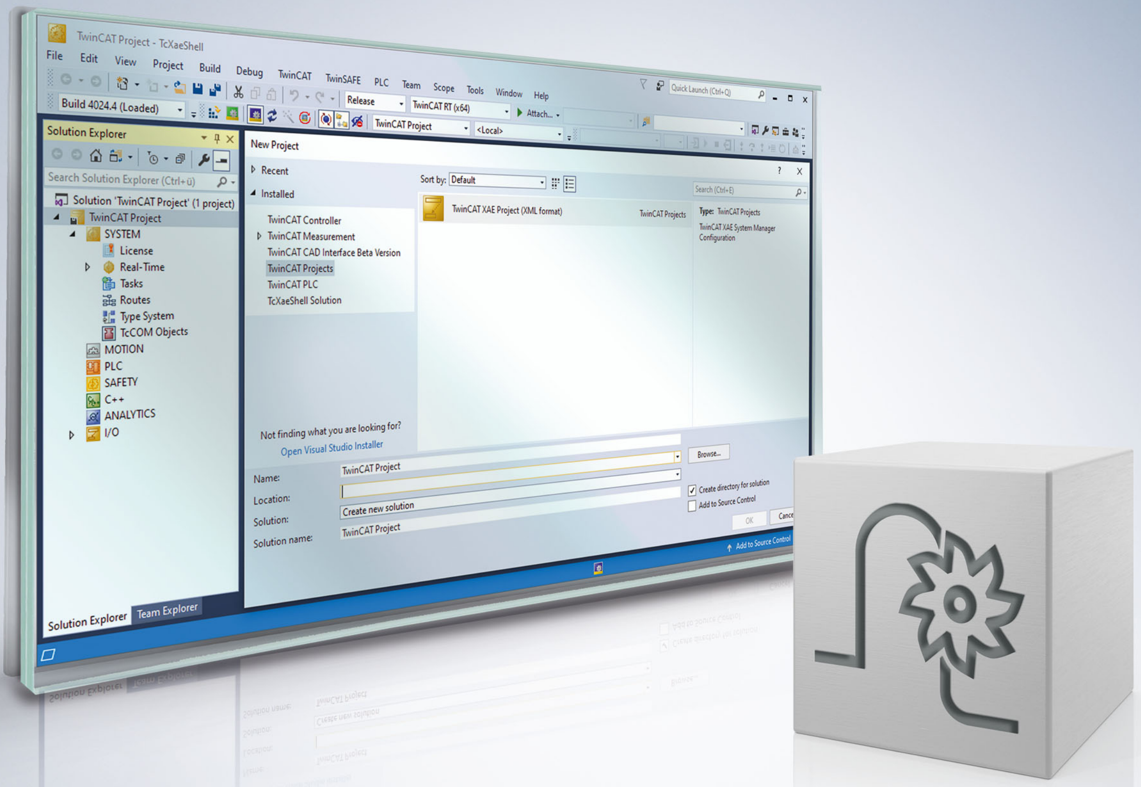
BECKHOFF New Automation Technology

Manual | EN

# TF5200 | TwinCAT 3 CNC

## Channel parameters



12/19/2023 | Version: 1.29

# Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

### Disclaimer

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

### Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

### Patent Pending

The EtherCAT technology is patent protected, in particular by the following applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

### Copyright

# General and safety instructions

### Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

### Icons in explanatory text

1. Indicates an action.

⇨ Indicates an action statement.

**⚠ DANGER**

**Acute danger to life!**

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.

**⚠ CAUTION**

**Personal injury and damage to machines!**

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.

***NOTICE***

**Restriction or error**

This icon describes restrictions or warns of errors.

**Tips and other notes**

This icon indicates information to assist in general understanding or to provide additional information.

### General example

Example that clarifies the text.

### NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.

**Specific version information**

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

# Table of contents

**Notes on the documentation** ........ 3

**General and safety instructions** ........ 4

**Overview of all channel parameters** ........ 17

**1 Description of elements** ........ 41

1.1 Configuration of the NC channel. ........ 41

1.1.1 Decoding ........ 41

1.1.2 Tool radius compensation ........ 52

1.1.3 Path preparation ........ 54

1.1.4 Interpolation ........ 61

1.2 Configuration of PLC functions ........ 65

1.2.1 Settings for M functions ........ 65

1.2.2 Settings for H functions ........ 83

1.2.3 Settings for tool change (tool.* or werkzeug.*) ........ 86

1.2.4 Process time of homing (P-CHAN-00077) ........ 88

1.2.5 Calculation model for M/H pre-output time (P-CHAN-00209) ........ 88

1.2.6 Tolerance for deviation with pre-output position of M/H function (P-CHAN-00760) ........ 90

1.2.7 Residual path/time calculation with M/H pre-output (P-CHAN-00212) ........ 90

1.2.8 Number of NC blocks at M/H code look ahead (P-CHAN-00274) ........ 91

1.3 Dimensional units and resolutions ........ 92

1.3.1 Dimensional unit for translatory axes (P-CHAN-00054) ........ 92

1.3.2 Dimensional unit for rotary axes (P-CHAN-00079) ........ 92

1.3.3 Scaling factor of axis resolution (P-CHAN-00315) ........ 92

1.3.4 Resolution of linear axes (P-CHAN-00034) ........ 93

1.3.5 Resolution of rotary axes (P-CHAN-00078) ........ 93

1.3.6 Resolution of spindles (P-CHAN-00083) ........ 94

1.4 Settings for tool management ........ 94

1.4.1 Implicit execution of D word with T (P-CHAN-00014) ........ 94

1.4.2 Tool data is located in an external tool management (P-CHAN-00016) ........ 95

1.4.3 Initiating FLUSH mechanisms ........ 97

1.4.4 Tool life data recording ........ 97

1.5 Settings for edge banding ........ 99

1.5.1 Residual path (P-CHAN-00030) ........ 99

1.5.2 Block global edge banding (P-CHAN-00029) ........ 99

1.5.3 Reject rest block (P-CHAN-00149) ........ 100

1.5.4 Modal active measurement (P-CHAN-00153) ........ 101

1.5.5 Suppress motion path check (P-CHAN-00227) ........ 102

1.5.6 Programmed measurement signal (P-CHAN-00257) ........ 103

1.5.7 Residual path related to contour (P-CHAN-00258) ........ 104

1.5.8 Warning on non-detection of probing signal (P-CHAN-00298) ........ 105

1.5.9 Warning message if residual motion path is too small (P-CHAN-00299) ........ 105

1.5.10 Outputting an MNE_SNS technology function at measuring event (P-CHAN-00435) ... 106

1.6 Settings for measurement ........ 107

1.6.1 Measurement type (P-CHAN-00057) ........ 107

1.6.2 Error reaction with measurement type 1 (P-CHAN-00176) 108
1.6.3 Active delay with measurement signal (P-CHAN-00214) 108
1.6.4 Error reaction with measurement type 7 (P-CHAN-00266) 108
1.6.5 Automatic travel range limit (P-CHAN-00268) 110
1.6.6 Active dynamic for measurement run (P-CHAN-00296) 110
1.6.7 Including ball radius in PCS measurement value (P-CHAN-00311) 112
1.6.8 Allow already actuated probes when starting a measurement run (P-CHAN-00312) 113
1.6.9 Measurement path without motion path (P-CHAN-00313) 113
1.6.10 Switch over between physical and clone axes during a measurement run (P-CHAN-00326) 114
1.6.11 Axes, measurement values and offsets in CS (P-CHAN-00327) 114
1.7 Global program effectiveness of P parameters (P-CHAN-00067) 115
1.8 Single-step operating mode (P-CHAN-00015) 115
1.9 Settings for diameter programming (P-CHAN-00091) 115
1.10 Settings for circle centre point correction and radius compensation 117
1.10.1 Permissible absolute centre point offset (P-CHAN-00059) 117
1.10.2 Permissible relative centre point offset (P-CHAN-00060) 118
1.10.3 Maximum permissible absolute radius difference (P-CHAN-00171) 118
1.10.4 Maximum permissible percentage radius difference (P-CHAN-00172) 118
1.11 Settings for RESET 120
1.11.1 Default axis configuration in the channel after reset (P-CHAN-00179) 120
1.11.2 Axis release after Reset (P-CHAN-00075) 121
1.11.3 Release of non-channel axes after Reset (P-CHAN-00115) 121
1.12 Retention of the last axis configuration (P-CHAN-00460) 121
1.13 Maximum channel override (P-CHAN-00056) 122
1.14 Sampling of override (P-CHAN-00065) 122
1.15 Mode of the override 123
1.15.1 Influence on feed (P-CHAN-00066) 123
1.15.2 Influence on acceleration (P-CHAN-00170) 123
1.15.3 Rapid traverse override (P-CHAN-00181) 124
1.15.4 Influence of time override on dwell time (P-CHAN-00111) 126
1.15.5 Influence of time override on dwell time (P-CHAN-00353) 127
1.16 Operating principle of PLC feed (P-CHAN-00102) 127
1.17 Settings for vector dynamic limits (vector.*) 127
1.17.1 Vector acceleration limit (P-CHAN-00002) 127
1.17.2 Path deceleration limit (P-CHAN-00208) 128
1.17.3 Radial acceleration limit (P-CHAN-00361) 128
1.17.4 Unit for path acceleration/deceleration and jerk (P-CHAN-00351) 128
1.17.5 Path velocity limit (P-CHAN-00090) 129
1.17.6 Dynamic limit in the circle (P-CHAN-00350) 129
1.18 Configuration of effector coordinate system (ECS) 129
1.18.1 Select the second axis of ECS (P-CHAN-00031) 130
1.18.2 Plane selection (P-CHAN-00050) 130
1.18.3 Examples 130
1.19 Tracking a rotary axis 133
1.19.1 Definition of the tracking axis (P-CHAN-00095) 133

1.19.2 Automatic orientation of a tracking axis (P-CHAN-00101) ... 133
1.19.3 Selection of tracking mode without stopping (P-CHAN-00109) ... 133
1.19.4 Suppress a rotary CS offset (P-CHAN-00144) ... 134
1.19.5 Tracking axis in the workpiece (P-CHAN-00185) ... 134
1.19.6 Including a contour rotation (P-CHAN-00215) ... 134
1.19.7 Suppress vector dynamic limitation (P-CHAN-00265) ... 135
1.19.8 PCS smoothing function for tracking axis (P-CHAN-00281) ... 135
1.20 Switch the default feed axis (P-CHAN-00096) ... 135
1.21 Valid deceleration ramp at FEEDHOLD (P-CHAN-00097) ... 136
1.22 Switch over the meaning of M functions (P-CHAN-00098) ... 136
1.23 Effectiveness of tool compensation data (P-CHAN-00100) ... 137
1.24 Configuration of path axes ... 138
1.24.1 Number of axis groups (P-CHAN-00023) ... 138
1.24.2 Axis group-specific elements ... 139
1.24.3 Axis structure (gruppe[i].achse[j].*) ... 139
1.25 Configuration of spindles ... 142
1.25.1 Number of spindles (P-CHAN-00082) ... 142
1.25.2 Spindle name within a path compound (P-CHAN-00010) ... 142
1.25.3 Machine ID with C axis face machining (P-CHAN-00008) ... 142
1.25.4 Definition of main spindle ... 143
1.25.5 Spindle data (spindel[i].*) ... 145
1.26 Settings for synchronous mode (synchro_data.*) ... 155
1.26.1 Default coupling specifications (synchro_data.koppel_gruppe[i].*) ... 156
1.27 Parameterisation of the change logging function (aep.*) ... 158
1.27.1 Logging mode (P-CHAN-00024) ... 158
1.27.2 Logging P parameters (P-CHAN-00068) ... 158
1.27.3 Logging self-defined variables (P-CHAN-00088) ... 158
1.27.4 Logging G functions (P-CHAN-00022) ... 158
1.27.5 Logging tool data (P-CHAN-00093) ... 160
1.28 Definition of macros (makro_def[i].*) ... 160
1.28.1 Specification of macro name (P-CHAN-00085) ... 160
1.28.2 Specification of macro content (P-CHAN-00062) ... 160
1.29 Settings for tool radius compensation (TRC) ... 161
1.29.1 Configuration with TRC (P-CHAN-00092) ... 161
1.29.2 Suppress warnings during contour masking (P-CHAN-00021) ... 161
1.29.3 Implicit contour masking (P-CHAN-00219) ... 161
1.29.4 Extended kerf detection for circular blocks (P-CHAN-00284) ... 162
1.30 Configuration of syntax check (syn_chk.*) ... 163
1.30.1 Number of errors per row (P-CHAN-00020) ... 163
1.30.2 Number of errors in an NC program (P-CHAN-00019) ... 163
1.30.3 Specifying the operation mode (P-CHAN-00028) ... 163
1.30.4 Write results of syntax check to file (P-CHAN-00416) ... 165
1.31 Parameterisation of kinematic transformations ... 165
1.31.1 Configuration for CNC Build V2.11.2xxx and higher: ... 165
1.31.2 Configuration for CNC Builds as of V3.00 and higher: ... 167
1.31.3 Parameters of the Universal Kinematics (kinematik[91].*, trafo[j].* ... 173

1.31.4 Parameters of the coupling kinematic ... 178
1.31.5 Orientation angle mode (P-CHAN-00112) ... 181
1.31.6 Activation of TCP display data (P-CHAN-00145) ... 182
1.31.7 Switchover of TCP display data (P-CHAN-00184) ... 184
1.31.8 Parameters for automatic alignment function (tool_ori_cs.*) ... 184
1.31.9 Transformation with gaps in the axis (P-CHAN-00213) ... 185
1.31.10 Tool orientation within the machining coordinate system (P-CHAN-00247) ... 186
1.31.11 Permissible calculation tolerance with forward / backward transformation (P-CHAN-00228) ... 187
1.31.12 Implicit selection of kinematic transformation (P-CHAN-00151) ... 188
1.31.13 Implicit deselection of kinematic transformation (P-CHAN-00152) ... 188
1.31.14 Mode of kinematic transformation (P-CHAN-00456) ... 189
1.31.15 Velocity limit of the TCP ... 189
1.32 Default settings at program start (prog_start.*) ... 191
1.32.1 Initialisation of G functions (prog_start.g_gruppe[i].*) ... 191
1.32.2 Initialisation of M functions (prog_start.m_gruppe[i].*) ... 193
1.32.3 Initialisation of slope parameters (prog_start.slope.*) ... 194
1.32.4 Initialisation of default feed rate (P-CHAN-00099) ... 195
1.32.5 Initialisation of feed rate unit (P-CHAN-00108) ... 196
1.32.6 Late synchronisation at program end (P-CHAN-00033) ... 196
1.32.7 Settings for Job Manager mode ... 196
1.33 Reduction of tangential transition velocity (P-CHAN-00009) ... 198
1.34 Including jerk in the polynomial (P-CHAN-00110) ... 198
1.35 Jerk limiting mode at block transition (P-CHAN-00117) ... 199
1.36 Additional consideration of curvature at block transition (P-CHAN-00245) ... 199
1.37 Look ahead speed limit (speed_limit_look_ahead.*) ... 200
1.37.1 Enable / disable (P-CHAN-00017) ... 200
1.37.2 Weighting of speed limit (P-CHAN-00089) ... 200
1.37.3 Unit (P-CHAN-00018) ... 201
1.37.4 Distance to corner (P-CHAN-00013) ... 201
1.37.5 Distance from corner (P-CHAN-00012) ... 201
1.37.6 Weighting the speed limit via override (P-CHAN-00155) ... 201
1.38 Settings for manual mode ... 203
1.38.1 Rejection of manual mode path (P-CHAN-00113) ... 203
1.38.2 Relative manual mode offset limits with G200 (P-CHAN-00114) ... 203
1.38.3 Override with manual mode (P-CHAN-00186) ... 203
1.38.4 Error message output at manual mode axis overrun (P-CHAN-00437) ... 205
1.38.5 Suppression of workspace monitoring in manual operation (P-CHAN-00442) ... 205
1.39 Exclusivity of measuring system programming (P-CHAN-00116) ... 205
1.40 Exclusivity of axes programming (P-CHAN-00148) ... 206
1.41 Subroutine call with M6 (P-CHAN-00118) ... 206
1.42 Subroutine call with D word (P-CHAN-00429) ... 207
1.43 Subroutine call at program start / MDI ... 208
1.43.1 Subroutine name (P-CHAN-00119) ... 208
1.43.2 Mode of subroutine call (P-CHAN-00260) ... 208
1.44 Subroutine call at program end / MDI ... 210

1.44.1 Subroutine name (P-CHAN-00252) ... 210
1.44.2 Mode of subroutine call (P-CHAN-00433) ... 211
1.45 Settings for software limit switch (SLS) monitoring ... 211
1.45.1 Software limit switch (SLS) monitoring with tolerance (P-CHAN-00120) ... 211
1.45.2 Error reaction for command value based software limit switch monitoring (P-CHAN-00147) ... 212
1.45.3 Suppress software limit switch monitoring on command value side (P-CHAN-00459) ... 212
1.45.4 Limitation of software limit switches in manual mode with parallel interpolation (P-CHAN-00759) ... 213
1.45.5 Behaviour with programmed software limit switches (P-CHAN-00498) ... 213
1.46 List interpretation with active NC program (P-CHAN-00146) ... 214
1.47 Adoption of block number when subroutine is called (P-CHAN-00150) ... 214
1.48 Parameterisation of HSC machining (hsc.*) ... 216
1.48.1 BSPLINE parameters (hsc.bspline.*) ... 216
1.48.2 General HSC parameters (hsc.gen.*) ... 220
1.49 Channel synchronisation on axis exchange (P-CHAN-00154) ... 224
1.50 Channel initialisation with actual values (P-CHAN-00455) ... 225
1.51 Program name for automatic streaming (P-CHAN-00158) ... 225
1.52 Modal effect of I/J/K for circle programming (P-CHAN-00159) ... 226
1.53 Subroutine calls using G functions ... 226
1.53.1 Implicit call with G80 (P-CHAN-00160 - P-CHAN-00169) ... 226
1.53.2 Implicit call with G81 (P-CHAN-00161) ... 227
1.53.3 Implicit call with G82 (P-CHAN-00162) ... 227
1.53.4 Implicit call with G83 (P-CHAN-00163) ... 228
1.53.5 Implicit call with G84 (P-CHAN-00164) ... 229
1.53.6 Implicit call with G85 (P-CHAN-00165) ... 229
1.53.7 Implicit call with G86 (P-CHAN-00166) ... 230
1.53.8 Implicit call with G87 (P-CHAN-00167) ... 231
1.53.9 Implicit call with G88 (P-CHAN-00168) ... 231
1.53.10 Implicit call with G89 (P-CHAN-00169) ... 232
1.53.11 Additional implicit calls with G8xx (P-CHAN-00187) ... 233
1.54 Active testing whether axis group is 'in position' (P-CHAN-00173) ... 233
1.55 Ignore internal stop conditions with rapid contour visualisation (P-CHAN-00183) ... 235
1.56 Behaviour with axis errors (P-CHAN-00218) ... 235
1.57 Path-dependent dynamic weighting ... 236
1.57.1 Activation of dynamic weighting (P-CHAN-00190) ... 236
1.57.2 Table of dynamic weightings (dynamic_weighting[i].*) ... 236
1.57.3 Motion path limit (P-CHAN-00191) ... 236
1.57.4 Weighting factor for speed (P-CHAN-00192) ... 237
1.57.5 Weighting factor for acceleration (P-CHAN-00193) ... 237
1.57.6 Weighting factor for ramp time (P-CHAN-00194) ... 237
1.57.7 Example of a table ... 238
1.58 Radius-dependent dynamic weighting ... 238
1.58.1 Activation of dynamic weighting (P-CHAN-00230) ... 239
1.58.2 Table of dynamic weighting (curve_dynamic_weighting[i].*) ... 239
1.58.3 Example of a table ... 240
1.59 Special manual mode data (man_mode.*) ... 242

