].....	282
7050 00330.....	75	7060 00120 [4].....	284
7050 00340.....	74	7060 00120 [5].....	283
7050 00500.....	297	7060 00130 [1].....	286
7050 00510.....	68	7060 00130 [2].....	286
7050 00600 [1].....	389	7060 00130 [4].....	288
7050 00600 [3].....	390	7060 00130 [5].....	287
7050 00610.....	390	7060 00210.....	295
7050 00615.....	391	7060 00310.....	320
7050 00620.....	392	7060 00320.....	320
7050 00640.....	392	7060 00330.....	321
7050 00645.....	393, 394	7060 00410.....	69
7050 00650.....	393, 395	7060 00510.....	289
7050 00700.....	361	7060 00610.....	290
7050 00702.....	362	7070 00010 [1].....	290
7050 00708.....	363	7070 00010 [2].....	290
7050 00710.....	364	7070 00020.....	181
7050 00712.....	364	7070 00030.....	295
7050 00714.....	365	7080 00010.....	85
7050 00720.....	365	7080 00020.....	84
7050 00721.....	366	8005 00001.....	273

Index

8003 00001.....	312
8008 00001.....	336
8009 00001.....	277
8002 00001.....	334
8006 00001.....	351
8005 00002.....	274
8008 00002.....	337
8002 00002.....	334
8006 00002.....	352
8008 00003.....	337
8008 00004.....	338
8008 00005.....	339
8008 00006.....	339
8008 00007.....	340
8008 00008.....	340
8007 00010.....	379
8001 00010.....	357
8002 00011.....	334
8002 00012.....	334
8007 00020.....	380
8001 00020.....	359
8002 00021.....	335
8001 00021.....	359
8002 00022.....	335
8007 00030.....	382
8001 00030.....	358
8002 00031.....	333
8002 00032.....	333
8007 00099.....	380, 381
8010 00001.....	276
8011 00001.....	313
9020 00010.....	314
9020 00020.....	315
9030 00001.....	165
9030 00002.....	242
9040 00001.....	48
9040 00010.....	296
9050 00001 [1].....	299
9050 00001 [2].....	300
9050 00001 [3].....	301
9050 00001 [4].....	301
9050 00001 [5].....	302
9050 00001 [6].....	303
9050 00001 [7].....	305
9050 00001 [8].....	306
9050 00001 [9].....	303
9050 00001 [10].....	304
9050 00010.....	315
9060 00001.....	322

A

Abort.....	458
Abort position.....	462
Abort state.....	458
AbortAction.....	445
AbortAsupChan.....	444
AbortAsupNo.....	444

AbortDelay.....	443
AbortEnable.....	442
AbortPos.....	462
AbortRetractPos.....	445
About this documentation.....	1
Information representation.....	6
Validity of the documentation.....	1
AbsWin.....	130
AccAxJog.....	102
AccCouple.....	437
Acceleration (Acc).....	391
Acceleration axis load diagnostics	
Dwell time for temporary acceleration	
peak values.....	546
Acceleration in position controller range[1] in	
GearStep[1..4].....	123
Acceleration in position controller range[2] in	
GearStep[1..4].....	124
Acceleration/brake data (Acc).....	75
Access rights.....	15
AccLoadDgn/DgnMode.....	545
AccLoadDgn/PeakTmpHoldTime.....	546
AccThrd.....	391
Accuracy specification for circular program-	
ming (CircleAccur).....	306
AcknBcd32.....	207
AcknDouble.....	212
AcknFloat.....	211
AcknInt.....	210
Acknowledgement behavior for bit-coded	
auxiliary functions (BitAckn).....	213
Acknowledgement requirement	
Bcd32-coded auxiliary functions.....	207
Double-coded auxiliary functions.....	212
Float-coded auxiliary functions.....	211
Integer-coded auxiliary functions.....	210
Acknowledgement requirement from bit	
0000 to bit 0127 (Byte000_015).....	213
Acknowledgement requirement from bit	
0128 to bit 1535 (Byte016 to Byte191).....	215
AcsPosDisp.....	243
AcsTargPos[1..8].....	510
Act/ChAxNames[1..8].....	508
ActAxConf[1..8].....	511
ActBlkSwitchWrng.....	541
ActCh.....	240
ActEd.....	601, 603
ActFeed.....	538
Action block.....	582
Activating axis-specific override.....	94
Activating contour handwheel.....	528
Activating fast retract.....	392
Activating rapid traverse override.....	72
Activating the path velocity.....	537
Active.....	513, 532, 534, 535, 537, 539, 587
Active adjustment state.....	519
Active and preselected tool.....	602
Active and preselected tool at block pre-run.....	618

Active channel axis names in adjustment.....	508	AddrChIndepBcd32.....	226
Active channel in the display after switch-on....	240	AddrChIndepDouble.....	228
Active firmware release.....	596	AddrChIndepFloat.....	227
Active inclined plane, additional zero point offset.....	591	AddrChIndepInt.....	227
Active inclined plane, Euler angle.....	590	Address of the BCD32 auxiliary functions Address of channel-independent auxiliary functions.....	226
Active inclined plane, zero point.....	590	Address of the BCD32-coded auxiliary functions Addresses in channel interface Chlf[1..60] of channel-dependent auxi- ary functions.....	224
Active input revolutions of the master axis of the system axis coupling.....	465	Address of the bit-coded auxiliary functions Addresses in channel interface Chlf[1..60] of channel-dependent auxi- ary functions.....	223
Active KNB version string.....	597	Address of the double-coded auxiliary functions Address of channel-independent auxiliary functions.....	228
Active KNS version string.....	596	Addresses in channel interface Chlf[1..60] of channel-dependent auxi- ary functions.....	226
Active master axis offset in the system axis coupling.....	466	Address of the float-coded auxiliary functions Address of channel-independent auxiliary functions.....	227
Active offset of slave axis to master axis of the system axis coupling.....	464	Addresses in channel interface Chlf[1..60] of channel-dependent auxi- ary functions.....	225
Active output revolutions of the slave axis of the system axis coupling.....	465	Address of the integer-coded auxiliary functions Addresses channel-independent auxiliary functions.....	227
Active system axis index of the master axis of the system axis coupling.....	464	Addresses in channel interface Chlf[1..60] of channel-dependent auxi- ary functions.....	225
Active table of the system axis coupling.....	466	Addresses in the channel interface[1..60] of channel-dependent auxiliary functions.....	223
ActLowLim.....	339	Addresses of auxiliary functions in the PLC interface (Plclf).....	223
ActMode.....	513	Addresses of channel-independent auxiliary functions.....	226
ActOffset.....	463	Adjust G-function group G96/97.....	484
ActPos.....	462	Adjustment required.....	488
ActStatus.....	519	Adjustment status C-axis.....	491
ActTool.BQ1.....	607, 623	Coupling data.....	493
ActTool.BQ2.....	608, 623	Coupling state.....	493
ActTool.BQ3.....	608, 624	Drive operation mode.....	492
ActTool.IKQ1.....	605, 620	G96 state.....	491
ActTool.IKQ2.....	605, 621	Gear stage.....	489
ActTool.IKQ3.....	606, 621	Movement function.....	489
ActTool.IQ1.....	606, 622	Speed.....	490
ActTool.IQ2.....	607, 622	Speed limitation	490
ActTool.IQ3.....	607, 623	Adjustment status of the spindle coupling.....	504
ActTool.K1.....	603, 619	Adjustment status of the tool corrections.....	631
ActTool.K2.....	604, 619	Advanced control data of the function "Pro- grammable distance-dependent signal"	428
ActTool.SKQ.....	604, 620	AnalOAllocOut[d].....	186
ActToolValid.....	603	AnalOStartAddrOut.....	186
Actual data of the system axes.....	434	Analog inputs/outputs (AnalO).....	185
Actual data of the system spindle.....	569		
ActUpLim.....	338		
ActVel.....	463		
Adapt display areas.....	19		
Kernel display area.....	19		
Separate windows.....	20		
Switch off.....	19		
Switch on.....	19		
Adding/removing channel and drive objects.....	28		
Additional axes: System axis number of the master.....	348		
Additional data of the function programma- ble distance-dependent signal.....	429		
Additional information display area.....	13		
AddrChDepBcd32.....	224		
AddrChDepBit.....	223		
AddrChDepDouble.....	226		
AddrChDepFloat.....	225		
AddrChDepInt.....	225		