1.59.1 Settings for Cartesian manual mode (man_mode.vector_limit.*) 242

1.60 Cycle programming settings 244

1.60.1 Modal effect of changes in the cycle (P-CHAN-00210) 244

1.60.2 Suppress NC blocks from cycles and M6 in trace data and display (P-CHAN-00211) 244

1.60.3 Memory size for @P parameters (P-CHAN-00481) 246

1.60.4 Automatic creation of @P parameters (P-CHAN-00463) 246

1.60.5 HSC settings for cycles 246

1.61 Settings for edge machining (edge_machining.*) 248

1.61.1 Enable / disable (P-CHAN-00220) 250

1.61.2 Critical bend angle (P-CHAN-00221) 250

1.61.3 Distance before the edge (P-CHAN-00222) 250

1.61.4 Feed before edge (P-CHAN-00223) 250

1.61.5 Waiting time in edge (P-CHAN-00224) 251

1.61.6 Distance after the edge (P-CHAN-00225) 251

1.61.7 Feed rate after the edge (P-CHAN-00226) 251

1.61.8 Switching feed rate adjustment (P-CHAN-00300) 251

1.61.9 Edge detection mode (P-CHAN-00301) 254

1.62 Decoder block ahead limiting 254

1.62.1 Block ahead limiting via number of blocks (P-CHAN-00216) 254

1.62.2 Block ahead limiting via number of NC motion blocks (P-CHAN-00246) 254

1.62.3 Time-based block ahead limiting (P-CHAN-00269) 255

1.62.4 Block ahead limiting in protected mode (P-CHAN-00270) 255

1.62.5 Selection/deselection of chamfers and roundings function (P-CHAN-00273) 255

1.62.6 Time ahead calculation based on mean feed velocity (P-CHAN-00428) 257

1.63 Exact stop with rapid traverse (P-CHAN-00235) 257

1.64 Check exact stop with additive movements (P-CHAN-00486) 257

1.65 Settings for lifting 258

1.65.1 Minimum path length for lift movements (P-CHAN-00244) 258

1.65.2 Lifting an axis with time-based calculation (P-CHAN-00345) 258

1.66 Settings for robotics 260

1.66.1 Orientation programming (ori.*) 260

1.66.2 Alternative programming of axis-specific movements in robotics (P-CHAN-00253) 262

1.66.3 Suppress an active kinematic transformation at G0 (P-CHAN-00423) 262

1.67 Limiting the path length of motion block (P-CHAN-00457) 263

1.68 Handling of polynomial blocks of zero length (P-CHAN-00254) 263

1.69 Settings for profile tube processing (tube_profile.*) 264

1.69.1 Type definition of technological functions (P-CHAN-00251) 264

1.69.2 Technological functions to display rounding transitions when exiting (P-CHAN-00250) 264

1.69.3 Technological functions to signal rounding transitions on engaging (P-CHAN-00249) 265

1.70 Optimised polynomial smoothing (P-CHAN-00259) 266

1.71 Configuration of interpolator input disable (P-CHAN-00267) 268

1.72 Settings for forward/backward movement on the path (forward_backward.*) 270

1.72.1 Backward movement with external position offsets (P-CHAN-00275) 271

1.72.2 M00 / M01 synchronisation 273

1.72.3 Automatic reversal after STOP 275

1.73 User-defined data (customer.*) 277

1.73.1 Free values (P-CHAN-00280) ........ 277
1.74 Encryption of NC programs (P-CHAN-00283) ........ 277
1.75 Settings for independent axes ........ 278
1.75.1 Position synchronisation of independent axes (P-CHAN-00297) ........ 278
1.75.2 Implicit synchronisation using independent asynchronous axes (P-CHAN-00451) ........ 278
1.76 Jog of path option after block search (P-CHAN-00305) ........ 279
1.77 Checking the license in the clone channel (P-CHAN-00306) ........ 280
1.78 Positioning of modulo axes on shortest path (P-CHAN-00346) ........ 280
1.79 Optimised insertion of #FLUSH CONTINUE (P-CHAN-00341) ........ 280
1.80 Enabling 2-path programming (P-CHAN-00261) ........ 281
1.81 Configuration of general display data ........ 282
1.81.1 Definition of channel display information ........ 282
1.81.2 Suppressing display of file name/file offset (P-CHAN-00180) ........ 282
1.81.3 Diameter display (P-CHAN-00256) ........ 283
1.81.4 Display format during machining simulation (P-CHAN-00121) ........ 283
1.81.5 Display of active path feed (P-CHAN-00328) ........ 283
1.81.6 Selecting the coordinate system to display axis positions (P-CHAN-00330) ........ 284
1.81.7 Display of axis command values and target points in Cartesian coordinate systems (P-CHAN-00331) ........ 284
1.81.8 Display of transformed actual positions (P-CHAN-00344) ........ 284
1.81.9 Consider mirroring in display and manual mode (P-CHAN-00434) ........ 285
1.81.10 Display motion range limits in PCS coordinate system (P-CHAN-00489) ........ 285
1.81.11 Display PCS positions inverse TRC (P-CHAN-00487) ........ 286
1.82 Logging of manual block commands ........ 287
1.82.1 Name of manual block log file (P-CHAN-00338) ........ 287
1.82.2 Maximum size of the manual block log file (P-CHAN-00339) ........ 288
1.83 Filter parameters for error handling in the NC channel (error_filter[i].*) ........ 289
1.83.1 Error cause (P-CHAN-00378) ........ 289
1.83.2 Error action (P-CHAN-00379) ........ 289
1.83.3 Conditional activation (P-CHAN-00380) ........ 290
1.83.4 Conditional action (P-CHAN-00381) ........ 290
1.83.5 Conditional filter activation (P-CHAN-00382) ........ 291
1.83.6 Output of additional error information (P-CHAN-00383) ........ 291
1.84 Settings for programming coordinate systems (coordinate_system.*) ........ 292
1.84.1 Coordinate system settings (CS, ACS, BCS) ........ 292
1.84.2 Transformation stack parameters ........ 294
1.84.3 Definition of rotation sequence (P-CHAN-00394) ........ 296
1.84.4 2-path programming: Selection of the display coordinate system (P-CHAN-00395) ........ 296
1.84.5 2-path programming: Selection of reference coordinate system (P-CHAN-00396) ........ 298
1.84.6 2-path programming: Managing axis offsets in tracked axes (P-CHAN-00397) ........ 301
1.84.7 2-path programming: Selection/deselection of intersection point calculation (P-CHAN-00398) ........ 301
1.85 Response of G91 with orientation axes (P-CHAN-00332) ........ 302
1.86 Use of axis numbers of master channel (P-CHAN-00282) ........ 304
1.87 Buffering motion blocks to optimise the feed rate profile (P-CHAN-00329) ........ 304
1.88 Units used in PLC Open functions (P-CHAN-00182) ........ 304
1.89 Starting NC programs without initialisation (P-CHAN-00347) ........ 305

1.90 Logical number of a NC channel (P-CHAN-00400) ........ 305
1.91 Settings for dynamic contour control (dcc.*) ........ 307
1.91.1 Selecting/cancellation of the dynamic contour control function (P-CHAN-00384) ........ 307
1.91.2 Selection of calculation method (P-CHAN-00385) ........ 307
1.91.3 REAL64 dynamic contour control input parameters (P-CHAN-00388) ........ 307
1.91.4 SGN32 dynamic contour control input parameters (P-CHAN-00389) ........ 307
1.92 Settings for surface feed rate (geo_feed_adapt.*) ........ 309
1.92.1 Selecting/deselecting the constant surface feed rate function (P-CHAN-00386) ........ 309
1.92.2 Selection of calculation method (P-CHAN-00387) ........ 309
1.92.3 REAL64 input parameters for surface feed rate (P-CHAN-00390) ........ 309
1.92.4 SGN32 input parameters for surface feed rate (P-CHAN-00391) ........ 309
1.93 NC program paths (path[i].*) ........ 310
1.93.1 Path specification (P-CHAN-00401) ........ 310
1.93.2 Logical path number (P-CHAN-00402) ........ 310
1.93.3 Path type (P-CHAN-00403) ........ 310
1.93.4 Priority (P-CHAN-00404) ........ 311
1.94 Settings for generating debug data (debug.*) ........ 312
1.94.1 Tracing parameters and variables (P-CHAN-00392) ........ 312
1.95 Configuration of channel interface (provide_channel_interface.*) ........ 312
1.95.1 Enabling channel interface for synchronous CS operations (P-CHAN-00399) ........ 312
1.96 Settings for 'Time preview' function ........ 313
1.96.1 Reserved distance for time calculation (P-CHAN-00307) ........ 313
1.96.2 Relating the reserved distance for time preview to contour (P-CHAN-00340) ........ 313
1.97 Merge short blocks for optimised contouring (P-CHAN-00321) ........ 313
1.98 Contouring method in inch units (P-CHAN-00439) ........ 314
1.99 Unit of constant cutting velocity (P-CHAN-00360) ........ 314
1.100 Orientation interpolation mode (P-CHAN-00417) ........ 315
1.101 Axis-specific orientation of tool length compensation (P-CHAN-00420) ........ 316
1.102 Deactivate feed factor effect (P-CHAN-00422) ........ 316
1.103 Settings for precalculation ........ 316
1.103.1 Precalculation - time offset (P-CHAN-00324) ........ 316
1.103.2 Precalculation mode (P-CHAN-00325) ........ 317
1.104 Dynamic look-ahead for axis polynomials (P-CHAN-00453) ........ 317
1.105 Memory size for contour machining (P-CHAN-00467) ........ 317
1.106 Real-time cycles ........ 317
1.106.1 Activating real-time cycles (P-CHAN-00406) ........ 317
1.106.2 Memory for real-time cycles (P-CHAN-00407) ........ 318
1.106.3 Max. execution time of realtime cycles (P-CHAN-00425) ........ 318
1.106.4 Number of elementary instructions for time check (P-CHAN-00426) ........ 318
1.106.5 Max. number of elementary instructions per CNC cycle (P-CHAN-00427) ........ 319
1.106.6 Max. number of actions in a real-time cycle (P-CHAN-00480) ........ 319
1.107 Compatibility for measuring with #CS/Transformation (P-CHAN-00440) ........ 320
1.108 Calculation mode of resulting ramp time with #CS and #TRAFO (P-CHAN-00758) ........ 320
1.109 Tolerance limit for recalculation with dynamic limitation of axis positions (P-CHAN-00751) ........ 321
1.110 Settings for thread tapping ........ 321
1.110.1 Tapping with actual positions of the spindle (P-CHAN-00761) ........ 321

1.110.2 Number of filter cycles to filter actual spindle positions (P-CHAN-00762) ..................... 321

**2 Support and Service ..... 323**

**Index ..... 324**

# List of figures

Fig. 1 Limits the pre-output to the maximum look-ahead range (default 10 blocks). ... 58
Fig. 2 Distance-related limiting of pre-output to maximum look-ahead range. ... 60
Fig. 3 Synchronisation type MOS (without synchronisation) ... 68
Fig. 4 Synchronisation type MVS_SVS ... 69
Fig. 5 Synchronisation type MVS_SNS ... 70
Fig. 6 Synchronisation type MNS_SNS ... 71
Fig. 7 Synchronisation type MNE_SNS (only for edge banding option) ... 72
Fig. 8 Synchronisation type MVS_SLM ... 73
Fig. 9 Synchronisation type MVS_SLP ... 74
Fig. 10 Without synchronisation MOS_TS ... 75
Fig. 11 Synchronisation types MET_SVS, MEP_SVS ... 76
Fig. 12 Planned output time with MET_SVS ... 77
Fig. 13 Synchronisation types MET_MOS, MEP_MOS ... 79
Fig. 14 Pre-output and acknowledgement of an M function with microjoints ... 81
Fig. 15 Activation of residual path/time calculation with M/H pre-output ... 91
Fig. 16 After output of the M function, motion is continued up to the measurement block (default) ... 100
Fig. 17 After output of the M function, reject the residual measurement block ... 101
Fig. 18 Suppress motion path check with edge banding ... 102
Fig. 19 Switch over measurement logic to a programmed measurement signal at edge banding ... 103
Fig. 20 Switch over to contour-related residual path for edge banding ... 104
Fig. 21 Output M function after measuring signal and remove residual path (P-CHAN-00435 = 0) ... 106
Fig. 22 Output M function directly after measuring signal (P-CHAN-00435 = 1) ... 107
Fig. 23 Motion path limiting during measurement run ... 110
Fig. 24 Zone of permissibly programmed centre points ... 117
Fig. 25 Restore axis configuration after RESET depending on influencing parameters ... 120
Fig. 26 Feed profile for override_weight_acc = 1 and override change ... 124
Fig. 27 Examples of rapid traverse override modes ... 126
Fig. 28 Definition of effector coordinate system (ECS) via the X axis ... 131
Fig. 29 Definition of effector coordinate system (ECS) via the Y axis ... 132
Fig. 30 Example of axis configuration with two axis groups. ... 138
Fig. 31 Example of channel-specific spindle configuration ... 145
Fig. 32 Difference between spindle control by spindle interpolator or PLC ... 146
Fig. 33 Example of a machine structure for synchronous operation with spindles ... 155
Fig. 34 Configuration example of a mapping of the classic CA kinematics according to Kinematic ID 9 (Y, X, Z, C, A) onto the configuration of Universal Kinematics ... 176
Fig. 35 Selecting the orientation resolution ... 185
Fig. 36 const_ramp_time = 0 ... 195
Fig. 37 const_ramp_time = 1 ... 195
Fig. 38 Timing example of speed limit look ahead ... 200
Fig. 39 Timing diagram without override weighting (f_override_weight_v_limit = 0) ... 202
Fig. 40 Timing diagram with override weighting (f_override_weight_v_limit = 1) ... 202
Fig. 41 Edge between two contour elements ... 249
Fig. 42 Timing diagram of s1, s3 signals ... 249
Fig. 43 Timing diagram of s1..s3 signals when P-CHAN-00300 = 1 ... 252

Fig. 44 Timing diagram of s1..s3 signals when P-CHAN-00300 = 2 ...................................................... 253
Fig. 45 Example: Pocket milling cycle with polynomial contouring ........................................................ 266
Fig. 46 Feed along the contour with opt_contour_mode=0:.................................................................. 267
Fig. 47 Feed along the contour with opt_contour_mode=1.................................................................... 268
Fig. 48 Scheme of forward/backward control......................................................................................... 270
Fig. 49 Program execution with offset overlapping ................................................................................. 271
Fig. 50 Movement with forward/backward control with offsets............................................................... 271
Fig. 51 Display CS for each single path (display_global = 0)................................................................. 297
Fig. 52 Display path 2 in CS of path 1 (display_global = 1) ................................................................... 298
Fig. 53 CS of path 2 sets on basic CS of path 1 on (2nd_path_on_actual_1st_path = 0) ...................... 299
Fig. 54 CS of path 2 refers to CS of path 1 (2nd_path_on_actual_1st_path = 1).................................. 300
Fig. 55 Offsets of tracked axes in Cartesian coordinate systems ........................................................ 301
Fig. 56 Intersection point calculation with the coordinate system planes............................................... 302

# Overview of all channel parameters

**The channel parameter overview is sorted into a 4-column table.**

- Column 1 contains the unambiguous identifier of the channel parameter called the “ID” which consists of the prefix “P-CHAN” and a unique 5-digit number, e.g.
  P-CHAN-00001.
- Column 2 represents the data structure which defines the parameters,
  e.g. prog_start.slope.
  The structure is a categorisation aid and is described in the following section. If an entry is missing in ‘structure’, this is not an error. The parameter in column 3 is then only valid on its own.
- Column 3 contains the “parameter” with its exact description, e.g.
  acceleration. The important thing is that “structure”+”parameter” always belong together and must therefore be configured in exactly the same way in the channel parameter list,
  e.g. prog_start.slope.acceleration.
- Column 4 contains the “functionality” in a summarised term/short description,
  e.g. Slope setting