Index

AnaOutLowVelLim.....	414	AX/AY/AZ.....	591
AnaOutUpVelLim.....	414	AxAccReserve.....	98
Angle.....	529	AxConfNotAdjusted.....	518
Angle settings for tool guidance.....	384	Axes (AX) - Basic parameters -.....	89
Angle up to which the block transitions are seen as continuous.....	275	AxFeedFwd.....	96
Angular offset in case of active coupling.....	503	AxG96.....	86
ANSI Z535.6-2006.....	6	AxIndex.....	447, 464
Apply system axis coupling.....	453	Axis Acceleration Data (Acc).....	97
Approach.....	555	Axis acceleration reserve.....	98
Approaching strategy after block pre-run.....	584	Axis classification of system axes/coordi- nates (FwdInCoordClass).....	405
Area control (AreaCtrl).....	332	Axis configuration in the block preparation.....	511
Area monitoring Area[1..10].....	332	Axis configuration in the target block.....	512
AreaTypeAreaCtrl.....	332	Axis configuration is not adjusted.....	518
Assign analog inputs[1..4].....	186	Axis coordinate system (ACS).....	79, 243
Assign digital inputs[1..8].....	184	Axis coupling (AxCouple).....	343
Assign digital outputs [1..8].....	185	Axis coupling state.....	456, 457
Assigned spindle group.....	154	Axis description.....	80
Assigned system spindle.....	153	Axis group index.....	343
Assignment of system spindle to channel spindle/spindle group.....	116	Axis interface for increment input.....	374
Asup active.....	587	Axis jerk data (Jerk).....	99
AsupActive.....	557, 558, 587	Axis jerk reserve.....	100
AsynchrAxUnit.....	314	Axis jump.....	76
Asynchronous.....	58	Axis mode.....	389
Asynchronous axis.....	58, 60	Axis name.....	79
Asynchronous subroutine for system axis coupling errors.....	444	Axis names for position adjustment.....	509
ATConf.....	175	Axis not adjusted.....	469
Attribute values of a parameter.....	15	Axis number of the physical axis.....	333
AutoIPAddr.....	562	Retract axis.....	393
Automatic gear selection.....	140, 148	Secondary axis.....	394
Automatic gear shift.....	495	Axis position not adjusted.....	470
Automatic location correction.....	110	Axis position of the reference position.....	475
Automatic program reselection.....	68	Axis SHAPE (SHAPE).....	94
Automatic program selection after system restart.....	431	Axis transformation (TRA) - Optional param- eters.....	401
Automatic program selection after system restart: Fixing the part program.....	433	Axis transformation is not adjusted.....	518
Automatic program selection after system restart: Part program.....	431	Axis transformation number.....	473
Automatic program selection after system restart: Program selection flags.....	432	Axis transformation type.....	473
AutoProgSel.....	68	Axis transformation type (TrafoType).....	401
AuxCodeGr[1..32].....	257	Axis type.....	59
AuxFuncChDep.....	205	Axis velocity data (Vel).....	92
AuxFuncName.....	203	Axis-parallel measurement - Measurement results.....	551
Auxiliary axis.....	60	Axis-specific stroke release times.....	358
Auxiliary function basics.....	199	Axis/machine positions of reference position (RefPos).....	404
Auxiliary function class definition (ClassDef)....	201	AxisAssignment[1..8].....	474
Auxiliary function class definition (GrDef).....	216	AxisClassification[1..8].....	474
Auxiliary function classes.....	199	AxJerkReserve.....	100
Auxiliary function groups.....	201	AxModeThrd.....	389
Auxiliary function types.....	201	AxName.....	643
Auxiliary function[1..15].....	217	AxOvrStep[1-32].....	113
Auxiliary functions (AUXF) – Basic paramet- ers -.....	199	AxShape.....	94
AuxNotAdjusted.....	517	AxTrafoNo.....	473
		AxZeroPos[1..8].....	475

B

Backed up point of interruption available.....	584
Backed up target block available.....	520
Basic coordinate system (BCS).....	243
Basic data.....	362
Basic NC configuration (MAIN).....	47
Basic parameters.....	12
Basic setting.....	16
Battery warning.....	531
BattFail.....	531
BCD32 auxiliary function	
Channel-dependent auxiliary functions	
(ChDep).....	219
Channel-independent auxiliary functions	
(ChInDep).....	221
BCD32-coded auxiliary functions.....	207
BcsPosDisp.....	243
Bit assignment signal control word.....	177
Bit assignments signal status word.....	176
Bit auxiliary function.....	219
Bit-coded auxiliary functions.....	199, 205
BlkAngTangT.....	385
BlkCycleTimeCount.....	542
BlkCycleTimeWrng.....	540
Block cycle time warnings	
Counter for warnings.....	542
Block look-ahead for "Programmable distance-dependent signal".....	429
Block pre-run after target block in Look-Ahead range active.....	515, 587
Block pre-run data.....	506, 579
Block pre-run data for axes	469
Block switching control.....	154
Block transfer inhibit	
Standstill type at block end for block transfer inhibit.....	313
Block transfer inhibit (BlkTransferLock).....	313
Board FPGA version.....	599
BQ3_Basic.....	632
BSplineApproxTol.....	381
Buffer size for change event of the DB tables.....	315
Buffer size for external block selection.....	289
Buffer size to process external programs via NFS.....	288
BuffSizeExternBlk.....	289
BuffSizeNfs.....	288
Byte000 to Byte015.....	213
Byte016 to Byte191.....	215

C

C-axis type.....	496
CacheSize.....	469
Calc.....	526
CartCoord[1..6].....	83
CCOMP table	
Active table 1 and 2.....	434

CCOMP_FileName1.....	434
CCOMP_FileName2.....	434
Center point of the monitoring area (in MCS)...	334
Center point position of the secondary axis.....	530
Center point programming.....	306
CentPointAreaCtrl.....	334
Cfg.RetractF.....	592
Cfg.RetractX/RetractY/RetractZ.....	591
Cfg.TNR.....	593
CfgErrAx.....	454
CfgErrMaster.....	455
CfgErrNum.....	453
CHAN.....	65
Changing the drive functionality.....	29
Channel assignment.....	60
Channel assignment not adjusted.....	470
Channel axes/coordinates[1..8] that can be activated.....	276
Channel axis.....	362
Channel axis for G96.....	86
Channel axis names in the block preparation.....	507
Channel axis names in the target block.....	508
Channel axis/Logic axis.....	397
Channel dependency.....	205, 216
Channel display (ChDisp).....	239
Channel for asynchronous subroutine for system axis coupling errors.....	444
Channel initialization program	
File name.....	66
Channel is not adjusted.....	516
Channel reset.....	65
Channel SHAPE (SHAPE).....	73
Channel spindle	
Active data.....	478
Channel spindles.....	116
Channel waiting state	
Active gear waiting condition.....	481
Active movement waiting condition.....	480
Channel spindle number in waiting state.....	479
Spindle-caused channel waiting state active.....	478
System spindle number in waiting state.....	479
Channel-dependent and channel-independent auxiliary functions.....	199
Channel-dependent auxiliary functions	
(ChDep).....	219
Channel-independent auxiliary functions	
(ChInDep).....	221
Channel-specific data.....	323
Channel-specific memory.....	289
Channel-specific scaling factors	
for feed velocities and feed/revolution.....	307
For rotary axis velocities.....	309
Channels (CHAN) - Basic parameters.....	65
ChanNotAdjusted.....	516
ChAxClass.....	80
ChAxMcsCorr.....	362

Index

ChAxName.....	79	context menu GUI/TechDisp/Ch[k]/	
ChAxTCorrWcs.....	397	SpSpGrDisp[1..8].....	253
ChDepBcd32.....	219	context menu MAIN.....	50
ChDepBit.....	219	Context menu MAIN.....	50, 51
ChDepDouble.....	221	Context menu Spindle SP.....	115
ChDepFloat.....	220	Contour deviation tolerance for "PrecProg"	
ChDepInt.....	220	(PRP).....	312
Check Sercos timing at startup.....	651	Contour type.....	528
Checksum for Sercos configuration.....	641	ContourType.....	528
ChIfSigMcsCorr.....	363	Control data for handwheel mode in WCS.....	527
ChIndepBcd32.....	221	Control data of the function programmable	
ChIndepDouble.....	222	distance-dependent signal	421
ChIndepFloat.....	222	Control of the axis jump (Jump).....	76
ChIndepInt.....	222	Control type.....	597
ChIniProgFile.....	66	Conversion factor for spindle speed.....	572
ChOvrStep[1..32].....	70	Coord.....	533, 534, 536, 539
ChResetState.....	65	CoordClass.....	405
ChShapeG8.....	73	CoordDir.....	406
ChShapeG9.....	73	CoordG96.....	86
Circle center point position of the main axis.....	529	Coordinate identifiers in the MCS.....	83
Circular corner rounding.....	274	Coordinate name[1..6].....	83
ClassDef.....	201	Coordinate selection for path velocity calcu-	
Classification system axis/coordinate.....	405	lation (CoordSel).....	415
Clock change to winter time.....	303, 304	Coordinate system display of the tool edi-	
CmdOffsIpoMode.....	345	tor.....	524, 525, 526
CollDet.....	368	Coordinate system for G96 reference axis.....	86
Collision detection.....	368	Coordinate system of the tool editor.....	526
Columns in the display area "Edit paramet-		Coordinate system selection.....	242
ers" - Adapt.....	23	Coordinates (Coord).....	79
Command spindle speed, active.....	569	CoordNotAdjusted.....	518
Command value at the analog output (Out).....	414	CoordProgName.....	324
Command value when reaching the lower		CoordSel.....	415
velocity limit value.....	414	CoordTCorrTcs.....	399
Command value when reaching the upper		CoordTcsXBcs.....	407
velocity limit value.....	414	CoordTcsYBcs.....	409
Communication server (CommServ).....	316	CoordTcsZBcs.....	410
Communication settings for PLC.....	387	Copying parameter objects.....	24
Compatibility mode spindles.....	116	Corner rounding (CornRound).....	274
Compensate for command value gantry off-		Corner rounding with splines.....	274
set.....	345	Correction direction.....	362, 398, 399
Complaints.....	8	Couple link programmed in pre-run.....	504
Component operating time.....	587	Coupling (Couple).....	85
Computation active.....	513, 587	Coupling distance.....	501
Computation mode.....	513	Coupling rigidity of the system axis coupling	
Configuration of the Ethernet interface.....	594	after successful coupling.....	440
Configuration of the MAC address.....	595	Cp.CoupleOn.....	505
Configuration of the network mask.....	594, 595	Cp.IsProg.....	504
Configuration of velocity.....	596	Cp.Master.....	505
ConfNotAdjusted.....	470	Cp.NotAdjusted.....	504
Constant cutting velocity		Cp.Slave.....	506
Channel reference axis.....	497	CpIfEndBit.....	181
Coordinate system.....	498	CpIfStartBit.....	180
ConstCutStatus.....	484	CpLinkBuff.....	290
ContDevTol.....	312	CpIStack.....	289
Context menu.....	152	CpIVarMcsCorr.....	365
context menu AUXF/ClassDef/Class[1..30]/.....	202	CpIVarWcsCorr.....	371
Context menu AUXF/ClassDef/Class[1..30]/.....	228	CPU time for block processing.....	295
		CpuTemp.....	588

CpuTimeBlkPrep.....	295	Dead area.....	332
Criticism.....	8	Deceleration (Dec).....	391
Current.....	645	Decimal places.....	208
Current actual position value difference.....	461	Decimal position for linear axes in case of metric display of tables.....	258
Current axis configuration in adjustment.....	511	Decimal position for rotary axes in case of metric display of tables.....	259
Current axis position.....	462	Decimal position for rotary axes in case of metric displaying of tables.....	259
Current axis velocity.....	463	DeciPlaceBcd32.....	208
Current master value offset.....	463	DecThrd.....	391
Current memory of internal permanent sys- tem data.....	567	Default measuring unit for asynchronous lin- ear axes.....	314
Current memory of internal volatile system data.....	568	Default value.....	14
Current memory of permanent customer system data.....	566	DefaultCh.....	60
Current memory of temporary customer sys- tem data.....	567	Defined step width.....	327
Current memory of temporary volatile sys- tem data.....	568	Defining a new auxiliary function.....	228
Current memory of volatile customer system data.....	567	Definition of possible synchronization devia- tions of system axis coupling.....	441
Current storage index of the program stack files (call chain files).....	580	Definition of tool axes.....	383
Current system axis coupling data.....	456	DefStepAxVel.....	104
Customer Feedback.....	8	DefStepVelWcs.....	327
CustPermMax.....	565	Degree of the connection spline.....	380
CustPermUsed.....	566	Delayed reaction in case of non-fatal drive errors.....	58
CustTmpUsed.....	567	DelayedErrorReaction.....	58
CustVolMax.....	566	Delete SetPos Offset with a control reset.....	68
CustVolUsed.....	567	DelZpo.....	68
Cut mode (ModeSel).....	389	Detail data.....	14, 16
CutCompActDeactMode.....	277	Device.....	363, 369
Cutting velocity.....	499	Device = CplVar (permanent CPL variable) Online correction in MCS (distance con- trol).....	365
Cycle G230, default for TNR.....	593	Online correction in WCS.....	371
Cycle G230, retract feed.....	592	Device = Dr (drive).....	364, 370
Cycle G230, retract position.....	591	Device = handw (handwheel).....	371
Cycle time.....	197	DiaCoord[1..8].....	276
D		Diameter programming (DiaProg).....	275
D/A converter[1..4] type.....	188	DigAnaConvGain[1..4].....	187
Data for cycle Swiveling G230.....	589	DigAnaConvOffs[1..4].....	187
Data groups.....	11	DigAnaConvType[1..4].....	188
Data of active axis transformation.....	472	DigIOAllocIn[d].....	184
Data of the memory space assignment of system data.....	565	DigIOAllocOut[d].....	185
Data of the zero point offsets (axis-related)	643	DigIOHw.....	182
Data of the zero point offsets (channel-rela- ted).....	644	DigIOStartAddrIn.....	183
Data of the zero point offsets (global).....	641	DigIOStartAddrOut.....	183
Database tables (DbTables).....	315	Digital inputs/outputs (DigIO).....	182
Date		Digitized data.....	351
Date/time setting (DateTime).....	303	Digitizing (DIG).....	40
Date/time setting to summer time.....	304	Digitizing (DIG) - Optional parameters.....	351
Date/time setting (DateTime).....	298	Digitizing buffers, number.....	351
Clock change to winter time.....	303	Dimension Dim[1..2].....	333
Summer.....	301	Dimensional tolerance for B-spline approxi- mation (default setting).....	381
Winter.....	299	DirCollMon.....	338
DbTabEventBuffSize.....	315	Direction of TCS_X in BCS.....	407
		Direction of TCS_Y in BCS.....	409
		Direction of TCS_Z in BCS.....	410

Index

Direction of the abscissa.....	525	DrIndAreaCtrl.....	333
Direction of the ordinate.....	525	DrIndIncrSource.....	374
Directory for re-entry data.....	321	DrIndMcsCorr.....	364
Directory structure of the MTX.....	291	DrIndMonAx1.....	337
DirMcsCorr.....	362	DrIndMonAx2.....	337
DirTangT.....	384	DrIndRetrAx.....	393
DirTCorrTcs.....	399	DrIndSeconAx.....	394
DirTCorrWcs.....	398	DrIndSpEnco.....	132
DirTcsXBcs.....	408	DrIndWayPoint.....	329
DirTcsYBcs.....	409	DrIndWcsCorr.....	370
DirTcsZBcs.....	411	Drive Dr[i] without specified command value.....	58
Disabling context filtering.....	25	Drive function type.....	53
Disabling optional filtering.....	26	Drive index.....	49
Display area		Drive number.....	364, 487
Data group selection.....	14	Online correction in WCS.....	370
Edit parameters.....	14	Drive number of the external spindle encod-	
Information.....	14	er.....	132
Parameter instances.....	16	Drive parameter sets per gear step	
Search results.....	15	(IndraDrive only).....	126
Status bar.....	17	Drive parameterization.....	54
Display of zero point offsets (ZotDisp).....	257	Drive system functions (AxFunc).....	59
Display order.....	244	Drive telegram.....	175
Display order of axes and channel coordi-		Drive telegram entry.....	175
nate packages.....	244	Drive type.....	63
Display precision.....	251	Drive without command value input.....	58
Display section "Edit Parameters" - Navigate.....	21	Drive-dependent move to positive stop.....	375
Display sections - NC Configurator.....	12	Drive-side parameter sets (ScsParSet).....	125
Displaying axis in the additional window.....	249	Drive-specific Safety parameters.....	341
Displaying axis in the channel window.....	248	Drives.....	48, 49
Displaying channel coordination in the addi-		DrType.....	63
tional window.....	250	Dynamic limitations.....	173
DispUnit.....	239		
DispUsrPara.....	174	E	
Distance control.....	361	Edge radius correction R.....	615
Distance control to programmed end point		Edge tool position Ori.....	616
(AxDistCtrl).....	335	Edge touch probe 1.....	354
distance to the programmed endpoint.....	335	Edge touch probe 2.....	354
Distance to the programmed Reset mark		Edge_variable.....	632
ADSLAB.....	424	Edit parameters.....	23
Distance to the programmed Set mark AD-		Electronic axis coupling (COUP).....	40, 343
SLAB.....	423	EnablAutoCorr.....	110
DistEndPoint.....	335	EnablAuxFunc.....	202
DistWcsJog.....	323	EnablAxAsynchrOvrd.....	94
Divisor		EnablAxDistCtrl.....	335
Online correction in MCS (distance con-		EnablBkSwiStrategy.....	163
trol).....	366	EnablChCtrl.....	135
Online correction in WCS.....	372	EnablChgPosLog.....	108
DivMcsCorr.....	366	EnablCollMon.....	336
DivWcsCorr.....	372	EnablCommServ.....	316
Documentation		EnablCoup.....	40
Change record.....	1	EnablDig.....	40
Double auxiliary function		EnablDr.....	56
Channel-dependent auxiliary functions		Enable.....	447, 522
(ChDep).....	221	Enable drive.....	56
Channel-independent auxiliary functions		Enable link process optimization (same proj-	
(ChInDep).....	222	ect names).....	294
Double-coded auxiliary functions.....	200, 212	Enable master axis for system axis coupling....	447
Downloading Sercos data in phase 3.....	166		

Enable nibbling.....	357	Encoder selection.....	172
Enable punching.....	357	End bit address.....	207
Enable Sercos communication.....	57	End code.....	206
Enable test mode.....	168	End of range.....	181
Enable the distance control.....	335	EndBitAddr.....	207
Enable the one-dimensional collision monitoring.....	336	Endless axis.....	105
Enable virtual mode.....	57	EpsAxTrafo.....	407
Enable web server.....	316	Epsilon environment.....	407
Enabled.....	641	Epsilon environments.....	477
EnablEqProgNamesLinkOpti.....	294	EpsilonRanges.....	477
Enabling automatic location correction.....	110	Error.....	460
Enabling auxiliary function.....	202	Error code.....	460
Enabling block switching strategy.....	163	Error reactions of the system axis coupling.....	445
Enabling communication server.....	316	Error when applying system axis coupling data.....	453
Enabling context filtering.....	25	Eth(1,2)/GatewayAddr].....	595
Enabling intelligent Safety Engineering.....	341	Eth(1,2)/IpAddr].....	594
Enabling optional filtering.....	25	Eth(1,2)/MACAddr].....	595
Enabling the axis coordinate storage.....	532	Eth(1,2)/SubNetMask].....	594
Enabling the basic workpiece positions.....	534	Ethx/ISpeed].....	596
Enabling the machine coordinate storage.....	535	Euler angle Phi, Theta, Psi.....	617
Enabling the MCS online correction.....	361	ExpandAreaCtrl.....	334
Enabling the modulo calculation.....	105	Exporting/importing a data block.....	30
Enabling the online correction in MCS.....	361	Exporting/importing configuration data.....	29
Enabling the online correction in WCS.....	369	Exporting/importing data group or object node.....	29
Enabling the software limit switch.....	90	ExtCond.....	455
Enabling the test mode.....	522	Extension of the monitoring area (side length).....	334
Enabling the WCS online correction.....	369	External real-time switching conditions for master axis switching of the system axis coupling.....	455
Enabling the workpiece coordinate storage.....	539	External spindle encoder.....	132
EnablMcsCorr.....	361	EXTERNAL.SP_EMPTY.....	640
EnablMeas.....	41	EXTERNAL.TOOL_FOL.....	639
EnablModCalc.....	105	ExtError.....	442
EnablMovePoStop.....	375	F	
EnablMps.....	42	FactMcsCorr.....	365
EnablOem.....	46	FactRotInch.....	310
EnablOnl.....	42	FactRotMetr.....	311
EnablProbeFunc.....	353	FactVel.....	307
EnablPun.....	41	FactVelPerRev.....	308
EnablPunch.....	357	FactWcsCorr.....	372
EnablRapTravOvrd.....	72	Fan operating time.....	588
EnablRefSwiMon.....	347	FanTime.....	588
EnablRetrThrd.....	392	Fast.....	326
EnablSafe.....	39	Fast retract (Retr).....	392
EnablSafeTech.....	341	FastAxVel.....	103
EnablScsCom.....	57	FastVelWcs.....	326
EnablSpl.....	43	Feed.....	69
EnablSynchrMon.....	346	Feed calculation.....	93
EnablTestMode.....	168	Feed display.....	253
EnablThrd.....	43	Feed Forward (FeedFwd).....	96
EnablTlwp.....	44	Feed forward can be activated (via FFW/FFWA).....	96
EnablTra.....	44	Feed override.....	70
EnablTrvLim.....	90		
EnablTtl.....	43		
EnablVdo.....	45		
EnablVirtMode.....	57		
EnablWcsCorr.....	369		
EnablWebServ.....	316		
Enco.....	172		