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00001 [▶ 194] | prog_start.slope. | acceleration | Default effect of acceleration weighting at program start |
| P-CHAN-00002 [▶ 127] | vector. | acceleration | Path acceleration limit |
| P-CHAN-00003 [▶ 139] | gruppe[i]. | achs_anzahl | Number of axes in each axis group |
| P-CHAN-00004 [▶ 153] | spindel[i]. | autom_range | Automatic range selection for spindle gear change |
| P-CHAN-00005 [▶ 139] | gruppe[i]. | bezeichnung | Name of the axis group in the NC channel |
| P-CHAN-00006 [▶ 139] | gruppe[i].achse[j]. | bezeichnung | Name of an axis in the NC channel |
| P-CHAN-00007 [▶ 145] | spindel[i]. | bezeichnung | Name of a spindle in the NC channel |
| P-CHAN-00008 [▶ 142] | | cax_face_id | Machine ID with C axis face machining |
| P-CHAN-00009 [▶ 198] | | corr_v_trans_jerk | Reduction of tangential transition velocity between circles |
| P-CHAN-00010 [▶ 142] | | default_ax_name_of_spindle | Spindle name within a path compound |
| P-CHAN-00011 [▶ 140] | gruppe[i].achse[j]. | default_feed_axis | Assign an axis to the default feed group |
| P-CHAN-00012 [▶ 201] | speed_limit_look_ahead. | dist_from_corner | Distance from corner for speed limit look ahead |
| P-CHAN-00013 [▶ 201] | speed_limit_look_ahead. | dist_to_corner | Distance to corner for speed limit look ahead |
| P-CHAN-00014 [▶ 94] | | einrechnen_mit_t | Implicit execution of D word with T |
| P-CHAN-00015 [▶ 115] | | einzelschrittmodus | Define single-step operating mode |
| P-CHAN-00016 [▶ 95] | | ext_wzv_vorhanden | Tool data is requested from an external tool management |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00017 [▶ 200] | speed_limit_look_ahead. | enable | Enable/disable speed limit look ahead |
| P-CHAN-00018 [▶ 201] | speed_limit_look_ahead. | time | Unit to interpret the SLD signal for speed look ahead |
| P-CHAN-00019 [▶ 163] | syn_chk. | errors_total | Number of errors in an NC program on syntax check |
| P-CHAN-00020 [▶ 163] | syn_chk. | errors_per_block | Number of errors per row on syntax check |
| P-CHAN-00021 [▶ 161] | | create_cont_mask_warnings | Suppress warnings during contour masking (tool radius compensation) |
| P-CHAN-00022 [▶ 159] | aep. | g_gruppe[i] | Enable logging of G functions |
| P-CHAN-00023 [▶ 138] | | grp_anzahl | Number of axis groups in the NC channel |
| P-CHAN-00024 [▶ 158] | aep. | output_mode | Enable change logging |
| P-CHAN-00025 [▶ 85] | | h_default_outp_ax_name[i] | Axis-specific H functions |
| P-CHAN-00026 [▶ 86] | | h_prozess_zeit[i] | Timeout / process times of H functions for machining time calculation |
| P-CHAN-00027 [▶ 83] | | h_synch[i] | Synchronisation type of H functions |
| P-CHAN-00028 [▶ 163] | syn_chk. | interactive | Specify the operation mode on syntax check |
| P-CHAN-00029 [▶ 99] | | kasto_multi_block | Block global edge banding |
| P-CHAN-00030 [▶ 99] | | kasto_residual_path | Residual path on arrival of measurement signal at edge banding |
| P-CHAN-00031 [▶ 130] | | kind_of_2nd_ecs_ax | Select the second axis of effector coordinate systems (ECS) |
| P-CHAN-00032 [▶ 165] | | kinematik_id | Select kinematic default transformation (kinematic type) |
| P-CHAN-00033 [▶ 196] | prog_start. | late_sync_ready | Default setting for 'Late synchronization at program end' at program start |
| P-CHAN-00034 [▶ 93] | | lin_aufloes | Resolution of linear axes |
| P-CHAN-00035 [▶ 140] | gruppe[i].achse[j]. | log_achs_nr | Logical number of an axis in the NC channel |
| P-CHAN-00036 [▶ 145] | spindel[i]. | log_achs_nr | Logical axis number of a spindle in NC channel |
| P-CHAN-00037 [▶ 156] | synchro_data.koppel_gruppe[i]. paar[j]. | log_achs_nr_master | Logical axis number of the master axis (synchronous operation) |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00038 [▶ 156] | synchro_data.koppel_gruppe[i]. paar[j]. | log_achs_nr_slave | Logical axis number of the slave axis (synchronous operation) |
| P-CHAN-00039 [▶ 82] | | m_default_outp_ax_name[i] | Axis-specific M functions |
| P-CHAN-00040 [▶ 82] | | m_prozess_zeit[i] | Timeout / process times of M functions for machining time calculation |
| P-CHAN-00041 [▶ 65] | | m_synch[i] | Synchronisation type of M functions |
| P-CHAN-00042 [▶ 152] | spindel[i]. | m19_prozess_zeit | Timeout and process time of M19 to calculate machining time |
| P-CHAN-00043 [▶ 149] | spindel[i]. | m19_synch | Synchronisation types for M19 |
| P-CHAN-00044 [▶ 151] | spindel[i]. | m3_prozess_zeit | Timeout and process time of M03 for calculation of machining time |
| P-CHAN-00045 [▶ 148] | spindel[i]. | m3_synch | Synchronisation method for M03 |
| P-CHAN-00046 [▶ 151] | spindel[i]. | m4_prozess_zeit | Timeout and process time of M04 for calculation of machining time |
| P-CHAN-00047 [▶ 148] | spindel[i]. | m4_synch | Synchronisation types for M04 |
| P-CHAN-00048 [▶ 152] | spindel[i]. | m5_prozess_zeit | Timeout and process time of M05 for calculation of machining time |
| P-CHAN-00049 [▶ 149] | spindel[i]. | m5_synch | Synchronisation types for M05 |
| P-CHAN-00050 [▶ 130] | | mach_plane_of_2nd_ecs_ax | Select the plane for the second axis of the effector coordinate systems (ECS) |
| P-CHAN-00051 [▶ 143] | | main_spindle_ax_nr | Logical axis number of the main spindle in the NC channel |
| P-CHAN-00052 [▶ 143] | | main_spindle_gear_change | Enable mechanical gear change of main spindle |
| P-CHAN-00053 [▶ 143] | | main_spindle_name | Designation of the main spindle in the NC channel |
| P-CHAN-00054 [▶ 92] | | mass_einh | Dimensional unit for translatory axes in NC program |
| P-CHAN-00055 [▶ 154] | range_table[j]. | max_speed | Maximum spindle speed of a speed range (spindle gear change) |
| P-CHAN-00056 [▶ 122] | | max_vb_override | Limiting maximum channel override |
| P-CHAN-00057 [▶ 107] | | messtyp | Predefine measurement type |
| P-CHAN-00058 [▶ 154] | range_table[j]. | min_speed | Minimum spindle speed of a speed range (spindle gear change) |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00059 [▶ 117] | | mittelpkt_diff | Permissible absolute centre point offset during circle programming |
| P-CHAN-00060 [▶ 118] | | mittelpkt_faktor | Permissible relative centre point offset during circle programming |
| P-CHAN-00061 [▶ 157] | synchro_data.koppel _gruppe[i]. paar[j]. | mode | Mode of the coupled axis pair (synchronous operation) |
| P-CHAN-00062 [▶ 160] | makro_def[i]. | nc_code | Specify macro content |
| P-CHAN-00063 [▶ 191] | prog_start.g_gruppe[i]. | nr | Default G function numbers at program start |
| P-CHAN-00064 [▶ 193] | prog_start.m_gruppe[i]. | nr | Default M function numbers at program start |
| P-CHAN-00065 [▶ 122] | | override_delay | Override scanning |
| P-CHAN-00066 [▶ 123] | | override_weight_prog_feed | Influence of override on feed |
| P-CHAN-00067 [▶ 115] | | p_param_haltend | Global program effectiveness of P parameters |
| P-CHAN-00068 [▶ 158] | aep. | p_parameter | Enable logging of P parameters |
| P-CHAN-00069 [▶ 146] | spindel[i]. | plc_control | Spindle control by PLC via channel specific interface |
| P-CHAN-00070 [▶ 81] | | m_pre_outp[i] | Path or time-related pre-output of M functions |
| P-CHAN-00071 [▶ 194] | prog_start.slope. | profile | Default acceleration profile at program start |
| P-CHAN-00072 [▶ 87] | tool. (old syntax: werkzeug.) | prozess_zeit | Timeout / process time of tool request for machining time calculation |
| P-CHAN-00073 [▶ 194] | prog_start.slope. | ramp_time | Default effect of ramp time weighting at program start |
| P-CHAN-00074 [▶ 153] | spindel[i]. | range_way | Direction of range selection for spindle gear change |
| P-CHAN-00075 [▶ 121] | | reset_no_axis_to_axv | Axis release after reset |
| P-CHAN-00076 [▶ 97] | | tool_life_to_wzv | Send tool life data to external tool management |
| P-CHAN-00077 [▶ 88] | | rpf_prozess_zeit | Process time of homing for machining time calculation |
| P-CHAN-00078 [▶ 93] | | rund_aufloes | Resolution of rotary axes |
| P-CHAN-00079 [▶ 92] | | rund_mass_einh | Dimensional unit for rotary axes in NC program |
| P-CHAN-00080 [▶ 151] | spindel[i]. | s_prozess_zeit | Timeout and process time of the spindle function S for the machining time calculation |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00081 [▶ 147] | spindel[i]. | s_synch | Synchronisation type of the spindle S function |
| P-CHAN-00082 [▶ 142] | | spdl_anzahl | Number of configured spindles in NC channel |
| P-CHAN-00083 [▶ 94] | | spind_aufloes | Resolution of spindles |
| P-CHAN-00085 [▶ 160] | makro_def[i]. | symbol | Specification of symbol strings |
| P-CHAN-00086 [▶ 86] | tool. (old syntax: werkzeug.) | synch | Synchronisation type of tool request (T function) |
| P-CHAN-00087 [▶ 95] | | t_info_to_wzv | Information of the external tool management in case of a T function |
| P-CHAN-00088 [▶ 158] | aep. | v_eigendef | Enable logging of self-defined variables |
| P-CHAN-00089 [▶ 200] | speed_limit_look _ahead. | v_limit | Weighting of speed limit for speed limit look ahead |
| P-CHAN-00090 [▶ 129] | vector. | velocity | Path velocity limit |
| P-CHAN-00091 [▶ 115] | | versch_im_durchm | Set the offset scaling for diameter programming |
| P-CHAN-00092 [▶ 161] | | wrk_im_kanal_vorhanden | NC channel configuration includes a module for tool radius compensation (TRC) |
| P-CHAN-00093 [▶ 160] | aep. | wz_daten | Enable the logging of tool data |
| P-CHAN-00094 [▶ 166] | kinematik[i]. | param[j] | Define kinematic parameters |
| P-CHAN-00095 [▶ 133] | | log_number_tracking_axis | Define the tracking axis (C axis tracking) |
| P-CHAN-00096 [▶ 135] | | feed_to_weakest_axis | Define the weakest axis as the default feed axis |
| P-CHAN-00097 [▶ 136] | | use_drive_curr_limit | Valid deceleration ramp at FEEDHOLD |
| P-CHAN-00098 [▶ 136] | | spindle_m_fct_free | Switch over the meaning of M functions M3/M4/M5/M19 |
| P-CHAN-00099 [▶ 195] | prog_start. | feedrate | Default feed rate at program start |
| P-CHAN-00100 [▶ 137] | | move_tool_offsets_directly | Time of effectiveness of tool compensation data |
| P-CHAN-00101 [▶ 133] | | auto_align_tracking_axis | Automatic orientation of a tracking axis |
| P-CHAN-00102 [▶ 127] | | plc_command_rapid_feed | Operating principle of PLC feed |
| P-CHAN-00103 [▶ 95] | | d_clear_to_wzv | Save and delete tool data in an external tool management |
| P-CHAN-00104 [▶ 157] | synchro_data. | restore_coupling_after_reset | Restore coupling after reset (synchronous operation) |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00105 [▶ 157] | synchro_data. | preserve_coupling_after_prog_end | Restore coupling after program end (synchronous operation) |
| P-CHAN-00106 [▶ 97] | | t_with_implicit_flush | Releasing motion blocks during T-selection |
| P-CHAN-00107 [▶ 84] | | h_pre_outp[i] | Path or time-related pre-output of H functions |
| P-CHAN-00108 [▶ 196] | prog_start. | feedrate_factor | Default feed rate unit at program start |
| P-CHAN-00109 [▶ 133] | | fast_tracking_transition | Select tracking mode without stopping |
| P-CHAN-00110 [▶ 198] | | check_jerk_on_poly_path | Including jerk in the polynomial |
| P-CHAN-00111 [▶ 126] | | time_override_weight_dwell_time | Influence of time override on dwell time |
| P-CHAN-00112 [▶ 181] | | ori_rotation_angle | Mode of orientation angle programming for kinematic transformations |
| P-CHAN-00113 [▶ 203] | | path_reject_std_manual_mode | Reject manual mode path at program end |
| P-CHAN-00114 [▶ 203] | | rel_offset_limits_std_manual_mode | Relative manual mode offset limits with G200 |
| P-CHAN-00115 [▶ 121] | | release_non_channel_axis | Release of non-channel axes after reset |
| P-CHAN-00116 [▶ 205] | | multi_dimension_in_block | Exclusivity of measuring system programming |
| P-CHAN-00117 [▶ 199] | | mode_trans_jerk | Jerk limiting mode at block transition of any blocks |
| P-CHAN-00118 [▶ 206] | | m6_prog_file | Subroutine call with M6 |
| P-CHAN-00119 [▶ 208] | | start_init_prog_file | Name of implicit subroutine call at program start |
| P-CHAN-00120 [▶ 211] | | soft_limit_tolerance | Software limit switch (SLS) monitoring with tolerance |
| P-CHAN-00121 [▶ 283] | | simu_output_wcs | Display format during machining simulation |
| P-CHAN-00122 [▶ 216] | hsc.bspline. | path_deviation | Maximum deviation of the B spline from the programmed path contour |
| P-CHAN-00123 [▶ 216] | hsc.bspline. | track_deviation | Deviation of the tracked axis (B Spline) |
| P-CHAN-00124 [▶ 216] | hsc.bspline. | max_path_length | Path length of relevant blocks (B Spline) |
| P-CHAN-00125 [▶ 216] | hsc.bspline. | max_angle | Contour bend angle for linear block transitions (B Spline) |
| P-CHAN-00126 [▶ 217] | hsc.bspline. | merge_retry | Cancellation criteria for merging blocks (B Spline) |
| P-CHAN-00127 [▶ 217] | hsc.bspline. | merge_window | Start criteria for merging blocks (B Spline) |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00128 [▶ 217] | hsc.bspline. | auto_off_g00 | Deselect spline with G00 blocks (B Spline) |
| P-CHAN-00129 [▶ 217] | hsc.bspline. | auto_off_path | Response to maximum path deviation (B Spline) |
| P-CHAN-00130 [▶ 218] | hsc.bspline. | auto_off_track | Response to maximum tracked axis deviation (B Spline) |
| P-CHAN-00131 [▶ 218] | hsc.bspline. | limit_corner_dist | Limit the corner distance (B Spline) |
| P-CHAN-00132 [▶ 220] | hsc.gen. | linear_segmentation | Linear block segmentation (HSC) |
| P-CHAN-00133 [▶ 220] | hsc.gen. | linear_center_point | Number of blocks for linear segmentation (HSC) |
| P-CHAN-00134 [▶ 221] | hsc.gen. | linear_continuous_split | Segmentation specification (HSC) |
| P-CHAN-00135 [▶ 221] | hsc.gen. | linear_split_length | Segment length with linear blocks (HSC) |
| P-CHAN-00136 [▶ 221] | hsc.gen. | circular_secant_error | Basis of segmentation of circular blocks (HSC) |
| P-CHAN-00137 [▶ 221] | hsc.gen. | circular_segmentation | Segmentation of circular blocks (HSC) |
| P-CHAN-00138 [▶ 222] | hsc.gen. | circular_param | Contour error in the case of segmentation of circular blocks (HSC) |
| P-CHAN-00139 [▶ 222] | hsc.gen. | filter_deviation | Handling short blocks (HSC) |
| P-CHAN-00140 [▶ 222] | hsc.gen. | min_segment_length | Minimum segment length for block segmentation (HSC) |
| P-CHAN-00141 [▶ 223] | hsc.gen. | jerk_monitoring_mode | Jerk limiting algorithm (HSC) |
| P-CHAN-00142 [▶ 223] | hsc.gen. | jerk_weighting_5ax | Factor for permissible jerk in 5-axis mode (HSC) |
| P-CHAN-00143 [▶ 223] | hsc.gen. | jerk_weighting | Factor for permissible jerk in 2.5D mode (HSC) |
| P-CHAN-00144 [▶ 134] | | suppress_cs_tracking_offset | Suppress a rotary CS offset in tracking mode |
| P-CHAN-00145 [▶ 182] | | kin_trafo_display | Activation of TCP display data |
| P-CHAN-00146 [▶ 214] | | parameter_change_during_execution | Adopt configuration lists when NC program is active |
| P-CHAN-00147 [▶ 212] | | soft_limit_warning_axes | Error reaction for software limit switch monitoring on command value side |
| P-CHAN-00148 [▶ 206] | | multi_axes_in_block | Exclusivity of axis programming |
| P-CHAN-00149 [▶ 100] | | kasto_reject_rest_block | Reject residual block for edge banding on receipt of the measuring signal and continue with next NC block |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00150 [▶ 214] | | remain_block_number_sub_prog_call | Adoption of block number 'N..' when subroutine is called |
| P-CHAN-00151 [▶ 188] | | auto_enable_kin_trafo | Implicit selection of kinematic transformation at program start |
| P-CHAN-00152 [▶ 188] | | auto_disable_kin_trafo | Implicit deselection of kinematic transformation at program end |
| P-CHAN-00153 [▶ 101] | | kasto_remain_active | Modal active measurement at edge banding |
| P-CHAN-00154 [▶ 224] | | ax_exchange_with_implicit_flush | Axis exchange with implicit channel synchronisation (FLUSH CONTINUE) |
| P-CHAN-00155 [▶ 201] | speed_limit_look_ahead. | override_weight_v_limit | Weighting the speed limit via override for speed limit look ahead function |
| P-CHAN-00156 [▶ 88] | tool. | minimum_length | Minimum permissible tool length |
| P-CHAN-00157 [▶ 88] | tool. | minimum_radius | Minimum permissible tool radius |
| P-CHAN-00158 [▶ 225] | | streaming_prog_file | Program name for automatic streaming |
| P-CHAN-00159 [▶ 226] | | modal_i_j_k_for_circle | Modal effect of I/J/K for circle programming |
| P-CHAN-00160 [▶ 226] | | g80_prog_file | Implicit call with G80 |
| P-CHAN-00161 [▶ 227] | | g81_prog_file | Implicit call with G81 |
| P-CHAN-00162 [▶ 227] | | g82_prog_file | Implicit call with G82 |
| P-CHAN-00163 [▶ 228] | | g83_prog_file | Implicit call with G83 |
| P-CHAN-00164 [▶ 229] | | g84_prog_file | Implicit call with G84 |
| P-CHAN-00165 [▶ 229] | | g85_prog_file | Implicit call with G85 |
| P-CHAN-00166 [▶ 230] | | g86_prog_file | Implicit call with G86 |
| P-CHAN-00167 [▶ 231] | | g87_prog_file | Implicit call with G87 |
| P-CHAN-00168 [▶ 231] | | g88_prog_file | Implicit call with G88 |
| P-CHAN-00169 [▶ 232] | | g89_prog_file | Implicit call with G89 |
| P-CHAN-00170 [▶ 123] | | override_weight_acc | Influence of override on acceleration |
| P-CHAN-00171 [▶ 118] | | max_radius_diff_circle | Maximum permissible absolute radius difference during circle programming |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00172 [▶ 118] | | max_proz_radius_diff_circle | Maximum permissible percentage radius difference during circle programming |
| P-CHAN-00173 [▶ 233] | | suppress_ax_group_in_pos_check | Active testing whether axis group is 'in position' |
| P-CHAN-00174 [▶ 282] | | channel_name | Define a channel name for display on the HLI |
| P-CHAN-00175 [▶ 282] | | channel_type | Definition of a channel type for display on the HLI |
| P-CHAN-00176 [▶ 108] | | meas_error_no_signal | Error reaction with measurement type 1 |
| P-CHAN-00177 [▶ 260] | ori. | mode | Type of orientation representation |
| P-CHAN-00178 [▶ 260] | ori. | fixed_axis_index | Index of fixed rotary axis to program Euler angles |
| P-CHAN-00179 [▶ 120] | | ax_default_config_after_reset | Restore default axis configuration in the channel after reset |
| P-CHAN-00180 [▶ 282] | | suppress_prg_display_level | Suppress display of file name/file offset |
| P-CHAN-00181 [▶ 124] | | g00_override_mode | Define and activate rapid traverse override mode |
| P-CHAN-00182 [▶ 304] | | plcopen_std_unit | Units used in PLC Open functions |
| P-CHAN-00183 [▶ 235] | | simu_ignore_internal_stop_cond | Ignore internal stop conditions with rapid contour visualisation |
| P-CHAN-00184 [▶ 184] | | kin_trafo_display_curr_pos | Switchover of TCP display data between command and actual positions |
| P-CHAN-00185 [▶ 134] | | tracking_axis_rot_wc | Define the position of the tracking axis in the workpiece |
| P-CHAN-00186 [▶ 203] | | override_v_handwheel | Override with manual mode |
| P-CHAN-00187 [▶ 233] | | g800_prog_file[i] | Additional implicit calls with G800-G819 |
| P-CHAN-00188 [▶ 184] | tool_ori_cs. | axis | Define the main axis of the current coordinate system to which the tool is to be aligned |
| P-CHAN-00189 [▶ 184] | tool_ori_cs. | mode | Selecting the orientation resolution |
| P-CHAN-00190 [▶ 236] | | dynamic_weighting_active | Activate path-dependent dynamic weighting |
| P-CHAN-00191 [▶ 236] | dynamic_weighting[i]. | path_limit | Motion path limit (path-dependent dynamic weighting) |
| P-CHAN-00192 [▶ 237] | dynamic_weighting[i]. | velocity_fact | Weighting factor for speed (path-dependent dynamic weighting) |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00193 [▸ 237] | dynamic_weighting[i]. | acceleration_fact | Weighting factor for acceleration (path-dependent dynamic weighting) |
| P-CHAN-00194 [▸ 237] | dynamic_weighting[i]. | ramp_time_fact | Weighting factor for ramp time (path-dependent dynamic weighting) |
| P-CHAN-00195 [▸ 242] | man_mode.vector_limit. | v_max_pos | Command speed of a linear axis in manual mode |
| P-CHAN-00196 [▸ 242] | man_mode.vector_limit. | a_max_pos | Command acceleration of a linear axis in manual mode |
| P-CHAN-00197 [▸ 242] | man_mode.vector_limit. | tr_pos | Ramp time of a linear axis in manual mode |
| P-CHAN-00198 [▸ 243] | man_mode.vector_limit. | v_max_ori | Command speed of an orientation axis in manual mode |
| P-CHAN-00199 [▸ 243] | man_mode.vector_limit. | a_max_ori | Command acceleration of an orientation axis in manual mode |
| P-CHAN-00200 [▸ 243] | man_mode.vector_limit. | tr_ori | Ramp time of an orientation axis in manual operation mode |
| P-CHAN-00208 [▸ 128] | vector. | deceleration | Path deceleration limit |
| P-CHAN-00209 [▸ 88] | | m_h_pre_outp_time_calc_mode | Calculation model for M/H pre-output time |
| P-CHAN-00210 [▸ 244] | | cycle_changes_modal | Modal effect of changes in the cycle |
| P-CHAN-00211 [▸ 244] | | suppress_cycle_logging | Suppress NC blocks from cycles and M6 in trace data and display |
| P-CHAN-00212 [▸ 90] | | m_h_pre_outp_calc_value_to_go | Activate residual path/time calculation with M/H code look ahead |
| P-CHAN-00213 [▸ 185] | | gap_in_trafo_axis_sequence | Permit gapped axis configuration when a transformation is selected |
| P-CHAN-00214 [▸ 108] | | meas_deceleration_mode | Active delay with measuring signal |
| P-CHAN-00215 [▸ 134] | | consider_rot_tracking_offset | Including a contour rotation in the tracking axis |
| P-CHAN-00216 [▸ 254] | | max_nc_blocks_ahead | Select decoder block ahead limiting via number of blocks |
| P-CHAN-00217 [▸ 223] | hsc.gen. | on_delay_output_one_block | Delayed HSC selection for contour spline processing |
| P-CHAN-00218 [▸ 235] | | independent_axis_error_stop | Mode of deceleration response in case of an axis error |
| P-CHAN-00219 [▸ 161] | | implicit_contour_masking | Activate implicit contour masking (tool radius compensation) |
| P-CHAN-00220 [▸ 250] | edge_machining. | enable | Enable / disable the edge machining function |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00221 [▶ 250] | edge_machining. | angle_limit | Critical bend angle (edge machining function) |
| P-CHAN-00222 [▶ 250] | edge_machining. | pre_dist | Distance before the edge (edge machining function) |
| P-CHAN-00223 [▶ 250] | edge_machining. | pre_feed | Feed before edge (edge machining function) |
| P-CHAN-00224 [▶ 251] | edge_machining. | wait_time | Waiting time in edge (edge machining function) |
| P-CHAN-00225 [▶ 251] | edge_machining. | post_dist | Distance after the edge (edge machining function) |
| P-CHAN-00226 [▶ 251] | edge_machining. | post_feed | Feed rate after the edge (edge machining function) |
| P-CHAN-00227 [▶ 102] | | kasto_suppress_path_check | Suppress motion path check with edge banding |
| P-CHAN-00228 [▶ 187] | | permitted_trafo_deviation_limit | Permissible calculation tolerance with forward / backward transformation |
| P-CHAN-00230 [▶ 239] | | curve_dynamic_weighting_active | Activate radius-dependent dynamic weighting |
| P-CHAN-00231 [▶ 239] | curve_dynamic _weighting[i]. | radius_limit | Radius limit (radius-dependent dynamic weighting) |
| P-CHAN-00232 [▶ 240] | curve_dynamic _weighting[i]. | velocity_fact | Weighting factor for velocity (radius-dependent dynamic weighting) |
| P-CHAN-00233 [▶ 240] | curve_dynamic _weighting[i]. | acceleration_fact | Weighting factor for acceleration (radius-dependent dynamic weighting) |
| P-CHAN-00234 [▶ 240] | curve_dynamic _weighting[i]. | ramp_time_fact | Weighting factor for ramp time (radius-dependent dynamic weighting) |
| P-CHAN-00235 [▶ 257] | | exact_stop_after_g00 | Exact stop after every rapid traverse |
| P-CHAN-00239 [▶ 218] | hsc.bspline. | auto_contour_mode | Automatic insertion of polynomials on internal B spline deselection |
| P-CHAN-00240 [▶ 219] | hsc.bspline. | fgroup | Define feed group for B spline |
| P-CHAN-00241 [▶ 219] | hsc.bspline. | no_hsc_for_deviation_zero | Deselect HSC if a deviation is equal to zero (B Spline) |
| P-CHAN-00243 [▶ 98] | | tool_life_to_wzv_suppress_zero | Suppress output of tool life parameters 0 |
| P-CHAN-00244 [▶ 258] | | lift_min_dist | Minimum path length for lift movements |
| P-CHAN-00245 [▶ 199] | | trans_limit_with_curvature | Additional consideration of curvature at block transition between linear blocks |
| P-CHAN-00246 [▶ 254] | | max_motion_blocks_ahead | Selecting decoder ahead limiting via number of NC motion blocks |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00247 [▶ 186] |  | ori_wcs | Mapping of the tool orientation in the active machining coordinate system |
| P-CHAN-00249 [▶ 265] | tube_profile. | techno_nr_rnd_on | Technological functions to signal rounding transitions on engaging with tube profile machining |
| P-CHAN-00250 [▶ 264] | tube_profile. | techno_nr_rnd_off | Technological functions to display rounding transitions when exiting for tube profile machining |
| P-CHAN-00251 [▶ 264] | tube_profile. | techno_type | Type definition of technological functions for tube profile machining |
| P-CHAN-00252 [▶ 210] |  | final_prog_file | Name of implicitly called subroutine at program end |
| P-CHAN-00253 [▶ 262] |  | use_alias_name_in_ax_list | Alternative programming of axis-specific movements in robotics |
| P-CHAN-00254 [▶ 263] |  | output_block_length_zero | Handling polynomial blocks of zero length |
| P-CHAN-00255 [▶ 98] |  | tool_life_capture_all_movements | Tool life data recording for rapid traverse motions |
| P-CHAN-00256 [▶ 283] |  | display_diameter_pos | Diameter display for turn machining |
| P-CHAN-00257 [▶ 103] |  | kasto_prog_meas_trigger | Switch over measurement logic to a programmed measurement signal at edge banding |
| P-CHAN-00258 [▶ 104] |  | kasto_relate_to_prog_contour | Switch over to contour-related residual path for edge banding |
| P-CHAN-00259 [▶ 266] |  | opt_contour_mode | Optimised polynomial smoothing for transitions in circular blocks |
| P-CHAN-00260 [▶ 208] |  | start_init_prog_file_mode | Mode of implicit subroutine call at program start |
| P-CHAN-00261 [▶ 281] |  | multi_path_configuration | Enable 2-path programming |
| P-CHAN-00262 [▶ 168] | kin_step[i].trafo[j].* or trafo[j].* | id | Define kinematic ID for multi-step transformations |
| P-CHAN-00263 [▶ 169] | kin_step[i].trafo[j].* or trafo[j].* | param[k] | Define kinematic parameters for multi-step transformations |
| P-CHAN-00264 [▶ 167] |  | default_id_of_kin_step[i] | Select default kinematic types for multi-step transformations |
| P-CHAN-00265 [▶ 135] |  | suppress_tracking_vec_limit | Suppress vector dynamic limitation in aligning blocks in tracking mode |
| P-CHAN-00266 [▶ 108] |  | meas_fixed_stop_no_error | Error response with measurement type 7 |
| P-CHAN-00267 [▶ 268] |  | hli_input_disable_condition | Configuring interpolator input disable |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00268 [▶ 110] | | meas_soft_limit_move_path | Automatic motion path limiting |