Index

Feed override (Ovrd).....	70	Gantry group.....	343
Feed override for step[1..32].....	70	Gear.....	494
Feed stop after returning to contour.....	556	Gear change method.....	120
FeedAxAcc.....	97	Gear idle selection (disengaged).....	143, 151
FeedAxJerk.....	99	Gear step[1..4].....	121
Feedback.....	8	Gear syntax of a spindle.....	486
FeedCalc.....	93	GearChgMethod.....	120
FeedHold.....	556	GearStr.....	486
FeedMode.....	556	GearTeeth.....	438
Feedrate2Ovr.....	477	General parameters of auxiliary function	
FeedVal.....	553	classes.....	202
File organization		General Sercos settings.....	171
Channel-specific settings.....	294	GFunc[1..80]/GFuncCode.....	261
File organization (FileOrg).....	291	GFuncSubProg.....	263
FileName.....	467	Global output behavior of auxiliary functions	
FileTime.....	468	(OutBehav).....	217
Filtering the display in the section "Edit pa-		Graphical user interface (GUI) - Basic pa-	
rameters".....	25	rameters -.....	239
FilterMcsCorr.....	367	GrAuxFunc[1..15].....	217
Finding parameter objects.....	27	GrChDep.....	216
fixDefault.....	433	Grid positions.....	110
Float auxiliary function			
Channel-dependent auxiliary functions		H	
(ChDep).....	220	Handwheel functionality (Handw).....	322
Channel-independent auxiliary functions		Handwheel number.....	371
(ChInDep).....	222	HandwWcsCorr.....	371
Float-coded auxiliary functions.....	200, 211	Hard disk-coded magazine.....	636
Follow-on tool programmed.....	639	Hardware selection.....	182
Following error.....	343	Hardware version.....	598
Formula.....	446	Hazard warnings.....	6
Formula expression diagnostics in the sys-		Helpdesk.....	675
tem axis coupling.....	460	Hide channel within the interface.....	240
Formula for the master value calculation in		Hiding Safety Engineering in the NC.....	341
the system axis coupling.....	446	Hirth axes.....	109
Formula runtime error in the system axis		Hole detection.....	368
coupling.....	458	HoleDet.....	368
Formula runtime error position in the system		horizontal.....	525
axis coupling.....	459	Hotline.....	675
FormulaDiag.....	458	HSDigIOSynchrMode.....	182
FormulaDiagPos.....	459	HW/IBoardFPGA].....	599
FormulaWatch.....	460	HW/ILpNmb].....	598
FPGA version of the Sercos connection 1 or		HW/ISerNmb].....	598
2.....	600	HW/IType].....	597
Function assignment of analog outputs.....	186	HW/IVersion].....	598
Function assignment of digital inputs.....	184		
Function assignment of digital outputs.....	185	I	
Function group.....	11	Icons.....	17
Function mode for re-entry.....	321	ID (parameter ID).....	14
FW/IKnRel].....	597	IfSigWayPoint.....	331
FW/IKnsRel].....	596	In-Position evaluation (InPosEval).....	297
FW/ISysRel].....	596	Incorrect axis when applying system axis	
FwdInCoordIndTrafo.....	403	coupling data.....	454
		Incorrect master axis data when applying	
		system axis coupling data.....	455
		IncrAxJog.....	101
		Increment calculation	
G			
G-code number.....	261		
Gain of the D/A converter[1..4].....	187		
Gain specification.....	187		

Online correction in MCS (distance control).....	365
Online correction in WCS.....	372
Incremental specifications when programming the diameter.....	275
IncrPerRev.....	322
IncrSetPointsDiaProg.....	275
IncrWcsJog.....	323
IndBcd32.....	209
IndDouble.....	212
Index	
Bcd32-coded auxiliary functions.....	209
Double-coded auxiliary functions.....	212
Float-coded auxiliary functions.....	211
Integer-coded auxiliary functions.....	210
Index of the selected signal.....	427
IndFloat.....	211
IndInt.....	210
IndraLogic.....	179
Information on the system configuration change.....	478
Information representation	
Names and Abbreviations.....	8
Init adjustment program (NameOfInitAdjPrg) active.....	515
Init program.....	66
InitAdjPrgActive.....	515
Initial address of digital inputs.....	183
Initial address of digital outputs.....	183, 186
Initialization of the channel status.....	65
InPos.....	461
Inpos window, rough.....	297
InPosEvalMode.....	297
InPosRough.....	297
Input revolutions of the master axis of the system axis coupling.....	451
Input revolutions of the tool.....	577
Integer auxiliary function	
Channel-dependent auxiliary functions (ChDep).....	220
Channel-independent auxiliary functions (ChInDep).....	222
Integer places.....	208
Integer-coded auxiliary functions.....	200, 210
Intelligent safety technology (SAFE).....	39
Intended use	
Areas of application.....	9
Introduction.....	9
Use cases.....	9
Interaction between simulation and NC program.....	645
Interface signal.....	363
Interface signal of the point.....	331
Intermediate block angle (after which an intermediate block is inserted).....	385
Interpolation cycle time.....	165
IP address.....	560
IP addresses.....	560

IPAddMtx.....	561
IpAddress.....	560
IPAddSubDev.....	561
IpoEnable.....	436
IpoJlaWhileWrng.....	541

J

JerkAxJog.....	102
Jog acceleration.....	102
Jog jerk.....	102
Jog mode.....	102
Jog mode (increments and acceleration).....	101
Jog mode (Jog).....	100
Jog speed[1..8].....	133
Jog spindles.....	133
Jog velocity defined steps.....	104
Jog velocity fast.....	103
Jog velocity medium.....	103
Jog velocity rapid traverse.....	104
Jog velocity slow.....	102
Jog velocity variable steps.....	105
Jogging axes in workpiece coordinates (WcsJog).....	323
Joint length and angle parameters (JointPar).....	404
JointParTrafo[1..16].....	404

K

Keep velocity in motion-neutral NC blocks.....	296
KeepPathVelNoIpoAct.....	296
Kernel display area.....	13
KindOfLinking.....	295
Kinematic limitation (KinLim).....	77
Kinematic limitation filter (presetting).....	78
Kinematic limitation level (presetting).....	77
Kinematic limitation scaling (presetting).....	77
KinLimFilter.....	78
KinLimLevel.....	77
KinLimWeight.....	77

L

L1.....	615
L2.....	615
L3.....	615
Lag.....	461
Laser power control.....	413
LastOriPos.....	569
LastSelProg.....	431
Latest activated orientation position.....	569
Length and angle parameter of the axis transformation.....	476
Length correction L1, L2, L3.....	615
Length parameter/angle parameter.....	404
LenParam[1..16].....	476
Limit switch monitoring (Pos).....	89
Limit velocities (Vel).....	413
Limits	

Index

Online correction in MCS (distance control).....	366	Master axis offset of the system axis coupling.....	452
Online correction in WCS.....	373	Master data telegram.....	174
LinCoord[1..3].....	84	Master data telegram entry.....	174
Linear axis.....	59	Master spindle	
Linear coordinates in the WCS.....	84	Channel spindle number.....	505
Linear/rotary axis.....	53	Master value calculation mode in the system	
Link process optimization for NC programs with the same name.....	294	axis coupling.....	450
Linking type.....	295	MastInd.....	343
List of the devices connected.....	559	MastIndSuplAx.....	348
Location		Max. switching speed to position control mode 2.....	576
Summer.....	301	MaxAccMcsCorr.....	367
Winter.....	299	MaxAccPoStop.....	376
Location programming (LocProg).....	109	MaxAccWcsCorr.....	373
Lock target block: Note text.....	580	MaxAng.....	275
Lock target blocks.....	520, 587	MaxAxOvrd.....	112
LockTargBlock.....	520, 587	MaxBlkBuffered.....	281
LockWarning.....	580	MaxBlkMiscFunc.....	283
LogAxisNmb.....	559	MaxBlkSpline.....	282
Logging Sercos parameters transmitted at startup.....	650	MaxBlkTCorr3D.....	282
Logic axis number.....	559	MaxBlkVelCalc.....	280
Look-ahead.....	279	MaxChAcc.....	75
LookAheadPreRun.....	515, 587	MaxChDec.....	76
Lower limit value.....	340	MaxChOvrd.....	70
Lower speed limit value of the spindle.....	571	MaxChVel.....	69
Lower velocity limit value.....	413	MaxDev.....	274
LowVelLim.....	413	MaxDevPoStop.....	377
LSEC table		MaxEd.....	542, 601
Active table.....	434	MaxFollErr.....	343
LSEC_FileName.....	434	MaxGrTorq.....	344
M		Maximum (parameter value).....	14
M-code number		Maximum acceleration	
Non-modal M-functions MFunc[1..8].....	266	Online correction in MCS (distance control).....	367
M-function for group[1..32]		Online correction in WCS.....	373
Modal M-function group display (Aux-FuncDisp).....	257	Maximum acceleration in position control mode 1.....	575
Machine coordinate system (MCS).....	242	Maximum acceleration in position control mode 2.....	576
Machine coordinates (MCS).....	83	Maximum acceleration of GearStep[1..4].....	122
Machine parameter ID.....	15	Maximum acceleration of the tool.....	575
Machine state display MZA (MachStatDiag).....	197	Maximum acceleration when coupling and decoupling in case of system axis coupling.....	437
Machining axis.....	60	Maximum allowed deviation from the contour.....	274
MachStatDiagCycTime.....	197	Maximum axis acceleration.....	376
MAIN context menu.....	53	Maximum axis acceleration in feed mode.....	97
Main spindle.....	85	Maximum axis acceleration in rapid traverse.....	98
MAIN/Dr[i] context menu.....	54	Maximum axis jerk in feed mode.....	99
MainCP.....	529	Maximum axis jerk in rapid traverse.....	99
MainSp.....	85	Maximum axis velocity.....	92, 375
Manual data input after block pre-run.....	585	Maximum edge number.....	542, 601
Manual selection gear step 1.....	140, 149	Maximum feed override.....	70
Manual selection gear step 2.....	141, 149	Maximum following error.....	343
Manual selection gear step 3.....	142, 150	Maximum memory of permanent customer system data.....	565
Manual selection gear step 4.....	143, 151		
ManWaitForStart.....	585		
Master axis.....	344		