| P-CHAN-00269 [▶ 255] | | max_time_ahead | Selecting time-based decoder block ahead limiting |
| P-CHAN-00270 [▶ 255] | | dec_max_ahead_protected | Block ahead limiting in protected mode |
| P-CHAN-00271 [▶ 97] | flush. | d_cmd | Implicit '#FLUSH WAIT' on selection of D word |
| P-CHAN-00273 [▶ 255] | | disable_chamfers_roundings | Select/deselect the chamfers and roundings function |
| P-CHAN-00274 [▶ 91] | | m_h_pre_outp_nbr_block | Number of NC blocks at M/H code look ahead |
| P-CHAN-00275 [▶ 272] | forward_backward. | with_offset | Backward motion with external position offsets |
| P-CHAN-00276 [▶ 273] | forward_backward. | disable_m00_backward | Programmed M00 stop during backward motions |
| P-CHAN-00277 [▶ 273] | forward_backward. | disable_m00_2nd_forward | Programmed M00 stop during next forward motions |
| P-CHAN-00278 [▶ 273] | forward_backward. | disable_m01_backward | Programmed M01 stop during backward movement |
| P-CHAN-00279 [▶ 273] | forward_backward. | disable_m01_2nd_forward | Programmed M01 stop during next forward movement |
| P-CHAN-00280 [▶ 277] | customer. | val[i] | Free user-defined values |
| P-CHAN-00281 [▶ 135] | | filter_tracking_axis | PCS smoothing function for tracking axis |
| P-CHAN-00282 [▶ 304] | | drive_cmd_use_physical_axis_number | Use of axis numbers of master channel in clone channel |
| P-CHAN-00283 [▶ 277] | | encryption_extension[i] | Definition of file extensions for encryption of NC programs |
| P-CHAN-00284 [▶ 162] | | trc_circular_kerf_masking | Extended kerf detection for circular blocks (TRC) |
| P-CHAN-00285 [▶ 173] | kinematik[91].* or<br>kin_step[i].trafo[j].* or<br>trafo[j].* | orientation[k] | Zero orientation of the tool (Universal Kinematics) |
| P-CHAN-00286 [▶ 173] | kinematik[91].* or<br>kin_step[i].trafo[j].* or<br>trafo[j].* | position[k] | Zero position of the tool (Universal Kinematics) |
| P-CHAN-00287 [▶ 173] | kinematik[91].* or<br>kin_step[i].trafo[j].* or<br>trafo[j].* | rtcp | Angle transformation (Universal Kinematics) |
| P-CHAN-00288 [▶ 174] | kinematik[91].* or<br>kin_step[i].trafo[j].* or<br>trafo[j].* | programming_mode | Programming mode (Universal Kinematics) |
| P-CHAN-00289 [▶ 174] | kinematik[91].* or<br>kin_step[i].trafo[j].* or<br>trafo[j].* | number_of_axes | Number of axes (Universal Kinematics) |
| P-CHAN-00290 [▶ 175] | kinematik[91].* or<br>kin_step[i].trafo[j].* or<br>trafo[j].* | chain[k] | Axis sequence (Universal Kinematics) |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00291 [▶ 175] | kinematik[91].axis[k].* or kin_step[i].trafo[j].axis[k].* or trafo[j].axis[k].* | type | Axis type (Universal Kinematics) |
| P-CHAN-00292 [▶ 175] | kinematik[91].axis[k].* or kin_step[i].trafo[j].axis[k].* or trafo[j].axis[k].* | orientation[l] | Axis orientation (Universal Kinematics) |
| P-CHAN-00293 [▶ 176] | kinematik[91].axis[k].* or kin_step[i].trafo[j].axis[k].* or trafo[j].axis[k].* | point[l] | Interpolation point on the axis (Universal Kinematics) |
| P-CHAN-00294 [▶ 177] | kinematik[91].* or kin_step[i].trafo[j].* or trafo[j].* | linkage_mode | Mode for the transformation between axis values and Cartesian coordinates (Universal Kinematics) |
| P-CHAN-00295 [▶ 178] | kinematik[91].* or kin_step[i].trafo[j].* or trafo[j].* | linkage[k][l] | Transformation between axis values and Cartesian coordinates (Universal Kinematics) |
| P-CHAN-00296 [▶ 110] | | meas_use_std_dynamic | Active dynamic for measurement run |
| P-CHAN-00297 [▶ 278] | | mode_exact_stop_indp_axis | Position synchronisation of independent axes |
| P-CHAN-00298 [▶ 105] | | kasto_warning_no_signal | Output a warning message if probing signal is not detected at edge banding |
| P-CHAN-00299 [▶ 105] | | kasto_warning_residual_path | Output of a warning message if residual motion path is too small at edge banding |
| P-CHAN-00300 [▶ 251] | edge_machining. | [ DISABLE_FEED_ADAPTION ] ] | Switching feed rate adjustment (edge machining function) |
| P-CHAN-00301 [▶ 254] | edge_machining. | mode | Edge detection mode |
| P-CHAN-00305 [▶ 279] | | block_search_restart_mode | Jog of path option after block search for coupled axes |
| P-CHAN-00306 [▶ 280] | | jog_of_path_only | Checking the license in the clone channel |
| P-CHAN-00307 [▶ 313] | | position_lookahead_distance | Time preview: Reserved distance for time calculation |
| P-CHAN-00308 [▶ 275] | forward_backward. | disable_stop_backward | Response at STOP marks during backward movement |
| P-CHAN-00309 [▶ 275] | forward_backward. | disable_stop_1st_forward | Response at STOP marks during forward movement |
| P-CHAN-00310 [▶ 275] | forward_backward. | disable_stop_2nd_forward | Response at STOP marks during repeated forward movement |
| P-CHAN-00311 [▶ 112] | | meas_pcs_value_no_probe_radius | Include ball radius in calculation of PCS measured value |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00312 [▸ 113] | | meas_allow_actuated_probe | Allow already-actuated measuring probes when starting a measurement run |
| P-CHAN-00313 [▸ 113] | | meas_error_no_move_path | Measurement run without programmed motion path |
| P-CHAN-00315 [▸ 92] | | resolution_factor | Scaling factor of axis resolution |
| P-CHAN-00316 [▸ 197] | prog_start. | initialize_on_actual_position | Job manager mode: Request current positions at program start |
| P-CHAN-00317 [▸ 197] | prog_start. | set_default_config | Job manager mode: Initialise working data at program start |
| P-CHAN-00318 [▸ 197] | prog_start. | report_scene_sample | Job manager mode: Enable the interface to log scene data |
| P-CHAN-00319 [▸ 197] | prog_start. | report_run_time_measure | Job manager mode: Enable the interface to log time stamps |
| P-CHAN-00320 [▸ 198] | prog_start. | report_axes_position_sample | Job manager mode: Enable the interface to log axis positions |
| P-CHAN-00321 [▸ 313] | | block_filter_tolerance | Merge short blocks for optimised smoothing |
| P-CHAN-00322 | | t_with_tool_data_provide | Output of tool data to the PLC interface (Tool Data Provider) |
| P-CHAN-00324 [▸ 316] | esa. | time | Default offset time to calculate future states |
| P-CHAN-00325 [▸ 317] | esa. | mode | Precalculation mode |
| P-CHAN-00326 [▸ 114] | | meas_dont_allow_suspend_axis_output | Block switch-over between physical and clone axes during a measurement run |
| P-CHAN-00327 [▸ 114] | | meas_all_axes | Axes, measurement values and offsets in CS |
| P-CHAN-00328 [▸ 283] | | mode_feed_display | Display of active path feed |
| P-CHAN-00329 [▸ 304] | | ipo_start_wait_cycles | Buffer motion blocks to optimise the feed rate profile |
| P-CHAN-00330 [▸ 284] | | display_top_coord_sys | Select the coordinate system for display of axis positions |
| P-CHAN-00331 [▸ 284] | | kin_trafo_enable_cs_coord_display | Display of axis command values and target points in Cartesian coordinate systems |
| P-CHAN-00332 [▸ 302] | | ori_prog | Response of G91 with orientation axes for complete transformation |
| P-CHAN-00338 [▸ 287] | | mdi_log_file | Name of the manual block log file |
| P-CHAN-00339 [▸ 288] | | mdi_log_file_max_size | Maximum size of the manual block log file |
| P-CHAN-00340 [▸ 313] | | position_lookahead_contour_path | Relating the reserved distance for time preview to contour |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00341 [▸ 280] | | opt_insert_flush_continue | Optimised insertion of #FLUSH CONTINUE |
| P-CHAN-00343 [▸ 243] | man_mode.vector_limit. | j_max_ori | Command jerk of an orientation axis in manual mode |
| P-CHAN-00344 [▸ 284] | | suppress_trafo_curr_pos | Suppressed output of actual positions transformed kinematically and/or in the Cartesian system |
| P-CHAN-00345 [▸ 258] | | enable_time_based_lift | Switch-over to time-based calculation when an axis is lifted |
| P-CHAN-00346 [▸ 280] | | enable_mod_axis_always_shortest_way | Positioning of modulo axes on shortest path |
| P-CHAN-00347 [▸ 305] | | no_init_prog_start | Start NC programs without initialisation |
| P-CHAN-00349 [▸ 195] | prog_start.slope. | mode_ramp_time | Profile calculation mode |
| P-CHAN-00350 [▸ 129] | vector. | cir_radial_acc_limit | Implicit dynamic limit in the circle |
| P-CHAN-00351 [▸ 128] | vector. | acc_dec_unit | Unit for path acceleration/ deceleration and jerk |
| P-CHAN-00353 [▸ 127] | | feed_override_weight_dwell_time | Influence of feed override on dwell time |
| P-CHAN-00359 [▸ 314] | | plcopen_implicit_sync | Synchronisation of PLCopen jobs |
| P-CHAN-00360 [▸ 314] | | enable_unit_feet_cut_speed | Unit of constant cutting velocity |
| P-CHAN-00361 [▸ 128] | vector. | radial_acceleration | Radial acceleration limit |
| P-CHAN-00362 | conveyor_sync. | log_number_master | Logical axis number of conveyor belt during belt synchronisation/throughfeed machining |
| P-CHAN-00363 | conveyor_sync. | move_direction | Conveyor belt movement direction in the case of belt synchronisation/throughfeed machining |
| P-CHAN-00364 | conveyor_sync. | x_virtual | Virtual axis in the case of belt synchronisation/throughfeed machining |
| P-CHAN-00365 | conveyor_sync. | sync_in_tolerance | Tolerance window for synchronisation to a conveyor belt (throughfeed machining) |
| P-CHAN-00366 | conveyor_sync. | hold_limit_vel_factor | Factor for reducing the velocity when limiting motion to the end position (throughfeed machining) |
| P-CHAN-00367 | conveyor_sync. | hold_limit_tolerance | Permissible tolerance for the end stop position of the X axis conveyor belt coordinate system (throughfeed machining) |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00368 | conveyor_sync. | cart_t0_shift_x | X offset of the Cartesian basic coordination system for belt synchronisation/throughfeed machining |
| P-CHAN-00369 | conveyor_sync. | cart_t0_shift_y | Y offset of the Cartesian basic coordination system for belt synchronisation/throughfeed machining |
| P-CHAN-00370 | conveyor_sync. | cart_t0_shift_z | Z offset of the Cartesian basic coordination system for belt synchronisation/throughfeed machining |
| P-CHAN-00371 | conveyor_sync. | cart_t0_rot_a | A rotation of the Cartesian basic coordination system for belt synchronisation/ throughfeed machining |
| P-CHAN-00372 | conveyor_sync. | cart_t0_rot_b | B rotation of the Cartesian basic coordination system for belt synchronisation/ throughfeed machining |
| P-CHAN-00373 | conveyor_sync. | cart_t0_rot_c | C rotation of the Cartesian basic coordination system for belt synchronisation/ throughfeed machining |
| P-CHAN-00374 | conveyor_sync. | pos_limit | End stop position of the X axis in the conveyor belt coordinate system (throughfeed machining) |
| P-CHAN-00375 [▶ 224] | hsc.gen. | slope_segmentation_nbr_blocks | Maximum number of segmented blocks for feed profile planning (HSC) |
| P-CHAN-00378 [▶ 289] | error_filter. | reason | Error cause (filtering error messages in the channel) |
| P-CHAN-00379 [▶ 289] | error_filter. | action | Error action (filtering error messages in the channel) |
| P-CHAN-00380 [▶ 290] | error_filter. | conditional_activation | Conditional activation (filtering error messages in the channel) |
| P-CHAN-00381 [▶ 290] | error_filter. | conditional_action | Conditional action (filtering error messages in the channel) |
| P-CHAN-00382 [▶ 291] | error_filter. | conditional_param | Conditional filter activation (filtering error messages in the channel)) |
| P-CHAN-00383 [▶ 291] | error_filter. | conditional_output | Output of additional error information (filtering error messages in the channel)) |
| P-CHAN-00384 [▶ 307] | dcc. | active | Selecting/deselecting the dynamic contour control function (Dynamic Contour Control) |
| P-CHAN-00385 [▶ 307] | dcc. | call | Select calculation method (Dynamic Contour Control) |
| P-CHAN-00386 [▶ 309] | geo_feed_adapt. | active | Select/deselect the constant surface feed rate function |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00387 [▶ 309] | geo_feed_adapt. | call | Selection of calculation method (surface feed rate) |
| P-CHAN-00388 [▶ 307] | dcc.param. | f[i] | REAL64 dynamic contour control input parameters (Dynamic Contour Control) |
| P-CHAN-00389 [▶ 307] | dcc.param. | i[i] | SGN32 Dynamic Contour Control input parameters |
| P-CHAN-00390 [▶ 309] | geo_feed_adapt.param. | f[i] | REAL64 input parameters for surface feed rate |
| P-CHAN-00391 [▶ 309] | geo_feed_adapt.param. | i[i] | SGN32 input parameters for surface feed rate |
| P-CHAN-00392 [▶ 312] | debug. | prg_trace | Trace parameters and variables (debug data of NC programs) |
| P-CHAN-00393 [▶ 292] | coordinate_system. | rotation_mode_fixed | Definition of rotation mode for a coordinate system |
| P-CHAN-00394 [▶ 296] | coordinate_system. | rotation_sequence | Definition of rotation sequence for a coordinate system |
| P-CHAN-00395 [▶ 296] | coordinate_system. | display_global | Select the display coordinate system of the second path with 2-path programming |
| P-CHAN-00396 [▶ 298] | coordinate_system. | 2nd_path_on_actual_1st_path | Select the reference coordinate system to define the coordinate system of the second path |
| P-CHAN-00397 [▶ 301] | coordinate_system. | axes_offsets_layer_specific | Manage axis offsets in the tracked axes for a specific coordinate system |
| P-CHAN-00398 [▶ 301] | coordinate_system. | intersection | Select/deselect calculation of intersecting points with coordinate system planes with 2-path programming |
| P-CHAN-00399 [▶ 312] | provide_channel_interface. | track_cs | Automatic enable of channel interface for synchronous dynamic CS operations |
| P-CHAN-00400 [▶ 305] | | channel_id | Logical number of an NC channel for CNC objects |
| P-CHAN-00401 [▶ 310] | path[i]. | dir | Path specification to NC programs |
| P-CHAN-00402 [▶ 310] | path[i]. | id | Logical path number for a program path |
| P-CHAN-00403 [▶ 310] | path[i]. | type | Type of path of a program path |
| P-CHAN-00404 [▶ 311] | path[i]. | priority | Priority of a program path |
| P-CHAN-00405 [▶ 224] | hsc.gen. | max_track_ratio | HSC deselection response for tracking axes |
| P-CHAN-00406 [▶ 47] | configuration.rt_cyles. | enable | Activating real-time cycles |
| P-CHAN-00407 [▶ 47] | configuration.rt_cyles. | memory | Memory for real-time cycles |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00415 [▶ 154] | spindel[i]. | optional | Identify an optional spindle |
| P-CHAN-00416 [▶ 165] | syn_chk. | record_result | Write results of syntax check to file |
| P-CHAN-00417 [▶ 315] | | ori_interpolation_mode | Orientation interpolation mode in conjunction with complete kinematic transformations |
| P-CHAN-00418 [▶ 48] | configuration.decoder. | var_memory | Memory size for V.CYC variables |
| P-CHAN-00419 [▶ 49] | | max_number_of_tools | Maximum number of tools which can be loaded from lists |
| P-CHAN-00420 [▶ 316] | | remain_tool_length_in_ax | Axis-specific orientation of tool length compensation |
| P-CHAN-00421 [▶ 220] | hsc.bspline. | auto_off_g60 | Deselecting spline at programmed exact stop G60 (B Spline) |
| P-CHAN-00422 [▶ 316] | | disable_feed_factor | Deactivating feed factor effect |
| P-CHAN-00423 [▶ 262] | | suppress_trafo_in_g0_blocks | Suppress an active kinematic transformation at G0 |
| P-CHAN-00424 [▶ 48] | | v_ch_memory | Memory size for V.CH. variables |
| P-CHAN-00425 [▶ 318] | rt_cycles. | max_duration | Max. execution time of real-time cycles |
| P-CHAN-00426 [▶ 318] | rt_cycles. | cont_steps | Number of elementary instructions for time check with real-time cycles |
| P-CHAN-00427 [▶ 319] | rt_cycles. | max_steps | Max. number of elementary instructions per real-time cycles per cycle |
| P-CHAN-00428 [▶ 257] | | calc_average_feed_ahead | Deactivate time ahead calculation based on mean feed velocity |
| P-CHAN-00429 [▶ 207] | | d_prog_file | Subroutine call with D word |
| P-CHAN-00433 [▶ 211] | | final_prog_file_mode | Mode of implicit subroutine call at program end |
| P-CHAN-00434 [▶ 285] | | mirror_display_positions | Consider mirroring in display and manual mode |
| P-CHAN-00435 [▶ 106] | | kasto_techno_output_with_signal | Output an MNE_SNS technology function at measurement event |
| P-CHAN-00436 [▶ 261] | ori. | tool_ax_in_plane | Plane parallel to tool axis plane |
| P-CHAN-00437 [▶ 205] | | err_outp_mode_move | Error message output after manual mode axis overrun |
| P-CHAN-00438 [▶ 171] | trafo[j]. | corr | Correction values for kinematic parameters |
| P-CHAN-00439 [▶ 314] | | contouring_consider_inch | Contouring method in inch units |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00442 [▸ 205] | | suppress_workspace_monitoring_manual_mode | Suppression of workspace monitoring function in manual operation |
| P-CHAN-00443 [▸ 171] | trafo[j] or kin_step[i].trafo[j]. | name | Unique name of the kinematic |
| P-CHAN-00446 [▸ 171] | trafo[j] or kin_step[i].trafo[j]. | base[k] | Coupling kinematic: Base offsets at base |
| P-CHAN-00447 [▸ 178] | trafo[j].group[k] or kin_step[i].trafo[j].group[k]. | name | Name of a group of a coupling kinematic |
| P-CHAN-00448 [▸ 179] | trafo[j].group[k] or kin_step[i].trafo[j].group[k]. | workpiece_cs | Definition of a group containing the tool of a coupling kinematic |
| P-CHAN-00449 [▸ 179] | trafo[j].group[k] or kin_step[i].trafo[j].group[k]. | chain[m] | Coupling kinematic: Structure of the kinematic chain in a group |
| P-CHAN-00450 [▸ 180] | trafo[j].group[k] or kin_step[i].trafo[j].group[k]. | move_prio[m] | Coupling kinematic: Split the TCP movement among the partial kinematics |
| P-CHAN-00451 [▸ 278] | | mode_implicite_sync_indp_asyn_axis | Operating principle of implicit synchronisation with independent asynchronous axes |
| P-CHAN-00452 [▸ 172] | | default_name_of_kin.step | Name of the default transformation |
| P-CHAN-00453 [▸ 317] | | dyn_calc_axis_poly | Dynamic look-ahead for programmed axis polynomials |
| P-CHAN-00454 [▸ 172] | | kinematik_name | Name of the default kinematic after start-up |
| P-CHAN-00455 [▸ 225] | | channel_init_actpos_exclusive | Channel initialisation with actual values |
| P-CHAN-00456 [▸ 189] | | trafo_mode | Mode of the kinematic transformation |
| P-CHAN-00457 [▸ 263] | | post_segmentation_length | Limiting the path length of motion blocks |
| P-CHAN-00458 [▸ 180] | | lock_dof | Coupling kinematics |
| P-CHAN-00459 [▸ 212] | | suppress_soft_limit_monitoring_axes | Suppress software limit switch monitor on command value side |
| P-CHAN-00460 [▸ 121] | | retain_last_ax_config | Retention of last axis configuration in the channel |
| P-CHAN-00463 [▸ 246] | | create_cycle_param_on_read | Create @P parameters automatically |
| P-CHAN-00464 [▸ 189] | limit.kin[i]. | active | Activate TCP velocity monitoring |
| P-CHAN-00465 [▸ 190] | limit.kin[i]. | name | Transformation name of the kinematic used |
| P-CHAN-00466 [▸ 190] | limit.kin[i].velocity. | max | TCP velocity monitoring: Velocity limit |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00467 [▸ 317] | | contour_processing_memory | Memory size for contour machining |
| P-CHAN-00468 [▸ 224] | hsc.gen. | use_analytic_dyn_calc | Analytical dynamic calculation for spline curves |
| P-CHAN-00469 [▸ 190] | limit.kin[i]. | mode | TCP velocity monitoring: Mode for tool head offsets |
| P-CHAN-00470 [▸ 246] | hscs.deselect. | prog | Cycles HSC settings: Deselect file name |
| P-CHAN-00471 [▸ 246] | hscs.deselect. | tolerance | Cycles HSC settings: Deselect - Tolerance |
| P-CHAN-00472 [▸ 247] | hscs.rough. | prog | Cycles HSC settings: Roughing file name |
| P-CHAN-00473 [▸ 247] | hscs.rough. | tolerance | Cycles HSC settings: Roughing - Tolerance |
| P-CHAN-00474 [▸ 247] | hscs.prefinish. | prog | Cycles HSC settings: Pre-finishing file name |
| P-CHAN-00475 [▸ 247] | hscs.prefinish. | tolerance | Cycles HSC settings: Pre-finishing - Tolerance |
| P-CHAN-00476 [▸ 248] | hscs.finish. | prog | Cycles HSC settings: Finishing file name |
| P-CHAN-00477 [▸ 248] | hscs.finish. | tolerance | Cycles HSC settings: Finishing - Tolerance |
| P-CHAN-00478 [▸ 190] | limit.kin[i].velocity. | ipo_weight_factor | TCP velocity monitoring: Weighting factor |
| P-CHAN-00479 [▸ 49] | configuration.decoder. | pattern_processing_memory | Memory size for pattern definition |
| P-CHAN-00480 [▸ 48] | configuration.rt_cycles. | buffers | Max. number of actions in real-time cycles |
| P-CHAN-00481 [▸ 246] | | cycle_stack_memory | Memory size for @P parameters |
| P-CHAN-00482 [▸ 99] | | tool_life_with_d | Tool life recording with D word |
| P-CHAN-00486 [▸ 257] | | in_position_ignored_movement | Check exact stop with additive movements |
| P-CHAN-00487 [▸ 286] | | trc_inverse_display_pcs_pos | TRC inverse display PCS positions |
| P-CHAN-00488 [▸ 49] | configuration.decoder. | cycle_stack_add_var_count | Additional number of variables for @P parameter |
| P-CHAN-00489 [▸ 285] | | display_pcs_limits | Display motion range limits in PCS coordinate system |
| P-CHAN-00490 [▸ 292] | coordinate_system.def[i]. | id | Coordinate system identifier |
| P-CHAN-00491 [▸ 292] | coordinate_system.def[i].path[j].translation. | t1 | Translation of first axis |
| P-CHAN-00492 [▸ 293] | coordinate_system.def[i].path[j].translation. | t2 | Translation of second axis |
| P-CHAN-00493 [▸ 293] | coordinate_system.def[i].path[j].translation. | t3 | Translation of third axis |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00494 [▶ 293] | coordinate_system.def[i].path[j].rotation. | a1 | Rotation angle of first axis |
| P-CHAN-00495 [▶ 293] | coordinate_system.def[i].path[j].rotation. | a2 | Rotation angle of second axis |
| P-CHAN-00496 [▶ 294] | coordinate_system.def[i].path[j].rotation. | a3 | Rotation angle of third axis |
| P-CHAN-00498 [▶ 213] | | range_check_prog_soft_limits | Behaviour with programmed software limit switches |
| | | | |
| P-CHAN-00500 [▶ 41] | configuration.decoder. | function | Define the decoder functionalities |
| P-CHAN-00501 [▶ 41] | configuration.decoder. | log_entry_number | Maximum number of logged decoder events |
| P-CHAN-00502 [▶ 42] | configuration.decoder. | log_level | Define the type of logged decoder events |
| P-CHAN-00503 [▶ 43] | configuration.decoder. | max_cache_number | Maximum number of possible cache files |
| P-CHAN-00504 [▶ 44] | configuration.decoder. | max_cache_size | Maximum size of a cache file |
| P-CHAN-00505 [▶ 44] | configuration.decoder. | max_local_subroutine_definitions | Maximum number of local subroutine definitions |
| P-CHAN-00506 [▶ 45] | configuration.decoder. | max_vol_comp_measurement_records | Maximum number of measurement records for machine calibration |
| P-CHAN-00507 [▶ 42] | configuration.decoder. | fct_enable[<idx>] where idx 0, 1 | Conditional enable of decoder functionalities |
| P-CHAN-00508 [▶ 42] | configuration.decoder. | fct_condition[<idx>] where idx 0, 1 | Condition for enabling the decoder functionalities |
| P-CHAN-00509 [▶ 46] | configuration.decoder. | macro_number | Total number of NC macros |
| P-CHAN-00510 [▶ 46] | configuration.decoder. | macro_definition_number | Maximum number of predefined MC macros |
| P-CHAN-00511 [▶ 46] | configuration.decoder. | macro_symbol_len | Maximum number of characters in the macro name |
| P-CHAN-00512 [▶ 46] | configuration.decoder. | macro_nc_code_len | Maximum number of characters in the macro content |
| P-CHAN-00514 [▶ 49] | configuration.decoder. | msg_diag_memory | Size of message memory for diagnosis data |
| P-CHAN-00515 [▶ 50] | configuration.decoder. | string_label_number | Maximum number of string labels |
| P-CHAN-00516 [▶ 50] | configuration.decoder. | expression_label_number | Maximum number of expression labels |
| P-CHAN-00517 [▶ 50] | configuration.decoder. | p_param_number | Maximum number of P parameters |
| P-CHAN-00550 [▶ 52] | configuration.tool_radius_comp. | function | Define the functionalities of the TRC |
| P-CHAN-00551 [▶ 52] | configuration.tool_radius_comp. | log_entry_number | Maximum number of logged TRC log_level events |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00552 [▸ 52] | configuration.tool_radius_comp. | log_level | Define the type of logged decoder events |
| P-CHAN-00553 [▸ 54] | configuration.tool_radius_comp. | fct_enable[<idx>] where idx 0, 1 | Conditional enable of TRC functionalities |
| P-CHAN-00554 [▸ 54] | configuration.tool_radius_comp. | fct_condition[<idx>] where idx 0, 1 | Condition for enabling the TRC functionalities |
| P-CHAN-00600 [▸ 54] | configuration.path_preparation. | function | Defining functionalities for path preparation |
| P-CHAN-00601 [▸ 56] | configuration.path_preparation. | log_entry_number | Maximum number of entries in the history buffer |
| P-CHAN-00602 [▸ 56] | configuration.path_preparation | log_level | Define the type of logged path events |
| P-CHAN-00603 [▸ 57] | configuration.path_preparation. | m_pre_output_lookahead | Maximum number of blocks considered for pre-output of M functions |
| P-CHAN-00604 [▸ 58] | configuration.path_preparation. | m_pre_output_max_distance | Maximum path for pre-output of M functions |
| P-CHAN-00605 [▸ 60] | configuration.path_preparation. | fct_enable[<idx>] where idx 0, 1 | Conditional enable of path functionalities |
| P-CHAN-00606 [▸ 61] | configuration.path_preparation. | fct_condition[<idx>] where idx 0, 1 | Condition for enabling the path functionalities |
| P-CHAN-00650 [▸ 61] | configuration.interpolator. | function | Definition of interpolator functionalities |
| P-CHAN-00651 [▸ 62] | configuration.interpolator. | log_entry_number | Maximum number of logged interpolator events |
| P-CHAN-00652 [▸ 62] | configuration.interpolator. | log_level | Define the type of logged interpolator events |
| P-CHAN-00653 [▸ 63] | configuration.interpolator. | (parameter, param) number_blocks_lah | User-specific size of look-ahead buffer |
| P-CHAN-00654 [▸ 63] | configuration.interpolator. | blocks_per_call | Reduce the computing time of the interpolator |
| P-CHAN-00655 [▸ 64] | configuration.interpolator. | fct_enable[<idx>] where idx 0, 1 | Conditional enable of interpolator functionalities |
| P-CHAN-00656 [▸ 64] | configuration.interpolator. | fct_condition[<idx>] where idx 0, 1 | Condition for enabling the interpolator functionalities |
| P-CHAN-00657 [▸ 64] | configuration.interpolator. | dyn_cs_history_max | Number of logs of the dynamic coordinate system |
| P-CHAN-00658 [▸ 65] | configuration.interpolator. | contour_lookahead_log_max | Maximum number of contour elements in the contour look-ahead |
| | | | |
| P-CHAN-00751 [▸ 321] | | dpl_tol_limit_change | Tolerance limit for dynamic limitation of axis positions |
| P-CHAN-00752 [▸ 294] | trafo_stack[i]. | name | Name of transformation stack |
| P-CHAN-00753 [▸ 295] | trafo_stack[i]. | kin[j] | Name of the kinematic |
| P-CHAN-00754 [▸ 295] | trafo_stack[i].cs. | id[k] | ID of CS offset |

| ID | Structure | Parameter | Functionality |
|---|---|---|---|
| P-CHAN-00755 [▶ 295] | trafo_stack[i].acs. | id[k] | ID or ACS offset |
| P-CHAN-00756 [▶ 295] | trafo_stack[i].bcs. | id[k] | ID or BCS offset |
| P-CHAN-00757 [▶ 296] | | trafo_stack_name_active_prog_start | Name of activated transformation stack at program start |
| P-CHAN-00758 [▶ 320] | backward_compatibility. | axis_spline_ramp_time | Calculation mode of resulting ramp time with #CS and #TRAFO |
| P-CHAN-00759 [▶ 213] | | swe_limits_additive_manual_mode | Limitation of software limit switches in manual mode with parallel interpolation |
| P-CHAN-00760 [▶ 90] | | pre_output_tolerance | Tolerance for deviation with pre-output position of M/H function |
| P-CHAN-00761 [▶ 321] | tapping. | use_actual_position | Tapping with actual positions of the spindle |
| P-CHAN-00762 [▶ 321] | tapping. | n_cycles | Number of filter cycles to filter actual positions of the spindle |

# 1 Description of elements

## 1.1 Configuration of the NC channel.

**The parameters listed in this section cannot be modified while the system is running. A parameter update requires a system restart.**

Parameter names start with the prefix "configuration".

### 1.1.1 Decoding

#### 1.1.1.1 Define the decoder functionalities (P-CHAN-00500)

| **P-CHAN-00500** | **Definition of decoder functions** |
|---|---|
| Description | The parameter defines specific functionalities for decoding. This disables specific functions for testing or for performance reasons. |
| Parameter | configuration.decoder.function |
| Data type | STRING |
| Data range | FCT_USE_CACHED_FILES: Enabling file caching<br>FCT_VOL_COMP_COMPUTATION: Calculations for machine calibration<br>-: No functionalities defined. |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>Parameterisation example:<br>Caching of maximal 4 files of maximum 4096 bytes each.<br>*configuration.decoder.function FCT_USE_CACHED_FILES*<br>*configuration.decoder.max_cache_number 4*<br>*configuration.decoder.max_cache_size 4096*<br>* Note: The default value of variables is a blank string.<br>Functions can be defined in P-CHAN-00507 [▶ 42] and P-CHAN-00508 [▶ 42] depending on the machining mode. |

#### 1.1.1.2 Maximum number of logged events (P-CHAN-00501)

| **P-CHAN-00501** | **Maximum number of entries in the history buffer** |
|---|---|
| Description | The CNC offers the options of filing events in a history memory (logging entries). This parameter defines the maximum number of logged events. If there are more entries, the oldest one is overwritten. |
| Parameter | configuration.decoder.log_entry_number |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.1.3 Defining the type of logged events (P-CHAN-00502)

| P-CHAN-00502 | Defining the type of logged events |
|---|---|
| Description | The CNC offers the options of filing events in a history memory (logging entries). The parameter permits the user-specific definition of CNC logging entries to be logged. Depending on troubleshooting or the requirement for an analysis, recording of events can be filtered to already reduce the number of entries to be recorded/ analysed from the outset |
| Parameter | configuration.decoder.log_level |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.1.4 Conditional activation of the functionalities (P-CHAN-00507)

| P-CHAN-00507 | Conditional enable of the functionalities. |
|---|---|
| Description | This parameter can be used to configure functions of the CNC that are activated in the channel depending on the program start mode.<br>Several settings can be predefined for activation as soon as an NC program is started in the machining mode listed under the condition P-CHAN-00508 [▶ 42]. |
| Parameter | configuration.interpolator.fct_enable[<idx>] where idx 0, 1 |
| Data type | STRING |
| Data range | FCT_USE_CACHED_FILES: Enabling file caching<br>FCT_VOL_COMP_COMPUTATION: Calculations for machine calibration<br>-: No functionalities defined. |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>* Note: The default value of variables is a blank string.<br>If no individual machining mode is set at program start, this corresponds to the setting ISG_STANDARD.<br>The default settings under index 0 ("fct_enable[0]") are used.<br>For reasons of backward compatibility, P-CHAN-00507 "fct_enable[0]" corresponds to the previous parameter "function" P-CHAN-00500 [▶ 41].<br>Both notations or parameters can be used alternatively. |

### 1.1.1.5 Condition for activating the functionalities (P-CHAN-00508)

| P-CHAN-00508 | Condition for activating the functionalities |
|---|---|

| | |
|---|---|
| Description | This parameter specifies a condition.<br>If the processing mode complies with the specified condition at program start, the related functionalities are activated at program start.<br>The corresponding functionalities are defined in P-CHAN-00507 [▶ 42] .<br>Example<br>`configuration.decoder.fct_enable[0]    FCT_DEFAULT`<br>`configuration.decoder.fct_condition[0] ISG_STANDARD`<br>`configuration.decoder.fct_enable[1]    FCT_USE_CACHED_FILES`<br>`configuration.decoder.fct_condition[1] BLOCK_SEARCH`<br>The condition is assigned to the corresponding functionality by an index. |
| Parameter | configuration.decoder.fct_condition[<idx>] where idx 0, 1 |
| Data type | STRING |
| Data range | See Condition for activating the functionalities (P-CHAN-00508) [▶ 43] |
| Dimension | ---- |
| Default value | ISG_STANDARD |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

| **Identifier7 Syntax** | **Meaning** |
|---|---|
| ISG_STANDARD | Normal program start |
| BLOCK_SEARCH (SV) | Block search (FCT-C6) |
| CONTOUR_VISU (SOLLKON) | Command contour visualisation (FCT-C17) |
| ON_LINE | Online visualisation (FCT-C17) |
| SYNTAX_CHECK (SYNCHK) | Syntax check (FCT-C9) |
| PROD_TIME | Machining time calculation |
| ONLINE_PROD_TIME | Online production time calculation |
| DRY_RUN (MACHINE_LOCK) | Dry run (FCT-C17) |
| SCENE | 3D visualisation (FCT-C17) |

The syntax in brackets is permissible as an alternative.