Maximum memory of volatile customer system data.....	566	McsPosDisp.....	242
Maximum number of axes in the axis display...	247	MDTConf.....	174
Maximum number of blocks		Meaning of channel axes with regard to machining function.....	80
for 3D milling cutter path correction.....	282	MeasSysSpTurrAx.....	134
for a jerk-limited velocity control.....	280	Measuring (MEAS).....	41
for block buffer.....	281	Measuring (MEAS) - Optional parameters.....	353
for other functions.....	283	Measuring unit for display (DispUnit).....	239
for spline functions.....	282	Medium.....	325
function-related blocks.....	280	MediumAxVel.....	103
Maximum number of channels.....	47	MedVelWcs.....	325
Maximum number of drives.....	48	Memory assignment of the system axis coupling table	469
Maximum number of jog movements to be recorded.....	320	Memory for CPL link tables (UP-tab, var.-tab., ...).....	290
Maximum number of NC kernel-controlled drives.....	542	Memory for CPL programs (subroutines, variable, ...).....	289
Maximum number of spindles.....	52	Memory for the IndraLogic user program.....	180
Maximum override.....	112	Memory management (Mem).....	288
Maximum path acceleration.....	75	Memory space assignment data.....	565
Maximum path deceleration.....	76	MFuncCode.....	266
Maximum path velocity.....	69	MFuncSubProg.....	267
Maximum permissible relative path velocity error.....	379	MinAngVelJump.....	76
Maximum permissible starting point/end point deviation.....	379	MinAxShape.....	95
Maximum speed of GearStep[1..4].....	121	MinChShape.....	74
Maximum spindle jerk in position control operation of GearStep[1..4].....	125	Minimum (parameter value).....	14
Maximum spindle override.....	128	Minimum angle for considering the axis jump.....	76
Maximum standstill torque.....	112	Minimum block length for corner rounding.....	274
Maximum standstill torque total of the group....	344	Minimum number of configured drives.....	51
Maximum velocity		Minimum SHAPE order for axis.....	95
Online correction in MCS (distance control).....	366	Minimum SHAPE order for path jerk limitation (G8, G9).....	74
Online correction in WCS.....	373	Minimum size of the PLC user program memory.....	180
Maximum velocity when coupling and decoupling in case of system axis coupling.....	438	Minimum speed of GearStep[1..4].....	122
MaxNofCh.....	47	MinLen.....	274
MaxNofDr.....	49	MinNofDrNodes.....	51
MaxNofSp.....	52	MinSpeedLimit.....	571
MaxOpenFile.....	294	MinSpSpeed.....	122
MaxPosDevSpline.....	379	Modal auxiliary functions are not adjusted.....	517
MaxRedCmdOffs.....	345	Modal M-function group display (AuxFuncDisp).....	257
MaxSpAcc.....	122	Modal NC function group display (NcCodeDisp).....	256
MaxSpAccPosCtrl[1].....	123	Modal NC functions are not adjusted.....	517
MaxSpAccPosCtrl[2].....	124	Modal subroutines SwiOn[1..63] / SwiOff.....	268
MaxSpeedLimit.....	571	Mode.....	450
MaxSpJerkPosCtrl.....	125	Mode of the acceleration axis load diagnostics	
MaxSpNumber.....	483	Mode.....	545
MaxSpOvrd.....	128	Mode of the In-Position evaluation.....	297
MaxSpSpeed.....	121	Mode of the velocity axis load diagnostics	
MaxStandstTorq.....	112	Mode.....	544
MaxTorqPoStop.....	376	ModeAddAx.....	249
MaxVel.....	92	ModeAxChWin.....	248
MaxVelDevSpline.....	379	ModeChPack.....	250
MaxVelMcsCorr.....	366	ModeLocProg.....	109
MaxVelPoStop.....	375	ModePosLog.....	106
MaxVelWcsCorr.....	373	ModeScsFiles.....	166

Index

Modulo value.....	105	Enabling optional filtering.....	25
Monitor lower limit.....	339	Exporting/importing a data block.....	30
Monitor upper limit.....	338	Exporting/importing configuration data.....	29
Monitoring.....	368	Exporting/importing data group or object node.....	29
Monitoring axis 1.....	337	Filtering the display in the section "Edit parameters".....	25
Monitoring axis 2.....	337	Finding parameter objects.....	27
Monitoring for rotary axes.....	331	Functions.....	11
Monitoring type.....	330	How to - configure the NC.....	31
MonRotWayPoint.....	331	Navigate in the display area "Edit Param- eters".....	21
MonTypeWayPoint.....	330	Select data groups.....	21
Motion direction.....	406	Selecting display areas.....	20
Motion function.....	494	Showing parameter attributes.....	24
Motion syntax of a spindle.....	485	Start.....	18
Move to positive stop (MPS).....	42	Undoing changes to parameters.....	24
MoveStr.....	485	NC diagnostics, control data.....	543
MpxCh.....	172	NC function for group[1..32] Modal NC function group display (NcCo- deDisp).....	256
Multiplex channel.....	172	NC function for the bit interface iCh_Act- Func[1..24].....	189
Multiplication		NC function-specific bit interface (NcFuncBi- tlf).....	189
Online correction in WCS.....	372	NC kernel diagnostics.....	540
Multiplier		NC optimization (NCO) - Basic parameters.....	279
Online correction in MCS (distance con- trol).....	365	NC programming (NCP) - Basic parameters....	261
N		NC-controlled drive.....	55
N_Input.....	451, 465	NC-PLC communication before executing an intermediate block.....	387
N_Output.....	451, 465	NC/PLC-Interface (PLC) -Basic parameters-....	179
Name.....	203	NcCodeGr[1..32].....	256
Name (parameter name).....	14	NcfNotAdjusted.....	517
Name of Init adjustment program.....	582	NcFunc[1..24].....	189
Name of target block adjustment program (Asup).....	581	NcFuncBitlf.....	189
Name of the linear coordinate[1..3].....	84	NcPlcComTangT.....	387
Name of the rotary coordinate[1..3].....	85	NcResetState.....	65
Name of time zone		Negative travel limit.....	92
Summer.....	301	NeTrvLim.....	92
Winter.....	299	NofAxPosDisp.....	247
NameOfAsup.....	555, 557, 581	NofBlkPrep.....	280
NameOfInitAdjPrg.....	582	NofBlkRetrace.....	290
NC configurator.....	11	NofCh.....	48
Troubleshooting MTX advanced: NC con- figurator reports error 31 after change in parameterization.....	32	NofDigBuff.....	351
NC Configurator		NofDr.....	50
Adapt columns in the display area "Edit parameters".....	23	NofDrNodes.....	52
Adapt display areas.....	19	NofEdgeTangT.....	383
Adding/removing channel and drive ob- jects.....	28	NofGears.....	120
Changing the drive functionality.....	29	NofJogSteps.....	320
Close.....	18	NofLoc.....	110
Configuration data structure.....	11	NofOverlapBlk.....	381
Copying parameter objects.....	24	NofPmtSets.....	258
Disabling context filtering.....	25	NofReservBuff.....	351
Disabling optional filtering.....	26	NofScsParSets.....	126
Display sections.....	12	NofZotSets.....	257
Edit parameters.....	23	Non-bit-coded auxiliary functions.....	200
Enabling context filtering.....	25	Non-modal G-functions GFunc[1..80].....	261

NotAdjusted.....	469
Note feed potentiometer for TimeOn and TimeOff.....	428
Number of auxiliary functions.....	218
Number of blocks for retrace mode.....	290
Number of channels.....	48
Number of configured drives.....	52
Number of decimal places for metric Display.....	251, 253
Number of decimal places for non-metric Display.....	252, 254
Number of decimal position for spindle speed.....	254
Number of drives.....	49
Number of edges.....	383
Number of edges displayed.....	542
Number of gear steps.....	120
Number of increments/revolution.....	322
Number of locations.....	110
Number of NC-controlled drives.....	50
Number of open files per channels.....	294
Number of parameters.....	272
Number of placement banks to be displayed....	258
Number of placements (PmtDisp).....	258
Number of spindle tools.....	634
Number of the active D-correction in the target block.....	630
Number of the active edge.....	603
Number of the active edge in the target block.....	630
Number of the currently active zero point offset table in the channel.....	645
Number of the spindle group to be displayed...	255
Number of the spindle to be displayed.....	255
Number of tool edges.....	383
Number of usable auxiliary functions (No-fAuxFunc).....	218
Number of zero offset banks to be displayed....	257

O

Object icons.....	17
Object node, default.....	17
Object node, locked.....	17
Object node, special.....	17
OEM data group (OEM).....	46
OEM-specific parameters.....	12, 45
OEM-specific parameters: Prerequisites.....	45
OffAngTangT.....	386
Offset.....	450, 452, 464, 466, 644
Offset angle between the symmetry axis of the tool and the path tangent	386
Offset of slave axis to master axis of the system axis coupling.....	450
Offset of the D/A converter[1..4].....	187
Offset specification.....	187
One-dimensional collision monitoring (Coll-Mon).....	336

ONL context menu.....	363
ONL/WcsOnlCorr/Ch[k]/Device context menu.....	370
Online correction in MCS (distance control)....	361
Online correction in WCS.....	369
Online Corrections (ONL).....	42
Online corrections (ONL) - Optional parameters -.....	361
OnOff.....	435
Operating functions (OPF) - Basic parameters.....	319
OptChAxName[1..8].....	82
OptFeedVal.....	553
OpTime.....	587
Optional axis name[1..8] (OptAxName).....	82
Optional parameter setup (SUP) - selection.....	39
Optional parameters.....	12, 39
Order of the axis in the additional window.....	250
Order of the axis in the channel window.....	248
Order of the channel coordinate package in the additional window.....	251
Ori.....	616
OriCoord[k].....	85
Orientation position.....	499
Orientation position for part program.....	573
Orientation position for PLC specifications.....	573
OriPos.....	573
OutBehavMode.....	217
Output revolutions of the system axis coupling.....	451
Output revolutions of the tool.....	578
Output type.....	217
OveForTime.....	428
Overlapping of two partial sequences.....	381
Override for asynchronous axes (AsynchrAxOvrldf).....	112
Override for asynchronous axes (Ovrd).....	94
Override value for step [1..32].....	113
Override value for step[1..32].....	128
OverTemp.....	530
OX/OY/OZ.....	590