### 1.1.1.6 Maximum number of possible cache files (P-CHAN-00503)

| **P-CHAN-00503** | **Maximum number of possible cache files** |
|---|---|
| Description | This parameter permits the user-specific definition of the maximum number of files available in the NC program cache. |
| Parameter | configuration.decoder.max_cache_number |
| Data type | UNS32 |
| Data range | 0 <= P-CHAN-00503 <= MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | If the File Caching function is active with *FCT_USE_CACHED_FILES*, the default value is 4.<br>Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>Parameterisation example:<br>Caching of maximal 6 files of maximum 6000 bytes each.<br>*configuration.decoder.function FCT_USE_CACHED_FILES*<br>*configuration.decoder.max_cache_number 6*<br>*configuration.decoder.max_cache_size 6000* |

### 1.1.1.7 Maximum size of a cache file (P-CHAN-00504)

| **P-CHAN-00504** | **Maximum size of a cache file** |
|---|---|
| Description | This parameter permits the user- specific definition of the maximum file size of an NC program in the cache. |
| Parameter | configuration.decoder.max_cache_size |
| Data type | UNS32 |
| Data range | 0 <= P-CHAN-00504 <= MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | If the File Caching function is active with *FCT_USE_CACHED_FILES*, the default value is 4096.<br>Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>Parameterisation example:<br>Caching of maximal 6 files of maximum 6000 bytes each.<br>*configuration.decoder.function FCT_USE_CACHED_FILES*<br>*configuration.decoder.max_cache_number 6*<br>*configuration.decoder.max_cache_size 6000* |

### 1.1.1.8 Maximum number of local subroutine definitions (P-CHAN-00505)

| **P-CHAN-00505** | **Maximum number of local subroutine definitions** |
|---|---|
| Description | This parameter permits the user-specific definition of the maximum number of local subroutine definitions (%L ...) in an NC program. |
| Parameter | configuration.decoder.max_local_subroutine_definitions |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 50 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>Parameterisation example:<br>*configuration.decoder.max_local_subroutine_definitions 70* |

### 1.1.1.9 Maximum number of measurement records for machine calibration (P-CHAN-00506)

| P-CHAN-00506 | Maximum number of measurement records for machine calibration |
|---|---|
| Description | This parameter defines the maximum number of measurement records during machine calibration using the ISG calibration cycles. This parameter is used internally by measurement cycles and should only be configured or changed in consultation with ISG. |
| Parameter | configuration.decoder.max_vol_comp_measurement_records |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>Parameterisation example: A maximum of 50 measurement records are logged.<br>*configuration.decoder.function FCT_VOL_COMP_COMPUTATION*<br>*configuration.decoder.max_vol_comp_measurement_records 50* |

### 1.1.1.10 Total number of NC macros (P-CHAN-00509)

| P-CHAN-00509 | Total number of NC macros |
|---|---|
| Description | This parameter defines the maximum number of user macros.<br>The total number comprises both the predefined macros in the channel parameter list and the macros directly defined within the NC programs.<br>For more information see:<br>• Programming macros in the NC program (Macroprogramming (# INIT MACRO TAB))<br>• and Definition of macros (makro_def[i].*) [▶ 160] |
| Parameter | configuration.decoder.macro_number |
| Data type | UNS16 |
| Data range | 0 < P-CHAN-00509 < 2000 |
| Dimension | ---- |
| Default value | 100 |
| Remarks | Parameter is available as of the following Builds: V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.1.11 Maximum number of predefined NC macros (P-CHAN-00510)

| P-CHAN-00510 | Maximum number of predefined NC macros |
|---|---|
| Description | This parameter defines the maximum number of predefined macros in the channel parameter list, see Definition of macros (makro_def[i].*) [▶ 160].<br>NC macros can then be used in the NC program, see Including the 'Macro' functionality. |
| Parameter | configuration.decoder.macro_definition_number |
| Data type | UNS16 |
| Data range | 0 < P-CHAN-00510 < P-CHAN-00509 [▶ 46] |
| Dimension | ---- |
| Default value | 50 |
| Remarks | Parameter is available as of the following Builds: V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.1.12 Maximum number of characters in macro name (P-CHAN-00511)

| P-CHAN-00511 | Maximum number of characters in macro name |
|---|---|
| Description | This parameter defines the number of characters for the maximum length of macro names.<br>For more information see<br>• Macroprogramming (# INIT MACRO TAB)<br>• Definition of macros (makro_def[i].*) [▶ 160] |
| Parameter | configuration.decoder.macro_symbol_len |
| Data type | UNS16 |
| Data range | 0 < P-CHAN-00511 <= 128 |
| Dimension | Byte |
| Default value | 30 |
| Remarks | Parameter is available as of the following Builds: V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.1.13 Maximum number of characters in macro content (P-CHAN-00512)

| P-CHAN-00512 | Maximum number of characters in macro content |
|---|---|

| | |
|---|---|
| Description | This parameter defines the maximum number of characters for the macro content (executable NC code).<br>For more information on NC macros, see Macroprogramming (# INIT MACRO TAB). |
| Parameter | configuration.decoder.macro_nc_code_len |
| Data type | UNS16 |
| Data range | 0 < P-CHAN-00512 <= 1024 |
| Dimension | Byte |
| Default value | 80 |
| Remarks | Parameter is available as of the following Builds: V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.1.14 Activating real-time cycles (P-CHAN-00406)

| **P-CHAN-00406** | **Activating real-time cycles** |
|---|---|
| Description | This parameter enables the real-time cycle function in the NC channel.<br>The controller must be restarted to adopt the change.<br>Example:<br>`configuration.rt_cycles.enable 1` |
| Parameter | configuration.rt_cycles.enable |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is available as of CNC Build V3.1.3107.10 or higher.<br>Use of the parameter “rt_cycles.enable“<br>`rt_cycles.enable 1`<br>(as of V3.1.3105) continues to be supported. |

### 1.1.1.15 Memory for real-time cycles (P-CHAN-00407)

| **P-CHAN-00407** | **Memory size for real-time cycles** |
|---|---|
| Description | This parameter defines the size of the memory for real-time cycles. The size of the memory is specified in bytes.<br>The controller must be restarted to adopt the change. Then the specified memory is also available for real-time cycles.<br>Example:<br>`configuration.rt_cycles.memory 60000` |
| Parameter | configuration.rt_cycles.memory |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) - 1 |
| Dimension | ---- |
| Default value | 48000 |
| Remarks | **Note:**<br>The assignment of P-CHAN-00407 is only necessary if the memory set by default is no longer sufficient due to activation of the real-time cycles (P-CHAN-00406 [▶ 47]).<br><br>This parameter is available as of CNC Build V3.1.3107.10 or higher.<br>Use of the parameter “rt_cycles.memory“<br>`rt_cycles.memory 60000`<br>(as of V3.1.3105) continues to be supported. |

## 1.1.1.16 Max. number of actions in a real-time cycle (P-CHAN-00480)

| P-CHAN-00480 | Max. number of actions in a real-time cycle |
|---|---|
| Description | This parameter defines the maximum number of possible actions within a real-time cycle.<br>Possible actions include single-axis movement, spindle command, etc.<br>If too many actions are commanded within a real-time cycle, error ID 51028 is output. |
| Parameter | configuration.rt_cycles.buffers |
| Data type | UNS16 |
| Data range | 0 ... MAX(UNS16) - 1 |
| Dimension | ---- |
| Default value | 5 |
| Remarks | The parameter is available as of V3.1.3107.10 |

## Size of additional memory for V.CYC variables (P-CHAN-00418)

| P-CHAN-00418 | Size of additional memory for V.CYC variables |
|---|---|
| Description | This parameter defines the size of the memory area in bytes for V.CYC variables in order to expand the existing standard memory of 8000 bytes.<br>A change in the parameter requires a controller restart. If the parameter is changed while the controller is running, the warning P-ERR-21916 is output and the previous parameter value is retained. |
| Parameter | configuration.decoder.var_memory |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) - 1 |
| Dimension | Byte |
| Default value | 0 |
| Remarks | Parameter is available as of the following Builds: V3.1.3079.17 ; V3.1.3107.10<br>Example:<br>`configuration.decoder.var_memory 2000`<br>Direct use of the parameter “var_memory” without any structure (as of V3.01.3064) is still supported but should no longer be used in new applications.<br>`var_memory 2000` |

## 1.1.1.17 Memory size for V.CH. variables (P-CHAN-00424)

| P-CHAN-00424 | Memory size for V.CH. variables |
|---|---|
| Description | This parameter defines the size of the memory in bytes for V.CH. variables.<br>Example:<br>`configuration.decoder.v_ch_memory 10000` |
| Parameter | configuration.decoder.v_ch_memory |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) - 1 |
| Dimension | Byte |
| Default value | 0 |
| Remarks | The memory contains application data and internal management data. This means that the actually available user memory is always smaller than the set value.<br>This parameter is available as of Version V3.1.3107.10.<br>Direct use of the parameter “v_ch_memory“ without any structure (as of V3.1.3104) is still supported but should no longer be used in new applications. |

| | |
|---|---|
| | `v_ch_memory 10000` |

### 1.1.1.18 Number of variables for P@ parameters (P-CHAN-00488)

| **P-CHAN-00488** | **Number of variables for @P parameter** |
|---|---|
| Description | This parameter extends the memory area for transfer parameters (@P parameters) of cycles by specifying the number of transfer parameters required. |
| Parameter | configuration.decoder.cycle_stack_add_var_count |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) - 1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | The parameter is available as of V3.1.3079.31 |

### 1.1.1.19 Memory size for pattern definition (P-CHAN-00479)

| **P-CHAN-00479** | **Memory size for pattern definition** |
|---|---|
| Description | This parameter defines the size in bytes of the memory area for saving patterns (#PATTERN BEGIN/ #PATTERN END).<br>A change in the parameter requires a controller restart. |
| Parameter | configuration.decoder.pattern_processing_memory |
| Data type | UNS32 |
| Data range | 0 <= P-CHAN-00479 < MAX_UNS32 |
| Dimension | Byte |
| Default value | 0 |
| Remarks | Parameter available as of V3.1.3079.37 |

### 1.1.1.20 Size of message memory for diagnosis data (P-CHAN-00514)

| **P-CHAN-00514** | **Size of message memory for diagnosis data** |
|---|---|
| Description | Outputting a message with the #MSG command may have diagnosis data as the receiver (#MSG DIAG).<br>These messages are logged when diagnosis data is generated.<br>This parameter indicates the size of the memory for these messages. The default value is 0, i.e. the function is disabled. |
| Parameter | configuration.decoder.msg_diag_memory |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) - 1 |
| Dimension | Byte |
| Default value | 0 |
| Remarks | With complex NC programs or NC cycles, it is helpful<br>to output messages occasionally using partly accomplished<br>subtasks. If there is an error, this facilitates the diagnosis. |

### 1.1.1.21 Maximum number of tools from lists (P-CHAN-00419)

| **P-CHAN-00419** | **Maximum number of tools which can be loaded from lists** |
|---|---|

| | |
|---|---|
| Description | This parameter defines the number of tools for which memory space is reserved in the controller<br>Example:<br>`configuration.decoder.max_number_of_tools 100` |
| Parameter | configuration.decoder.max_number_of_tools |
| Data type | SGN32 |
| Data range | -1 ... MAX(SGN32) |
| Dimension | ---- |
| Default value | -1 * |
| Remarks | *P-CHAN-00419 is only relevant if tools are loaded from a tool list [TOOL] at controller start-up. If an external tool management system is used (P-CHAN-00016 [▶ 95]), P-CHAN-00419 has no meaning. No memory space is reserved,<br>If the parameter is not assigned (-1), memory space is created for 200 tools. If the value is >= 0, the exact amount of memory space is reserved for the number specified.<br>Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>Use of the parameter “max_number_of_tools” is still supported.<br>`max_number_of_tools 100` |

### 1.1.1.22 Maximum number of string labels (P-CHAN-00515)

| P-CHAN-00515 | Maximum number of string labels. |
|---|---|
| Description | This parameter defines the number of usable string labels. |
| Parameter | configuration.decoder.string_label_number |
| Data type | UNS16 |
| Data range | 1 < 1000 |
| Dimension | --- |
| Default value | 200 |
| Remarks | Available as of V3.1.3079.42 |

### 1.1.1.23 Maximum number of expression labels (P-CHAN-00516)

| P-CHAN-00516 | Maximum number of expression labels. |
|---|---|
| Description | This parameter defines the number of usable expression labels. |
| Parameter | configuration.decoder.expression_label_number |
| Data type | UNS16 |
| Data range | 1 < 1000 |
| Dimension | --- |
| Default value | 200 |
| Remarks | Available as of V3.1.3079.42 |

### 1.1.1.24 Maximum number of P parameters (P-CHAN-00517)

| P-CHAN-00517 | Maximum number of P parameters |
|---|---|
| Description | This parameter defines the number of P parameters. |
| Parameter | configuration.decoder.p_param_number |
| Data type | UNS32 |
| Data range | 0 <= P-CHAN-00517 <= MAX_UNS32 |
| Dimension | --- |

| | |
|---|---|
| Default value | 1000 |
| Remarks | Available as of Build V3.1.3079.41 or V3.1.3107.30 |

### 1.1.1.25 Behaviour with messages to ISG_DIAG_BED (P-CHAN-00518)

| **P-CHAN-00518** | **Behaviour with messages to ISG_DIAG_BED** |
|---|---|
| Description | This message defines the behaviour with messages sent via the NC command #MSG to ISG_DIAG_BED, |
| Parameter | configuration.decoder.msg_to_default |
| Data type | STRING |
| Data range | STANDARD : Messages to receiver ISG_DIAG_BED are written.<br>DISABLED : Messages to receiver ISG_DIAG_BED are discarded.<br>WARNING : Messages to receiver ISG_DIAG_BED are discarded and a warning (ID 22194) is output.<br>ERROR : Messages to receiver ISG_DIAG_BED are discarded and an error (ID 22195) is output. |
| Dimension | --- |
| Default value | STANDARD |
| Remarks | Parameter available as of CNC Build V2.11.2059, V2.11.2830, V3.1.3079.43 or V3.1.3107.33. |

### 1.1.1.26 Behaviour with messages to HMI (P-CHAN-00519)

| **P-CHAN-00519** | **Behaviour with messages to HMI** |
|---|---|
| Description | This message defines the behaviour with messages sent via the NC command #MSG to the HMI, |
| Parameter | configuration.decoder.msg_to_hmi |
| Data type | STRING |
| Data range | STANDARD : Messages to receiver HMI are written.<br>DISABLED : Messages to receiver HMI are discarded.<br>WARNING : Messages to receiver HMI are discarded and a warning (ID 22194) is output.<br>ERROR : Messages to receiver HMI are discarded and an error (ID 22195) is output. |
| Dimension | --- |
| Default value | STANDARD |
| Remarks | Parameter available as of CNC Build V2.11.2059, V2.11.2830, V3.1.3079.43 or V3.1.3107.33. |

### 1.1.1.27 Behaviour with messages to PLC (P-CHAN-00520)

| **P-CHAN-00520** | **Behaviour with messages to PLC** |
|---|---|
| Description | This message defines the behaviour with messages sent via the NC command #MSG to the PLC, |
| Parameter | configuration.decoder.msg_to_plc |
| Data type | STRING |

| | |
|---|---|
| Data range | STANDARD : Messages to receiver PLC are written.<br>DISABLED : Messages to receiver SPS are discarded.<br>WARNING : Messages to receiver SPS are discarded and a warning (ID 22194) is output.<br>ERROR : Messages to receiver SPS are discarded and an error (ID 22195) is output. |
| Dimension | --- |
| Default value | STANDARD |
| Remarks | Parameter available as of CNC Build V2.11.2059, V2.11.2830, V3.1.3079.43 or V3.1.3107.33. |

## 1.1.2 Tool radius compensation

### 1.1.2.1 Defining the functionalities for tool radius compensation (P-CHAN-00550)

| **P-CHAN-00550** | **Definition of functionalities for tool radius compensation** |
|---|---|
| Description | This parameter defines individual functionalities for tool radius compensation. |
| Parameter | configuration.tool_radius_comp.function |
| Data type | STRING |
| Data range | MULTI_PATH: 2-path configuration and programming active<br>-: No functionalities defined. |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>* Note: The default value of variables is a blank string.<br>The parameters P-CHAN-00555 [▶ 54] and P-CHAN-00556 [▶ 54] can be used to define functions depending on the machining mode. |

### 1.1.2.2 Maximum number of logged events (P-CHAN-00551)

| **P-CHAN-00551** | **Maximum number of entries in the history buffer** |
|---|---|
| Description | The CNC offers the options of filing events in a history memory (logging entries). This parameter defines the maximum number of logged events. If more entries occur than there is memory space, the oldest entry is overwritten.. |
| Parameter | configuration.tool_radius_comp.log_entry_number |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.2.3 Defining the type of logged events (P-CHAN-00552)

| **P-CHAN-00552** | **Defining the type of logged events** |
|---|---|

| | |
|---|---|
| Description | The CNC offers the options of filing events in a history memory (logging entries). The parameter permits the user-specific definition of CNC logging entries to be logged. Depending on troubleshooting or the analysis requirement, event logging can be filtered in order to reduce the number of entries to be logged or analysed right from the outset. |
| Parameter | configuration.tool_radius_comp.log_level |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.2.4 Conditional enable of the functionalities (P-CHAN-00553)

| **P-CHAN-00553** | Conditional enable of the functionalities |
|---|---|
| Description | This parameter can be used to configure functions of the CNC that are activated in the channel depending on the program start mode.<br><br>Several settings can be predefined for activation as soon as an NC program is started in the machining mode listed under the condition P-CHAN-00554 [▶ 54] . |
| Parameter | configuration.tool_radius_comp.fct_enable[<idx>] where idx 0,1 |
| Data type | STRING |
| Data range | MULTI_PATH: 2-path configuration and programming active<br>-: No functionalities defined. |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>* Note: The default value of variables is a blank string.<br>If no individual machining mode is set at the start of the program, this corresponds to the setting ISG_STANDARD.<br>The default settings under index 0 ("fct_enable[0]") are used.<br>For reasons of backward compatibility, P-CHAN-00556 "fct_enable[0]" corresponds to the previous parameter "function" P-CHAN-00550 [▶ 52].<br>Both notations or parameters can be used alternatively. |

### 1.1.2.5 Condition for enabling the functionalities (P-CHAN-00554)

| **P-CHAN-00554** | Condition for enabling functionalities |
|---|---|
| Description | This parameter specifies a condition.<br><br>If the machining mode corresponds to the specified condition at program start, the associated functionalities are activated at program start.<br>The corresponding functionalities are defined by P-CHAN-00553 [▶ 54] .<br>The condition for the corresponding functionality is assigned by the index: |
| Parameter | configuration.tool_radius_comp.fct_condition[<idx>] where idx 0,1 |
| Data type | STRING |
| Data range | See Condition for activating the functionalities (P-CHAN-00508) [▶ 43] |
| Dimension | ---- |
| Default value | ISG_STANDARD |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

## 1.1.3 Path preparation

### 1.1.3.1 Defining the functionalities for path preparation (P-CHAN-00600)

| **P-CHAN-00600** | **Defining functionalities for path preparation.** |
|---|---|
| Description | This parameter defines the individual functionalities for path preparation. The individual functions can be enabled or disabled for testing or for performance reasons. |
| Parameter | configuration.path_preparation.function |

| | |
|---|---|
| Data type | STRING |
| Data range | See Defining the functionalities for path preparation (P-CHAN-00600) [▸ 55] |
| Dimension | ---- |
| Default value | FCT_DEFAULT |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>Functions can be defined in P-CHAN-00605 [▸ 60] and P-CHAN-00606 [▸ 61] depending on the machining mode. |

**Path preparation function table**

| **Flag** | **Description** |
|---|---|
| FCT_DEFAULT | The functions FCT_FFM \| FCT_PRESEGMENTATION \| FCT_SPLINE \| FCT_POLY \| FCT_CAX \| FCT_CAX_TRACK \| FCT_SEGMENTATION are available. |
| FCT_FFM | Free-form surface mode, #HSC [OPMODE 1 CONTERR 0.01], #HSC [OPMODE 2] |
| FCT_PRESEGMENTATION | Linear pre-segmentation in HSC mode |
| FCT_SPLINE | #HSC[], AKIMA, B-Spline, G150/G151 |
| FCT_POLY | #CONTOUR MODE[], G61, G261/G260 |
| FCT_CAX | C axis processing, i.e. the spindle is embedded in the NC channel. |
| FCT_CAX_TRACK | #CAX TRACK, tracking an axis according to the contour angle |
| FCT_SEGMENTATION | For dynamic segmentation of the path contour, e.g. if the curvature of a polynomial segment varies significantly. |

| **The following functions must also be enabled:** | |
|---|---|
| FCT_LIFT_UP | Automatic lifting/lowering of an axis (path-based coupling).<br>Example: FCT_DEFAULT \| FCT_LIFT_UP |
| FCT_EMF | Edge machining (sharp angle contours).<br>Example: FCT_DEFAULT \| FCT_EMF |
| FCT_EMF_POLY_OFF | Edge machining inactive with polynomials.<br>Contrary to the setting with FCT_EMF, edge signal generation is masked when path polynomial generation is active in the channel. Polynomials are generated for smoothing G261 or when B Spline is active. The resulting geometry is then tangential.<br>Example: FCT_DEFAULT \| FCT_EMF_POLY_OFF |
| FCT_SYNC | Synchronisation of an axis on a path group.<br>Example: FCT_DEFAULT \| FCT_SYNC |
| FCT_PRECON | Optimised planning using #HSC[BSPLINE].<br>Example: FCT_DEFAULT \| FCT_PRECON |
| FCT_LIFT_UP_TIME | Automatic lifting/lowering of an axis (time-based coupling).<br>Example: FCT_DEFAULT \| FCT_LIFT_UP_TIME |
| FCT_PTP | Dynamically optimised contouring of the complete contour.<br>Example: FCT_DEFAULT \| FCT_PTP |
| FCT_M_PRE_OUTPUT | Pre-output of M/H functions (microjoints).<br>Example: FCT_DEFAULT \| FCT_M_PRE_OUTPUT |
| FCT_SURFACE | HSC machining with Surface Optimiser<br>Example: FCT_DEFAULT \| FCT_SURFACE |
| FCT_SEG_CHECK | Block segmentation in combination with path-controlled offset of M functions (dwell time),<br>see P-CHAN-00650 [▸ 61] and Defining the functionalities for path preparation (P-CHAN-00600) [▸ 55]<br>Example: FCT_DEFAULT \| FCT_SEG_CHECK |

| | |
|---|---|
| FCT_NIBBLING | Activate the nibbling function<br>Example: FCT_DEFAULT \| FCT_NIBBLING |
| FCT_PUNCHING | Activate the punching function<br>Example: FCT_DEFAULT \| FCT_PUNCHING |
| FCT_VSM | Activate the velocity smoothing function<br>Example: FCT_DEFAULT \| FCT_VSM<br>as of V3.1.3079.21 |

### 1.1.3.2 Maximum number of logged events (P-CHAN-00601)

| **P-CHAN-00601** | **Maximum number of entries in the history buffer** |
|---|---|
| Description | The CNC offers the options of filing events in a history memory (logging entries). This parameter defines the maximum number of logged events. If there are more entries, the oldest one is overwritten. |
| Parameter | configuration.path_preparation.log_entry_number |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.3.3 Defining the type of logged events (P-CHAN-00602)

| **P-CHAN-00602** | **Defining the type of logged events** |
|---|---|
| Description | The CNC offers the options of filing events in a history memory (logging entries). The parameter permits the user-specific definition of the CNC logging entries to be logged. Depending on troubleshooting or the analysis requirement, event logging can be filtered in order to reduce the number of entries to be logged or analysed right from the outset. |
| Parameter | configuration.path_preparation.log_level |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.3.4 Maximum number of blocks considered for pre-output of M functions (P-CHAN-00603)

| **P-CHAN-00603** | **Maximum number of blocks considered for pre-output of M functions** |
|---|---|
| Description | This parameter sets the configuration of the look-ahead range for the pre-output of M functions (see [FCT-C1]). |
| Parameter | configuration.path_preparation.m_pre_output_lookahead |
| Data type | UNS32 |
| Data range | 0 … 1000 |
| Dimension | ---- |
| Default value | 10 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>Without an explicit setting, the range is limited by default to 10 NC blocks. This number of blocks may be insufficient for a pre-output of the M function at the desired position if the motion blocks are too short or too many control commands are programmed without any motion. In this case, the M function is pre-output to the maximum known path position and a warning is output.<br>**Parameterisation example:**<br>*configuration.path_preparation.function FCT_DEFAULT* \| *FCT_M_PRE_OUTPUT*<br>*configuration.path_preparation.m_pre_output_lookahead* 15 |

Maximum number of blocks considered for pre-output of M functions

```
%microjoint4
N01 G00 G90 X0 Y0
N02 G01 F10000

N01 V.G.M_FCT[100].PRE_OUTP_PATH = 28.6 ; in mm
N20 G91 Y1
N21 Y1 ; -> planned M output at Y1.4 mm
N22 Y1
N23 Y1
…
N39 Y1
; -> real M output due to limitation of the number of blocks
N40 Y1
N41 Y1
N42 Y1
N43 Y1
N44 Y1
N45 Y1
N46 Y1
N47 Y1
N48 Y1
N49 Y1
N50 M100 M26
N99 M30
```

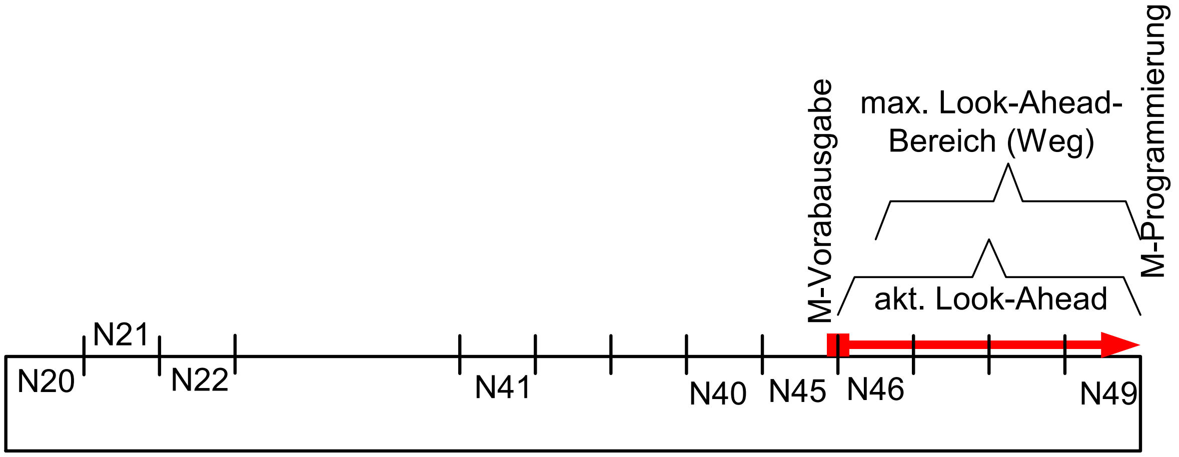


Fig. 1: Limits the pre-output to the maximum look-ahead range (default 10 blocks).

The look-ahead range causes a delay at program start. As a result, only select the number of blocks that are absolutely necessary.

## 1.1.3.5 Maximum path for pre-output of M functions (P-CHAN-00604)

| **P-CHAN-00604** | **Maximum path for pre-output of M functions** |
|---|---|
| Description | This parameter sets an additional limitation of the look-ahead range for the pre-output of M functions (see [FCT-C1]) to a maximum distance.<br>If this maximum distance exceeds the sum of all currently considered motion blocks (except for the ‘oldest’ motion block), the ‘oldest’ motion block is output. In other words, an M function can be pre-output by at least the specified distance. |
| Parameter | configuration.path_preparation.m_pre_output_max_distance |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 0 |

| | |
|---|---|
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>If the maximum number of blocks P-CHAN-00603 [▶ 57] is set to a high value, it may cause a long delay in channel reaction. To avoid this, a distance limit can also be specified. With long motion blocks in particular, this maximum distance is already reached after a few blocks. This prevents additional delay caused by saving motion blocks in the pre-output of M functions.<br>Without an explicit specification, the range is not additionally limited (only by the number of blocks P-CHAN-00603 [▶ 57]).<br>If a pre-output is set greater than the distance currently saved in the look-ahead range, the M function is pre-output at the maximum known path position and a warning is issued.<br>**Parameterisation example:**<br>*configuration.path_preparation.function FCT_DEFAULT* \| *FCT_M_PRE_OUTPUT*<br>*configuration.path_preparation.m_pre_output_lookahead 100*<br>*configuration.path_preparation.m_pre_output_max_distance 35000 [0.1µm]* |

Maximum distance for pre-output of M functions

```
%microjoint62
N01 G00 G90 X0 Y0
N02 G01 F10000

'MOS' = '1'

N01 V.G.M_FCT[100].PRE_OUTP_PATH = 28.6 (* in mm *)
N02 V.G.M_FCT[100].SYNCH = 'MOS'

N20 G91 Y1
N21 Y1 ; -> MicroJoint at Y1.4 mm
…
N43 Y1
N44 Y1
N45 Y1
; Warning 120693: -> MicroJoint due to distance limitation 3.5mm
N46 Y1
N47 Y1
N48 Y1
N49 Y1
N50 M100
N99 M30
```

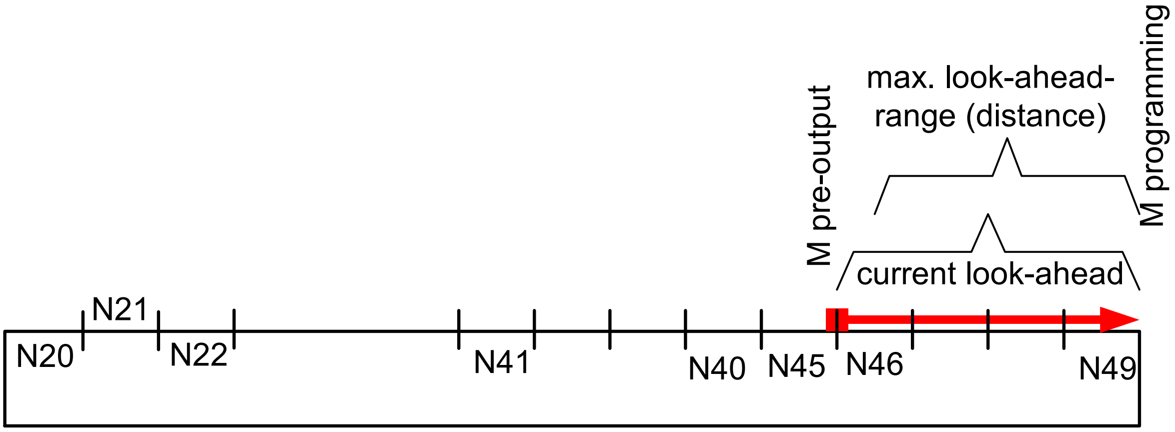


Fig. 2: Distance-related limiting of pre-output to maximum look-ahead range.

## 1.1.3.6 Conditional activation of the functionalities (P-CHAN-00605)

| **P-CHAN-00605** | **Conditional enable of the functionalities** |
|---|---|
| Description | This parameter can be used to configure functions of the CNC that are activated in the channel depending on the program start mode.<br><br>Several settings can be predefined for activation as soon as an NC program is started in the machining mode listed under the condition P-CHAN-00606 [▶ 61] . |
| Parameter | configuration.path_preparation.fct_enable[<idx>] where idx 0, 1 |
| Data type | STRING |
| Data range | See Path preparation function table [▶ 55] |
| Dimension | ---- |
| Default value | FCT_DEFAULT |

| | |
|---|---|
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>If no individual machining mode is set at the start of the program, this corresponds to the setting ISG_STANDARD.<br>The default settings under index 0 ("fct_enable[0]") are used.<br>For reasons of backward compatibility, P-CHAN-00605 "fct_enable[0]" corresponds to the previous parameter "function" P-CHAN-00600 [▶ 54].<br>Both notations or parameters can be used alternatively. |

### 1.1.3.7 Condition for activating the functionalities (P-CHAN-00606)

| **P-CHAN-00606** | **Condition for activating the functionalities** |
|---|---|
| Description | This parameter specifies a condition.<br>If the processing mode complies with the specified condition at program startn, the related functionalities are activated at program start.<br>The corresponding functionalities are defined in P-CHAN-00605 [▶ 60] .<br>The condition is assigned to the corresponding functionality by an index. |
| Parameter | configuration.path_preparation.fct_condition[<idx>] where idx 0,1 |
| Data type | STRING |
| Data range | See Condition for activating the functionalities (P-CHAN-00508) [▶ 43] |
| Dimension | ---- |
| Default value | ISG_STANDARD |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

## 1.1.4 Interpolation

### 1.1.4.1 Defining interpolator functionalities (P-CHAN-00650)

| **P-CHAN-00650** | **Definition of interpolator functionalities** |
|---|---|
| Description | This parameter defines individual functionalities and the size of the look-ahead buffer in the interpolator, i.e. it defines the number of blocks required to calculate deceleration distance and dynamic planning. |
| Parameter | configuration.interpolator.function |
| Data type | STRING |
| Data range | See Defining interpolator functionalities (P-CHAN-00650) [▶ 62] |
| Dimension | ---- |
| Default value | FCT_IPO_DEFAULT |
| Remarks | |

**Interpolation function table**

| Identifier | Description |
|---|---|
| FCT_IPO_DEFAULT | FCT_LOOK_AHEAD_STANDARD |
| FCT_LOOK_AHEAD_LOW | 30 blocks |
| FCT_LOOK_AHEAD_STANDARD | 120 blocks |
| FCT_LOOK_AHEAD_HIGH | 190 blocks |
| FCT_LOOK_AHEAD_CUSTOM | Any number of look-ahead blocks in the interval [ 0; 200]. Specification by parameter P-CHAN-00653 [▶ 63]. |
| FCT_SYNC | Synchronisation of an axis on a path group.<br>Example: FCT_IPO_DEFAULT \| FCT_SYNC |
| FCT_LOOK_AHEAD_OPT | The path velocity curve can be further improved for HSC machining by additional calculations. This generally reduces machining time. The additional calculations place greater demands on the controller hardware. |
| FCT_LIFT_UP_TIME | Automatic lifting/lowering of an axis (time-based coupling).<br>Example: FCT_IPO_DEFAULT \| FCT_LIFT_UP_TIME |
| FCT_SHIFT_NCBL | Path-controlled offset of M functions (dwell time).<br>Example: FCT_IPO_DEFAULT \| FCT_SHIFT_NCBL |
| FCT_CALC_STATE_AT_T | Calculation of path velocity at a time in the future. Function only available in combination with HSC slope and only as of V3.1.3057.0<br>Example: FCT_IPO_DEFAULT<br>\| FCT_CALC_STATE_AT_T |
| FCT_CALC_TIME | Calculation of interpolation time to next feed block (G01,G02,G03).<br>Example: FCT_IPO_DEFAULT \| FCT_CALC_TIME |
| FCT_CONTOUR_LAH | Contour look-ahead: advance output of motion blocks to the PLC as of V3.1.3104.07 |
| FCT_DYN_POS_LIMIT | Dynamic limitation of axis positions |

The look-ahead buffer size specified above applies as of CNC Build V2.11.2800 and higher. The following values apply as of CNC Build V2.11.20xx:

| | |
|---|---|
| FCT_LOOK_AHEAD_LOW | 30 blocks |
| FCT_LOOK_AHEAD_STANDARD | 70 blocks |
| FCT_LOOK_AHEAD_HIGH | 120 blocks |

### 1.1.4.2 Maximum number of logged events (P-CHAN-00651)

| P-CHAN-00651 | Maximum number of entries in the history buffer |
|---|---|
| Description | The CNC offers the options of filing events in a history memory (logging entries). This parameter defines the maximum number of logged events. If more entries occur, the oldest entry is overwritten. |
| Parameter | configuration.interpolator.log_entry_number |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 40 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.4.3 Defining the type of logged events (P-CHAN-00652)

| P-CHAN-00652 | Defining the type of logged events |
|---|---|

| | |
|---|---|
| Description | The CNC offers the options of filing events in a history memory (logging entries). The parameter permits the user-specific definition of the CNC logging entries to be logged. Depending on troubleshooting or the analysis requirement, event logging can be filtered in order to reduce the number of entries to be logged or analysed right from the outset. |
| Parameter | configuration.interpolator.log_level |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.4.4 User-specific size of look-ahead buffer (P-CHAN-00653)

| **P-CHAN-00653** | **User-specific size of look-ahead buffer** |
|---|---|
| Description | This parameter permits the user-defined definition of the number of NC blocks in the look-ahead buffer.<br>This parameter is used only if P-CHAN-00650 [▶ 61] Is set with FCT_LOOK_AHEAD_CUSTOM. |
| Parameter | configuration.interpolator.number_blocks_lah * |
| Data type | UNS32 |
| Data range | 0 ... 10000 |
| Dimension | ---- |
| Default value | 120 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>As of Build V2.11.20 and higher, the default size of the look-ahead buffer is 70 blocks. As of Build V2.11.28 and higher, the default size is 120 blocks. As the size increases, the additional calculations make greater demands on the controller hardware.<br>As of Build V3.1.3067.07 the upper limit of the data range is 500 blocks.<br>If #SLOPE[TYPE=STEP] is used, the upper limit is 10000 blocks as of Build V3.1.3060.0.<br>* P-CHAN-00653 can be used in V2.11.20 and higher with configuration.interpolator.parameter for compatibility reasons. |

### 1.1.4.5 Reducing interpolator computing time (P-CHAN-00654)

| **P-CHAN-00654** | **Definition of interpolator functionalities** |
|---|---|
| Description | The mircroprocessor load can be limited by specifying the number of blocks per cycle considered in the look ahead process. Calculating the look ahead profile is then split into partial calculations over several cycles.<br>**Example:** number_blocks_lah = 10000, blocks_per_call = 1000<br>The look ahead profile is then calculated split over 10 cycles. One disadvantage of this is the acceptance of real-time influences delayed by this time, e.g. an override change. Therefore, do not select a value that is too low. |
| Parameter | configuration.interpolator.blocks_per_call |
| Data type | UNS32 |

| | |
|---|---|
| Data range | 1 ... The value is defined by P-CHAN-00650 [▶ 61] . |
| Dimension | ---- |
| Default value | 200 |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.4.6 Conditional activation of the functionalities (P-CHAN-00655)

| **P-CHAN-00655** | **Conditional enable of the functionalities** |
|---|---|
| Description | This parameter can be used to configure functions of the CNC that are activated in the channel depending on the program start mode.<br>Several settings can be predefined for activation as soon as an NC program is started in the machining mode listed under the condition P-CHAN-00656 [▶ 64] . |
| Parameter | configuration.interpolator.fct_enable[<idx>] where idx 0, 1 |
| Data type | STRING |
| Data range | See Defining interpolator functionalities (P-CHAN-00650) [▶ 62]. |
| Dimension | ---- |
| Default value | FCT_IPO_DEFAULT |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10<br>If no individual machining mode is set at the start of the program, this corresponds to the setting ISG_STANDARD.<br>The default settings under index 0 ("fct_enable[0]") are used.<br>For reasons of backward compatibility, P-CHAN-00655 "fct_enable[0]" corresponds to the previous parameter "function" P-CHAN-00650 [▶ 61].<br>Both notations or parameters can be used alternatively. |

### 1.1.4.7 Condition for activating the functionalities (P-CHAN-00656)

| **P-CHAN-00656** | Condition for enabling functionalities |
|---|---|
| Description | This parameter specifies a condition.<br>If the machining mode corresponds to the specified condition at program start, the associated functionalities are activated at program start.<br>The corresponding functionalities are defined by P-CHAN-00650 [▶ 61] .<br>The condition for the corresponding functionality is assigned by the index: |
| Parameter | configuration.interpolator.fct_condition[<idx>] where idx 0,1 |
| Data type | STRING |
| Data range | See Condition for activating the functionalities (P-CHAN-00508) [▶ 43] |
| Dimension | ---- |
| Default value | ISG_STANDARD |
| Remarks | Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.4.8 Number of logs of the dynamic coordinate system (P-CHAN-00657)

| **P-CHAN-00657** | **Number of logged input and output values of the dynamic CS** |
|---|---|

| | |
|---|---|
| Description | When the dynamic coordinate system is calculated, the input and output values and the current dynamic coordinate system can also be logged for diagnostic purposes. Logged data is loaded from the controller when diagnostic data is uploaded and written to a file. |
| Parameter | configuration.interpolator.dyn_cs_history_max |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 20 |
| Remarks | Parameter is available as of the following Builds: V3.1.3079.17 ; V3.1.3107.10 |

### 1.1.4.9 Maximum number of contour elements in the look-ahead contour (P-CHAN-00658)

| **P-CHAN-00658** | **Maximum number of logged contour elements in the contour look-ahead.** |
|---|---|
| Description | This parameter can be used to set the maximum number of stored motion blocks that can be supplied to the PLC in advance.<br>The CNC command #CONTOUR LOOKAHEAD LOG [] can be used to activate the save function.<br>FCT_CONTOUR_LAH must be enabled in P-CHAN-00650 [▶ 61] for this functionality.<br>`configuration.interpolator.fct_enable[0]`<br>`FCT_IPO_DEFAULT\|`**`FCT_CONTOUR_LAH`** |
| Parameter | configuration.interpolator.contour_lookahead_log_max |
| Data type | UNS32 |
| Data range | 0 <= contour_lookahead_log_max < MAX_UNS32 |
| Dimension | ---- |
| Default value | 128 |
| Remarks | Parameter available as of V3.1.3107.10 |