P

Parameter attributes.....	15
Parameter content and properties.....	33
Parameter names.....	33
Parameter set for C-axis operation (IndraDrive only).....	127
Parameter set for G63 and position-controlled spindle mode (IndraDrive only).....	127
Parameterize manual probing.....	547
Parameterizing example: auxiliary functions....	231
ParkAx.....	562
Parking axis.....	562
Path acceleration.....	391
Path data (Path).....	69
Path deceleration acceleration.....	391

Index

Path of the PLC application.....	179	PosWinSel.....	440
Path velocity (PathVel).....	296	PoTrvLim.....	91
PathMode.....	528	PrecFeedInch.....	254
PercBuffered.....	286	PrecFeedMetr.....	253
Percentage distribution of the total number of blocks.....	284	Precision programming (PrecProg).....	312
Percentage number of blocks		PrecLinInch.....	259
for 3D milling cutter path correction.....	287	PrecLinMetr.....	258
for block buffer.....	286	PrecPosInch.....	252
for jerk-limited velocity control.....	285	PrecPosMetr.....	251
for number for spline functions.....	286	PrecRotInch.....	259
Percentage of number of blocks		PrecRotMetr.....	259
for other functions.....	287	PrecSpSpeed.....	254
PercMiscFunc.....	287	Predefined settings.....	524
PercSpline.....	286	Preferred axis in the tool editor.....	526
PercTCorr3D.....	287	PrelPlaceBcd32.....	208
PercVelCalc.....	285	Prep/ChAxNames[1..8].....	507
PercWin.....	131	PrepAxConf[1..8].....	511
Phase offset		Preselected tool edge number.....	601
Speed to enter the phase offset.....	503	PreTool.BQ1.....	613, 628
Phi.....	617	PreTool.BQ2.....	613, 629
phi/theta/psi.....	590	PreTool.BQ3.....	614, 629
Physical drive designation.....	117	PreTool.IKQ1.....	610, 626
Physical drive name.....	55	PreTool.IKQ2.....	611, 626
Physical unit.....	15	PreTool.IKQ3.....	611, 627
Placement banks.....	258	PreTool.IQ1.....	611, 627
PLC selection.....	179	PreTool.IQ2.....	612, 627
PLC/Plc context menu.....	179	PreTool.IQ3.....	612, 628
PlcMemIL.....	180	PreTool.K1.....	609, 624
PlcOriPos.....	573	PreTool.K2.....	609, 625
PlcSymbFilePath.....	179	PreTool.SKQ.....	610, 625
Polar coordinate programming (PolarCoord)....	273	PrgCpNumber.....	484
PolarCoordAngle1.....	273	PrgSpNumber.....	483
PolarCoordAngle2.....	273	Printed circuit board number.....	598
PosAdjus[1..8]t.....	509	PrioAddAx.....	250
PosCtrlCAxSet.....	127	PrioAxChWin.....	248
PosCtrlSpSet.....	127	PrioChPack.....	251
Position.....	330	Probe/commissioning data.....	546
Position adjustment required.....	471	Probe1Edge.....	354
Position controller Kv factor.....	173	Probe2Edge.....	354
Position display (PosDisp).....	241	Processor load (Perf).....	295
Position number of spindle tool.....	635	Processor overtemperature warning.....	530
Position of the reference point.....	167	Processor temperature.....	588
Position offset of multiple-spindle.....	634	ProgFeed.....	537
Position window (InPosWin).....	297	ProgModeAreaCtrl.....	333
Positioning logic.....	106	Program name "Block pre-run" (optional).....	639
Positioning logic, can be switched.....	108	Program name for the coordinate selection.....	324
Positioning Logic/Enable Modulo Calculation (PosLog).....	105	Program name for the default configuration.....	324
Positive stop.....	375	Program name of tool header (optional).....	638
Positive travel limit.....	91	Program name of tool trailer (optional).....	638
PosLoopKv.....	173	ProgSelFlags.....	432
PosLowLim.....	340	Psi.....	617
PosNotAdjusted.....	470	Punching and nibbling (PUN).....	41
PosOffsPoStop.....	377	Punching and nibbling (PUN) - Optional pa- rameters.....	357
PosUpLim.....	339		
PosWayPoint.....	330	R	
PosWin[1..2].....	441	R.....	615

Radius programming.....	307	Reference position[1..8].....	404
RadTolCentProg.....	306	Reference switch monitoring.....	347
RadTolRadProg.....	307	RefPos.....	167
Rapid traverse.....	69, 326	RefPosTrafo[1..8].....	404
Rapid traverse override (RapTravOvrd).....	72	Release channel.....	472
RapidAxAcc.....	98	ReleaseChan.....	472
RapidAxJerk.....	99	Releasing the tool-specific limitations.....	574
RapidAxVel.....	104	Remaining time up to the switching point.....	431
RapidVelWcs.....	326	RemTime[].....	431
Re-entry (ReEntry).....	321	ReqLookAhead.....	429
Re-initialization of SD elements after NC restart		ReqPosAdjust.....	471
One-time.....	564	Reserve buffers, number.....	351
Repetitive.....	564	Reset.....	436
Re-initialization of selected SD elements after NC restart.....	563	Reset CPL customer outputs to PLC.....	180
Reaction-triggering error sources of system axis coupling.....	442	Reset system axis coupling.....	436
Real-time condition range of the system axis coupling		RetPaStrategy.....	584
Lower switching position, switching velocity.....	448	Retrace logic	
Upper switching position, switching velocity.....	449	Adjustment program (RetraceAsup) active.....	558
Real-time conditioned master axis switching		Retrace logic (RetraceAsup)	
Active input revolutions.....	465	Adjustment program name (RetraceAsup).....	557
Active master axis.....	464	Retrace logic data.....	557
Active offset.....	464	Retract position for error handling of system axis coupling.....	445
Active output revolutions.....	465	Retract vector of retract axis	
Active table	466	Main axis.....	393
Real-time conditioned master axis switching of a system axis coupling.....	448	Secondary axis.....	394
Real-time coordinates		RetrVectRetrAx.....	393
ACS.....	533	RetrVectSeconAx.....	394
BCS.....	534	Return point.....	320
MCS.....	536	Return strategy.....	319
WCS.....	539	Return to contour.....	555
Real-time master axis offset		Return to path	
Active offset.....	466	Adjustment programs (ReentryAsup).....	555
Real-time path velocities		Return to path (ReturnPath).....	319
Programmed path velocity.....	537	Return to path logic	
Real-time velocities		Adjustment program (ReentryAsup) active.....	557
Current path velocity.....	538	Return to path logic, data.....	552
RecoupleEnable.....	437	Return to path radius.....	554
RedChVel.....	69	Return to path velocity	
RedMaxTorq.....	111	Individual.....	553
Reduced maximum torque when torque reduction enabled.....	111	Optional.....	553
Reduced rapid traverse.....	69	Return to path velocity selection.....	556
ReEntryDataPath.....	321	ReturnMode.....	319
ReEntryFuncMode.....	321	ReturnPoint.....	320
ReEntryIndex.....	580	Reversed direction of the spindle rotation.....	579
ReentryInfoDir.....	521	ReverseRotation.....	579
ReEntryStore.....	586	Rigidity.....	440
Reference of the stroke release times.....	358	Rotary axis.....	59
Reference orientation of the tool coordinate system TCS(TcsRefOri).....	407	Rotary coordinates.....	85
Reference orientation of tool coordinate system.....	476	Rotate tool for change in direction.....	384
		RotAxTangT.....	383
		Runtime monitoring for block pre-run.....	583
		Runtime warning	
		For block cycle time problems.....	540
		For blocked current block switching.....	541

Index

For uncompleted action at the current block switching.....	541	of the control.....	561
S		of the devices.....	562
Safety Engineering (SAFE) - Optional parameters -	341	of the subnet.....	561
Safety functions (SPF) - Basic parameters.....	329	Sercos parameter of the external spindle encoder.....	132
Safety instructions.....	6	Sercos parameterization	
Save program stack (call chain) at reset.....	586	Settings for automatic Sercos parameterization.....	562
Scale factors (VelScaleFact).....	307	Sercos parameters.....	364
Scaling factor for feed programming.....	307	Online correction in WCS.....	370
Scaling factor for feed/revolution.....	308	Sercos parameters (SCSP) - Basic parameters -	171
Scaling factor for metric rotary axis velocity.....	311	Sercos ring (ScsIf).....	165
Scaling factor for non-metric Rotary axis velocity.....	310	Sercos system functions (SCS).....	62
Scaling factor for save velocity.....	342	Sercos timing.....	651
Scaling factors for American controls.....	309	Sercos(1,2)/FPGA.....	600
Scaling factors for European controls.....	309	Sercos(1,2)/Type.....	600
Scaling of a jog increment.....	101	Sercos(1,2)/Version.....	599
Scaling of jog increment.....	323	Serial number of the control hardware.....	598
ScndCP.....	530	Service hotline.....	675
ScsAddr.....	62	SetFanTime.....	527
ScsAddress.....	559	SETUP.ADJM6PRG.....	639
ScsCycTime.....	165	SETUP.MAGFIX.....	636
ScsIdCmdVal.....	167	SETUP.MAGSEC.....	636
ScsParMcsCorr.....	364	SETUP.MODE.....	634
ScsParSpEnco.....	132	SETUP.POSTPRG.....	638
ScsParWcsCorr.....	370	SETUP.PREPRG.....	638
ScsTimeOut.....	165	SETUP.SEARCH1.....	637
SDForceCmd1SnglRestart.....	564	SETUP.SEARCH2.....	637
SDForceCmd2EachRestart.....	564	SETUP.SEARCH3.....	637
Search path for link tables.....	293	SETUP.SPINDLES.....	634
Search paths.....	291	SETUP.SPIPLA.....	635
Search paths for subroutines.....	291	SETUP.SPISEC.....	635
Sector number of magazine.....	636	SETUP.SPOFFSET.....	634
Sector number of spindle tool.....	635	SHAPE order for axis.....	94
SelBlkSwiStrategy.....	162	SHAPE order for jerk-limited path mode (G8).....	73
Select data groups.....	21	SHAPE order for jerk-limited Positioning mode (G9).....	73
Select permitted synchronization deviation of system axis coupling.....	440	Showing parameter attributes.....	24
Selecting a data group.....	21	Sig[].DistCalc.....	426
Selecting display areas.....	20	Sig[].DistOff.....	424
Selecting the block switching strategy.....	162	Sig[].DistOn.....	423
Selection of drive function types.....	53	Sig[].Duration.....	426
Sensor selection (device).....	363, 369	Sig[].Index.....	427
Sercos address.....	62, 559	Sig[].TimeOff.....	425
Sercos communication (SCS) - basic parameters -	165	Sig[].TimeOn.....	424
Sercos communication to drive.....	57	Sig[].Type.....	427
Sercos cycle time (= interpolation cycle time).....	165	SigCtrlWAssign.....	177
Sercos data handling.....	166	SigCtrlWConf.....	176
Sercos devices.....	559	Sign of TCS_X in BCS.....	408
Sercos ID for NC command values.....	167	Sign of TCS_Y in BCS.....	409
Sercos initialization.....	649	Sign of TCS_Z in BCS.....	411
sequence.....	649	Signal alert symbol.....	6
Sercos interface.....	560	Signal control word.....	176
Sercos IP address		Signal control word configuration.....	176
		Signal duration.....	426
		Signal status word.....	175
		Signal status word configuration.....	175

Signal type selection.....	427	Speed window	130
Signal words.....	6	Speed window in percent.....	131
SignalStatus.....	422	Speed window in RPM.....	130
SigStatWAssign.....	176	SpeedGearRatio.....	572
SigStatWConf.....	175	SpeedStr.....	485
Simulation.....	420	SpGear1Sel.....	140
Slave spindle		SpGear2Sel.....	141
Channel spindle number.....	506	SpGear3Sel.....	142
Slow.....	325	SpGear4Sel.....	143
SlowAxVel.....	102	SpGearAutoSel.....	140
SlowVelWcs.....	325	SpGearIdleSel.....	143
Smoothing/filter.....	367	SpGrCCW.....	145
Software limit switch (TrvLim[1..2]).....	90	SpGrCCWCool.....	146
Software limit switches.....	90	SpGrCW.....	144
Sp.AutoGearShift.....	495	SpGrCWCool.....	145
Sp.CAxisNotAdj.....	491	SpGrGear1Sel.....	149
Sp.CAxisOn.....	495	SpGrGear2Sel.....	149
Sp.CAxisType.....	496	SpGrGear3Sel.....	150
Sp.CcvAxis.....	497	SpGrGear4Sel.....	151
Sp.CcvCoordSystem.....	498	SpGrGearAutoSel.....	148
Sp.CcvNotAdj.....	491	SpGrGearIdleSel.....	151
Sp.CcvOn.....	497	SpGrInd.....	154
Sp.CcvVelocity.....	499	SpGrIndSpGrDisp.....	255
Sp.CoupleDistance.....	501	SpGrPos.....	147
Sp.CoupleNotAdj.....	493	SpGrStop.....	147
Sp.CouplePosOffset.....	503	SpInd.....	61
Sp.CouplePosOffsetSpeed.....	503	SpIndChSp.....	153
Sp.CpDataNotAdj.....	493	Spindle.....	53
Sp.DriveNo.....	487	Spindle actuation for channel [k=1..n].....	135
Sp.Gear.....	494	Spindle assignment.....	61
Sp.GearNotAdj.....	489	Spindle can be actuated from the channel.....	135
Sp.IsProg.....	487	Spindle clockwise rotation.....	136, 144
Sp.LimitNotAdj.....	490	Spindle clockwise rotation - Coolant on....	136, 145
Sp.Move.....	494	Spindle counterclockwise rotation.....	137, 145
Sp.MoveNotAdj.....	489	Spindle counterclockwise rotation - Coolant on.....	138, 146
Sp.NotAdjusted.....	488	Spindle empty.....	640
Sp.OpModeNotAdj.....	492	Spindle Functions (SpFunc).....	115
Sp.OriPos.....	499	Spindle group functions (SpGrFunc).....	144
Sp.PosCtrlOn.....	498	Spindle group syntax [1..4].....	144
Sp.ProgMaxSpeed.....	501	Spindle groups.....	116
Sp.ProgMinSpeed.....	500	Spindle index.....	61
Sp.ProgSpeed.....	500	Spindle management (SpAdmin).....	152
Sp.SpeedNotAdj.....	490	Spindle mode.....	389
Sp.SyncErrorWindow.....	502	Spindle number in pre-run	
Sp.SyncWindow.....	502	Highest programmed spindle number.....	483, 484
Sp.SysSpNo.....	486	Spindle number in the channel	
SpCCW.....	137	Highest spindle number available.....	483
SpCCWCool.....	138	Spindle override.....	128
SpConfNotAdjusted.....	488	Spindle positioning.....	139, 147
SpCW.....	136	Spindle pre-run data.....	481
SpCWCool.....	136	Spindle programmed in pre-run.....	487
Speed.....	569, 572	Spindle speed.....	572
Programmed speed.....	500	Spindle stop.....	138, 147
Speed display.....	254	Spindle syntax [1..8].....	136
Speed limit value		Spindle syntax for system / channel spindles...	135
Maximum speed.....	501	Spindle system functions (SpFunc).....	61
Minimum speed.....	500		
Speed syntax of a spindle.....	485		

Index

Spindle type.....	118	Strategy 2: Spindle oriented.....	158
Spindle with C-axis operation.....	53	Strategy 2: Spindle standstill.....	157
Spindle/turret axis.....	134	Strategy 3: Gear range changed.....	161
Spindles.....	52	Strategy 3: Speed reached.....	159
Spindles (SP) - Basic parameters.....	115	Strategy 3: Spindle oriented.....	160
Spindles in the "ChSp" selection (channel spindle mode).....	153	Strategy 3: Spindle standstill.....	160
Spindles in the "Sp" selection (compatibility mode).....	153	Strategy determination for block switching control.....	162
SpIndSpDisp.....	255	Stroke monitoring.....	357
Spline parameters.....	379	Stroke monitoring time.....	357
Spline variant for vector orientation.....	381	Stroke release times.....	359
Splines (SPL).....	43	StrokeMonTime.....	357
Splines (SPL) - Optional parameters.....	379	Structure display area.....	13
SplineType.....	380	Subroutine name	
Splitting.....	110	Modal subroutine SwiOn[1..63] / SwiOff.....	270
SpModeThrd.....	389	Non-modal M-functions MFunc[1..8].....	267
SpOvrDStep[1..32].....	128	Non-modal M-functions MFunc[1..80].....	263
SpPos.....	139	Subroutine Technology (SubProg).....	261
SpStop.....	138	Suggestions.....	8
SpType.....	118	SummerDiffUtc.....	302
SRAM assignment.....	640	SummerLoc.....	301
SrchPathLinkTab.....	293	SummerZoneName.....	301
SrchPathSubProg.....	291	Support.....	675
Standstill torque limit.....	112	SupprChDisp.....	240
Standstill window.....	131	SupprCmdValOut.....	58
StandstillWin.....	131	Suppressing message output in block pre- run.....	586
StandstModeBlkTransferLock.....	313	SuppressMsg.....	586
Start bit address.....	206	SupprSafeTechNc.....	341
Start code.....	205	SwiOffSynSubProg.....	268
Start of range.....	180	SwiOn[1..63]/NameSubProg.....	270
StartAngTangT.....	387	SwiOn[1..63]/NofParSubProg.....	272
StartBitAddr.....	206	SwiOnSynSubProg.....	268
StartCode.....	205	SwiSpeed.....	124
Starting and exiting correction build-up strat- egy for tool path correction.....	277	SwiSummerDate.....	304
Starting angle for the offset angle.....	387	SwiSummerTime.....	305
State.....	456, 467	Switch speed to 2nd acceleration of Gear- Step[1..4].....	124
State after startup.....	65	Switch system axis coupling on/off.....	435
State of the system axis coupling table	467	Switch-off function SwiOff.....	268
State of the ZO functionality.....	641	Switch-off syntax.....	268
StateAddOn.....	457	Switch-on function SwiOn[1..63].....	268
Status of bit events.....	421	Switch-on syntax.....	268
Status: all signals.....	422	SwitchCond.....	448
Storage directory of re-entry data.....	521	Switched-on state.....	65
Storage mode.....	171	Switching on C-axis.....	495
StorMode.....	171	Switching on constant cutting velocity.....	497
Straight line angle.....	529	Switching on coupling link.....	505
Strat2WaitSpGear.....	158	Switching on position interface.....	498
Strat2WaitSpOri.....	158	SwitchMax.....	449
Strat2WaitSpSpeed.....	155	SwitchMin.....	448
Strat2WaitSpStop.....	157	SwiWinterDate.....	303
Strat3WaitSpGear.....	161	SwiWinterTime.....	304
Strat3WaitSpOri.....	160	Symbols.....	8
Strat3WaitSpSpeed.....	159	SynchrMonWin.....	347
Strat3WaitSpStop.....	160	Synchronization mode for slave axis.....	439
Strategy 2: Gear step changed.....	158	Synchronization type of the high-speed I/Os....	182
Strategy 2: Speed reached.....	155	Synchronization window error for coupling.....	502

Synchronization window for coupling.....	502	SysSRun.....	579
Synchronous axis.....	60	SysStates.....	587
Synchronous machining axis.....	59	SysSwivel.....	589
Synchronous run monitoring.....	461	SysTargetInfo.....	593
Synchronous run monitoring on.....	346	SysTch.....	633
Synchronous run monitoring window.....	347	System axes of the axis transformation.....	474
SyncMode.....	439	System axes/coordinates of forward transformation (FwdInCoord).....	403
Syntax of angle 1.....	273	System axis classification of the axis transformation.....	474
Syntax of angle 2.....	273	System axis coordinate index (SACI) [1..8].....	403
Sys_Simulation.....	420	System axis coupling command.....	435
SysActBitEv.....	421	System axis coupling control.....	453
SysAdvDistSig.....	421	System axis coupling status.....	456
SysAPrg.....	431	System axis coupling tables.....	467
SysAxActData.....	434	System axis coupling with additional interpolation.....	436
SysAxCoupleCmd.....	435	System axis coupling with recoupling in case of asynchronism.....	437
SysAxCoupleCtr.....	453	System axis coupling: Delay before error handling.....	443
SysAxCoupleTab.....	467	System axis index of the master axis of the system axis coupling.....	447
SysAxSRun.....	469	System axis name for the basic zero point offset.....	643
SysAxTrafo1.....	472	System axis number.....	329, 374
SysAxTrafo2.....	472	System axis number of the interface for jogging in the WCS.....	328
SysAxType.....	59	System axis number of the master.....	348
SysCfgInfo.....	478	System axis number of the rotary axis.....	383
SysChSpActData.....	478	System axis offset for the basic zero point offset.....	644
SysChSpSRun.....	481	System axis/system coordinate[1..8].....	405
SysChSRun.....	506	System control.....	527
SysChTestMode.....	522	System coordinate classification of the axis transformation.....	474
SysChWaitBitEv.....	523	System coordinates of the axis transformation.....	474
SysCoordSystem.....	524	System data.....	417, 420
SysCorrUnit.....	313	System hardware status information.....	530
SysCtrl.....	527	System interpolation axis positions.....	532
SysDrName.....	55	System interpolation basic workpiece positions.....	533
SysHandwWCS.....	527	System interpolation machine positions.....	535
SysHwStates.....	530	System interpolation path velocities.....	536
SyslpoAcsPos.....	532	System interpolation workpiece positions.....	538
SyslpoBcsPos.....	533	System memory.....	288
SyslpoMcsPos.....	535	System name.....	55
SyslpoPathFeed.....	536	System simulation.....	420
SyslpoWcsPos.....	538	System spindle command data.....	570
SysKernelDgn.....	540	System spindle definition.....	115
SysMaxAxNbr.....	542	System spindle index.....	61
SysMaxEd.....	542	System spindle number.....	486
SysNcDgnCtrl.....	543	System spindle, actual data.....	569
SysPermUsed.....	567	System spindles[1..60] - Spindle group[1..4]....	153
SysPrBase.....	546	System spindles[1..60] - Spindle group[1..8]....	153
SysPrMan.....	547	System status information.....	587
SysPrResult.....	551	System units (CorrUnit).....	313
SysReentry.....	552		
SysRetrace.....	557		
SysScsDevice.....	559		
SysScsDrPar.....	562		
SysScsIpAdr.....	560		
SysSDForce.....	563		
SysSdInfo.....	565		
SysSercCfgInfo.....	641		
SysSpActData.....	569		
SysSpCmdData.....	570		
SysSpName.....	117		
SysSRamMemInfo.....	640		

Index

SysTIActMaxEd.....	601
SysTmpUsed.....	568
SysTool.....	602
SysToolEd.....	614
SysToolInsert	617
SysToolSRun.....	618
SysToolType.....	631
SysVolUsed.....	568
SysZOData.....	641, 642
SysZODataA.....	643
SysZODataC.....	644

T

Tab.....	452, 466
Table display (TabDisp).....	258
Table for the system axis coupling.....	452
Table source file of the system axis coupling table	467
TangCircleRadius.....	554
Tangential tool control (TTL).....	43
Tangential Tool Guidance (TTL) -Optional parameters.....	383
Targ/ChAxNames[1..8].....	508
TargAdjPrgActive.....	514
TargAxConf[1..8].....	512
TargBlockSrc.....	519
TargChan.....	471
TargDNr.....	630
TargEd.....	630
Target block adjustment program active.....	514
Target block code.....	519
Target channel.....	471
Target groups.....	1
Target information.....	593
Target position in the axis coordinate sys- tem.....	510
Target position of approaching block.....	510, 587
TargetBlkAvailable.....	520
TargetBlkCheck.....	584
TargPos[1..8].....	510, 587
TCorrNotAdj.....	631
Tcs.....	398
TCS coordinate.....	399
TCS coordinate X.....	407
TCS coordinate Y.....	409
TCS coordinate Z.....	410
TCSOrientation[1..8].....	476
TechAxAdjust.....	512
Technological axis adjustment.....	512
Technology display for feed and speed (TechDisp).....	253
Telegrams.....	174
Terms.....	37
Test mode.....	168, 522
TestFeedrate.....	522
TestRapid.....	523
Text colors.....	18

Theta.....	617
Thread cutting (THRD).....	43
Thread cutting (THRD) - Optional parame- ters.....	389
Thread cutting with spindle override.....	578
ThreadOverride.....	578
Time	
Clock change to summer time.....	305
Clock change to winter time.....	304
Time difference UTC >Local time	
Summer.....	302
Winter.....	300
Time for fan change.....	527
Time monitoring for Sercos startup.....	165
Time of enabling.....	15
Time stamp of the last update.....	430
Time stamp of the system axis coupling ta- ble	468
Time zone.....	298
TimeOffs.....	359
Timeout.....	583
TimeRef.....	358
TimeStamp.....	430
TLWP/TCorr/Ch[k]/ context menu.....	397
Tolerance from which a runtime error is generated	
Center point programming.....	306
Radius programming.....	307
Tool and workpiece management (TLWP).....	44
Tool and workpiece management (TLWP) - Optional parameters -.....	397
Tool change cycle TCH (variables)	633
Tool changer type.....	634
Tool correction calculation.....	397
Tool corrections in TCS-coordinates - Fix as- signment (Tcs).....	398
Tool corrections in the active WCS working plane (WcsActPlane).....	397
Tool corrections in WCS coordinates-fixed assignment.....	397
Tool corrections per edge.....	614
Tool data element BQ3 in the tool catalog.....	632
Tool edge numbers of the tool correction.....	601
Tool length corrections.....	397
Tool path correction (CutComp).....	277
Tool search condition 1.....	637
Tool search condition 2.....	637
Tool search condition 3.....	637
Tool type data of the tools in the tool catalog...	631
Tool.Enable.....	574
Tool.MaxAcc.....	575
Tool.MaxPosAcc1.....	575
Tool.MaxPosAcc2.....	576
Tool.MaxSpeed.....	574
Tool.MaxSwiSpeed.....	576
Tool.N_Input.....	577
Tool.N_Output.....	578
Tooth pitch of the slave axis.....	438
Torque limit value.....	173, 376

Torque reduction (TorqRed).....	111
Total number of blocks for block preparation and interpolation.....	280
touch probe function available.....	353
touch probes.....	353
TrafoProgName.....	324
TrafoType.....	401
Transformation (TRA).....	44
Traveling direction.....	338
Traveling to positive stop (MPS) - optional parameters.....	375
Traverse to positive stop can be enabled.....	375
Trigger coupling error in system axis coupling.....	442
Troubleshooting MTX advanced: NC configurator reports error 31 after change in parameterization.....	32
TrqLim.....	173
Turret mode and measuring system.....	134
Type.....	473
Type (parameter type).....	14
Type number in the tool catalog.....	631
Type of coordinate system determination in the tool editor.....	526
Type of distance calculation.....	426
Type of location programming.....	109
Type of monitoring area.....	332
Type of the Sercos connection 1 or 2.....	600
Type selection.....	188
TypeNo.....	631

U

Undoing changes to parameters.....	24
Unintended use.....	10
Consequences, disclaimer.....	9
Unit.....	423
Unit of the distance.....	423
Units for tables and corrections.....	313
Units of the basic ZO values.....	642
Update actual values in workpiece coordinates.....	241
UpdateTimePosDisp.....	241
Upper limit for gantry command value offset....	345
Upper limit value.....	339
Upper speed limit value of the spindle.....	571
Upper speed limit value of the tool.....	574
Upper velocity limit value.....	413
UpVelLim.....	413
UsabAuxFunc.....	218
Usable auxiliary functions.....	218
User parameter display.....	174

V

Validate.....	453
Validity of the active tool.....	603
Validity of the remaining time "RemTime".....	430
Validity[].....	430

Value.....	524
VarDistAxJog.....	101
VarDistAxVel.....	105
Variable jog distance.....	101, 323
Variable name.....	365
Online correction in WCS.....	371
Variable number of edges in the tool catalog....	632
VarStepVelWcs.....	327
VectOriSpline.....	381
VelCouple.....	438
VelJumpThrd.....	390
VelLoadDgn/DgnMode.....	544
VelLoadDgn/PeakTmpHoldTime.....	544
Velocity (Vel).....	390
Velocity axis load diagnostics	
Dwell time for temporary velocity peak values.....	544
Velocity data (Vel).....	69
Velocity data for the WCS jog interface.....	325
Velocity jump.....	390
Velocity-dependent output (VDO).....	45
Velocity-dependent output (VDO) - optional parameters.....	413
VelWeightFact.....	342
Version of the Sercos connection 1 or 2.....	599
vertical.....	525
Virtual drive.....	57
Virtual mode (VirtMode).....	167

W

WaitForSpindle.Active.....	478
WaitForSpindle.ChanSpNo.....	479
WaitForSpindle.Gear.....	481
WaitForSpindle.Move.....	480
WaitForSpindle.SysSpNo.....	479
Waiting condition when waiting on bit events...	523
Waiting points in time for block switching control with strategy 2.....	155
Waiting points in time for block switching control with strategy 3.....	159
Warnings.....	6
Waypoints (WayPoint).....	329
WcsActPlane.....	397
WcsFixed.....	397
WcsJogLf.....	328
WcsPosDisp.....	242
Web server (WebServ).....	316
WinterDiffUtc.....	300
WinterLoc.....	299
WinterZoneName.....	299
Workpiece coordinates (WCS).....	84
Workspace.....	332
World coordinate system (WCS).....	242

Z

Z_axis_pref.....	526
Zero point offset banks.....	257

Notes

Notes

Bosch Rexroth AG

P.O. Box 13 57

97803 Lohr a.Main, Germany

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr a.Main, Germany

Phone +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911400176