# 1.2 Configuration of PLC functions

## 1.2.1 Settings for M functions

### 1.2.1.1 Synchronisation types of M functions (P-CHAN-00041)

| **P-CHAN-00041** | **Synchronisation type of M functions** |
|---|---|
| Description | In the array m_synch[i], the synchronization type of the corresponding M function is defined. Here, the field index 'i' defines the number of the M function. The value indicates the synchronisation type of the M function, i.e. when a check is made for presence of a PLC acknowledgement. A motion is not executed, or is stopped at the latest towards the end of the block, if no acknowledgement has arrived from the PLC. The synchronisation type is defined as a string constant or a hexadecimal value. |
| Parameter | m_synch[i] where i = 0 999 (maximum number of M functions, application-specific) |
| Data type | STRING |
| Data range | See the table below. |
| Dimension | ---- |
| Default value | NOT_VALID * |

| | |
|---|---|
| Remarks | M functions are consumption information and must be fetched (read) by the PLC. This also applies to M functions of the type MOS, MOS_TS, MEP_MOS and MET_MOS. Otherwise, this results in a blocked interface to the HLI in the CNC and as a consequence to an unexpected processing stop.<br><br>***Note:**<br>**The default value for internal M functions (M0, M1, M2, M17, M29, M30, M3, M4, M19) is NO_SYNCH.**<br><br>**Caution:**<br>The following applies to synchronisation types with associated time and path-related pre-output (MET_SVS, MET_MOS, MEP_SVS, MEP_MOS):<br>If one of these synchronisation types is later changed into one which requires no pre-output value, P-CHAN-00070 [▶ 81] (m_pre_outp[i]) must be assigned to 0. Otherwise, a license error is generated in case of microjoints if this function is not licensed or not enabled (seeP-CHAN-00600 [▶ 54] Alternatively: P-STUP-00060) is inactive.<br><br>Example:<br>m_synch[12] MVS_SVS 0x00000002<br><br>Note: Programming a UNS32 variable is permissible for downward compatibility reasons.<br>Example: m_synch[12] 0x00000002 |

| **Constant** | **Value** | **Meaning** |
|---|---|---|
| NOT_VAILD | -1 | No valid M function |
| NO_SYNCH | 0x00000000 | No output of M function to PLC |
| MOS | 0x00000001 | Output of M function to PLC without synchronisation. If the M function is programmed within a motion block, the output of the M function is executed before the movement. M function must be fetched from PLC! |
| MVS_SVS | 0x00000002 | Output of the M function to the PLC before the motion block, synchronisation before the motion block |
| MVS_SNS | 0x00000004 | Output of the M function to the PLC before the motion block, synchronisation after the motion block |
| MNS_SNS | 0x00000008 | Output of the M function to the PLC after the motion block, synchronisation after the motion block |
| MNE_SNS | 0x00000020 | Output of M function to PLC after measurement event and removal of distance to go,<br>synchronisation after motion block (for edge banding option only) |
| MVS_SLM | 0x00004000 | Late synchronisation, output of M function to PLC within the block, synchronisation during transition to G01/G02/G03 (implicit synchronisation) |
| MVS_SLP | 0x00008000 | Late synchronisation, output of M function to PLC within the block, synchronisation by NC command #EXPL SYN (explicit synchronisation) |
| MOS_TS | 0x00040000 | Output of the M function to the PLC before motion block without synchronisation,<br>CNC calculates sampling time offset for high-precision time output in the PLC.<br>M function must be fetched from PLC. |
| MEP_MOS | 0x00100000 | Pre-output of M function with specified path, without synchronisation.<br>M function must be fetched from PLC. |

| | | |
|---|---|---|
| MET_MOS | 0x00200000 | Pre-output of M function with specified time, without synchronisation.<br>M function must be fetched from PLC. |
| BWD_SYNCH | 0x00400000 | Synchronisation of M function during backward motion with MVS_SVS |
| FWD_SYNCH | 0x00800000 | Synchronisation of M function during 'Simulated forward motion' based on the defined synchronisation type |
| MEP_SVS | 0x01000000 | Output of M function with specified path, synchronisation before next block |
| MET_SVS | 0x02000000 | Pre-output of M function with specified time, synchronisation before next block |
| FAW_SYNCH | 0x10000000 | Decoding stop (Flush and Wait): Output of M function to PLC and stop of program decoding at block end until program run is completed. FAW_SYNCH can be set in addition to other synchronisation types. M functions with FAW_SYNCH may not be used when tool radius compensation (TRC), polynomial contouring and HSC mode are active. |

### 1.2.1.1.1 Examples of synchronisation of M functions

#### Synchronisation type MOS (without synchronisation)

```
N20 G00 X25
N30 X50
N40 X75 M25 ( M25 of type MOS )
N50 G01 X100 F2000
N60 X125 Z100
M30
```

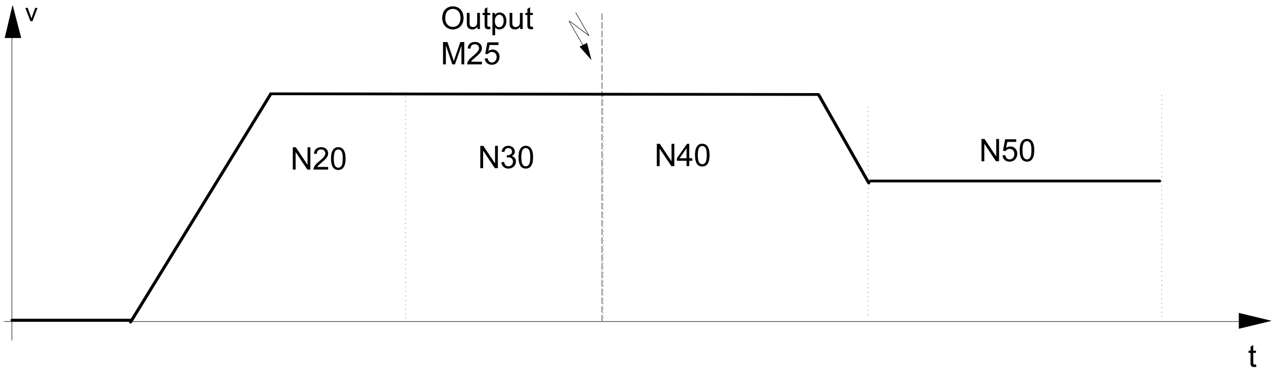


Fig. 3: Synchronisation type MOS (without synchronisation)

### Synchronisation type MVS_SVS

```
N20 G00 X25
N30 X50
N40 X75 M25 ( M25 of the MVS_SVS type )
N50 G01 X100 F2000
N60 X125 Z100
M30
```

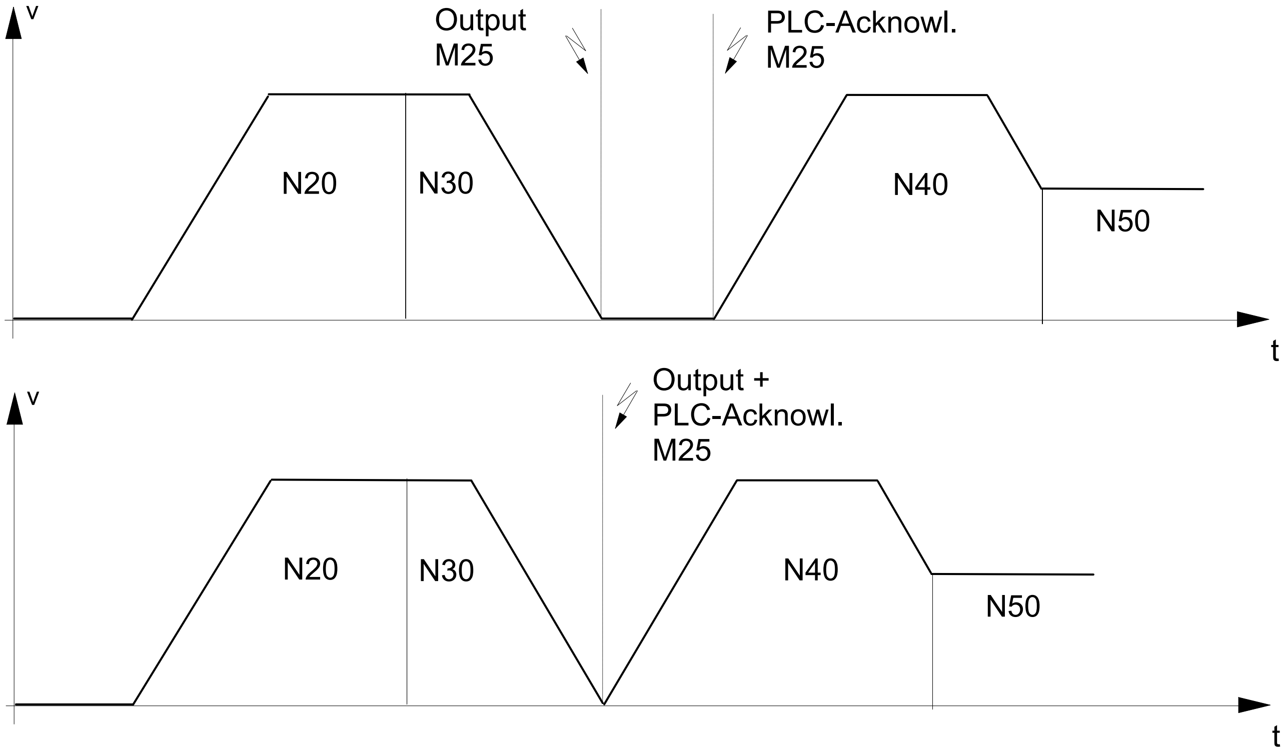


Fig. 4: Synchronisation type MVS_SVS

### Synchronisation type MVS_SNS

```
N20 G00 G90 X25
N30 X50
N40 X75 M25 ( M25 of the MVS_SNS type )
N50 G01 X100 F2000
N60 X125 Z100
M30
```

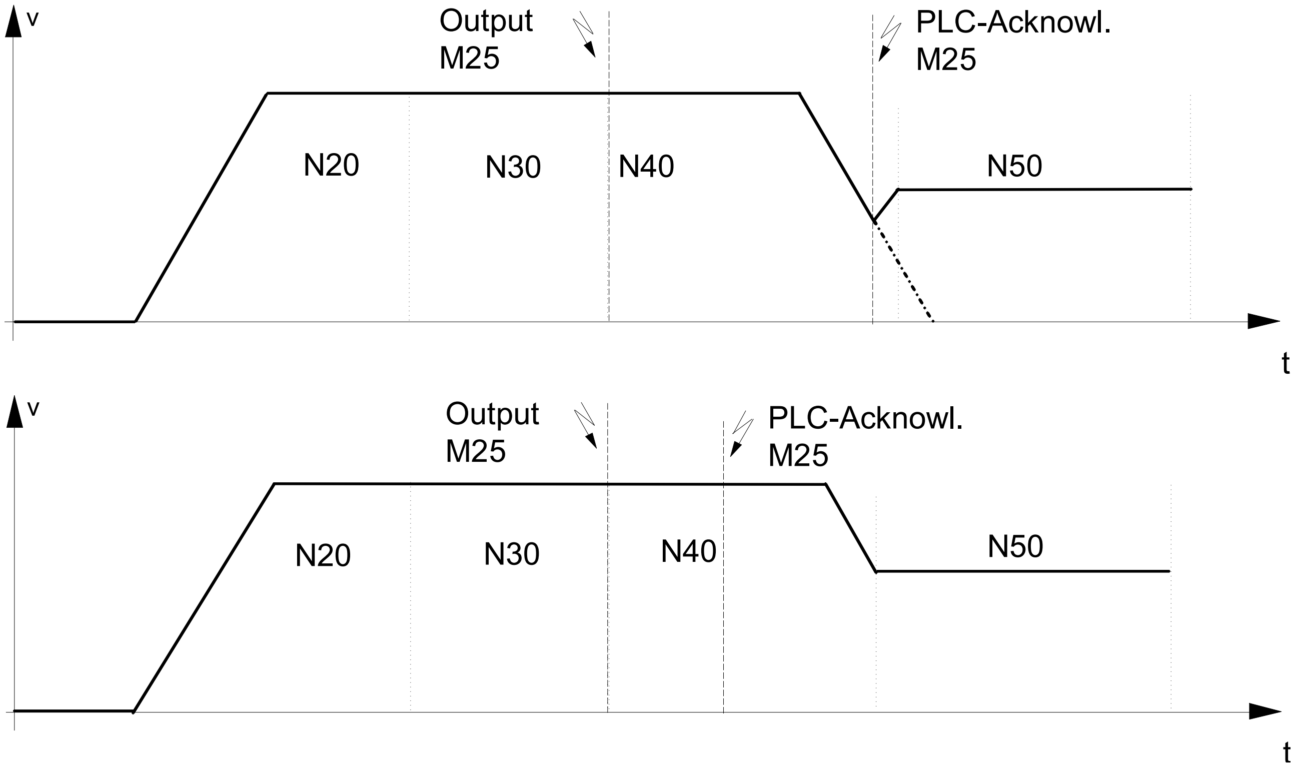


Fig. 5: Synchronisation type MVS_SNS

### Synchronisation type MNS_SNS

```
N20 G00 X25
N30 X50
N40 X75 M25 ( M25 of the MNS_SNS type )
N50 G01 X100 F2000
N60 X125 Z100
M30
```

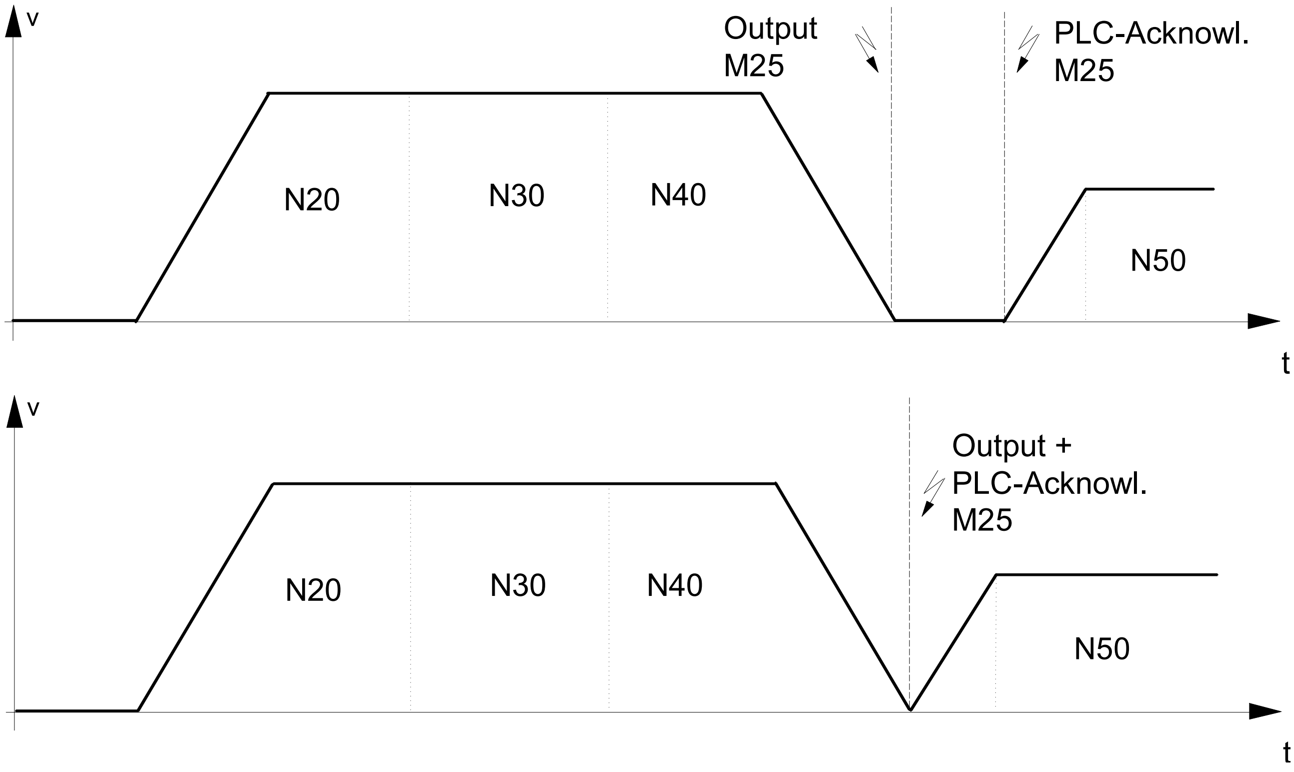


Fig. 6: Synchronisation type MNS_SNS

### Synchronisation type MNE_SNS (only for edge banding option)

```
N05 X0 Y0
N10 G108 (start measurement of edge banding)
N20 G01 X90 Y90 F20
N30 G01 X150 Y150 M33 F8 (M33 of type MNE_SNS)
N40 G107 (end of measurement of edge banding)
N50 G00 X200 Y200
M30
```

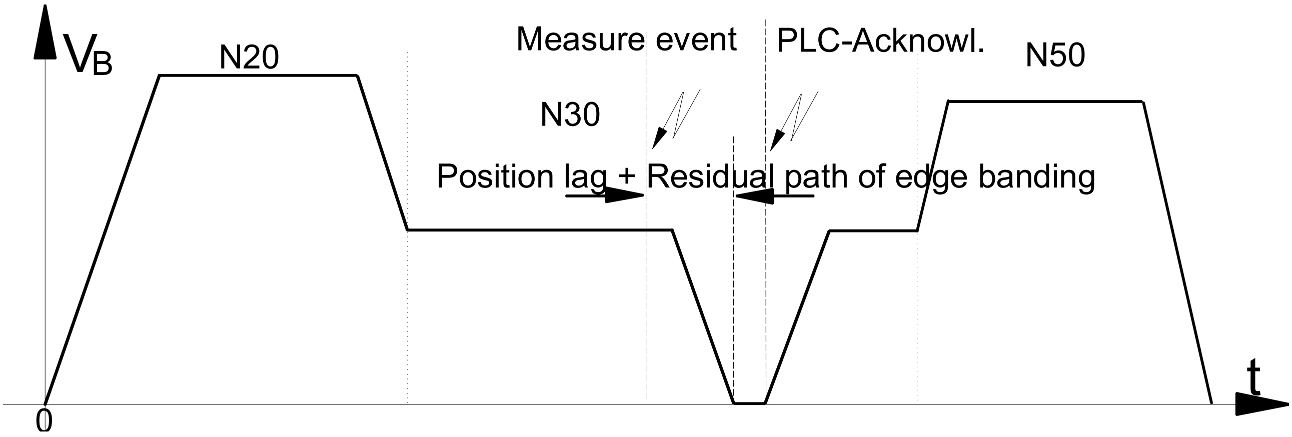


Fig. 7: Synchronisation type MNE_SNS (only for edge banding option)

### Synchronisation type MVS_SLM

```
N05 M24 (M24, synchronisation type MVS_SLM)
N10 M25 G00 X25 (M25, synchronisation type MVS_SLM)
N20 X50
N30 X75
N40 X100
N50 G01 X125 F2000 <-- trigger for M24, M25 before execution of motion block
N60 Z100
M30
```

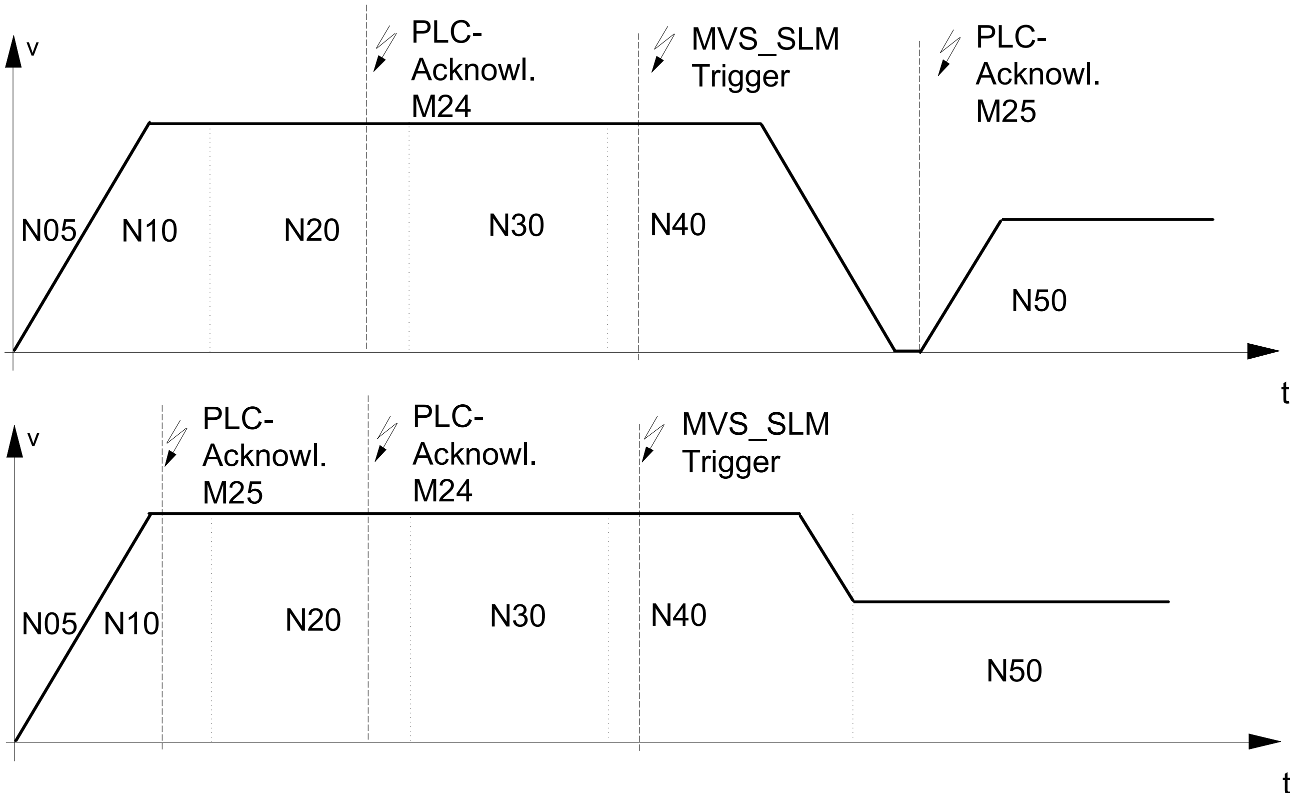


Fig. 8: Synchronisation type MVS_SLM

## Synchronisation type MVS_SLP

```
N05 M26 G00 X25 (M26, synchronisation type MVS_SLP)
N10 M27 (M27, synchronisation type MVS_SLP)
N20 X50
N30 X75
N40 X100
N50 G01 X125 F2000
N60 #EXPL SYN <-- trigger for M26, M27 before execution of next motion block
N70 G00 X0
N80 X0 Y0
M30
```

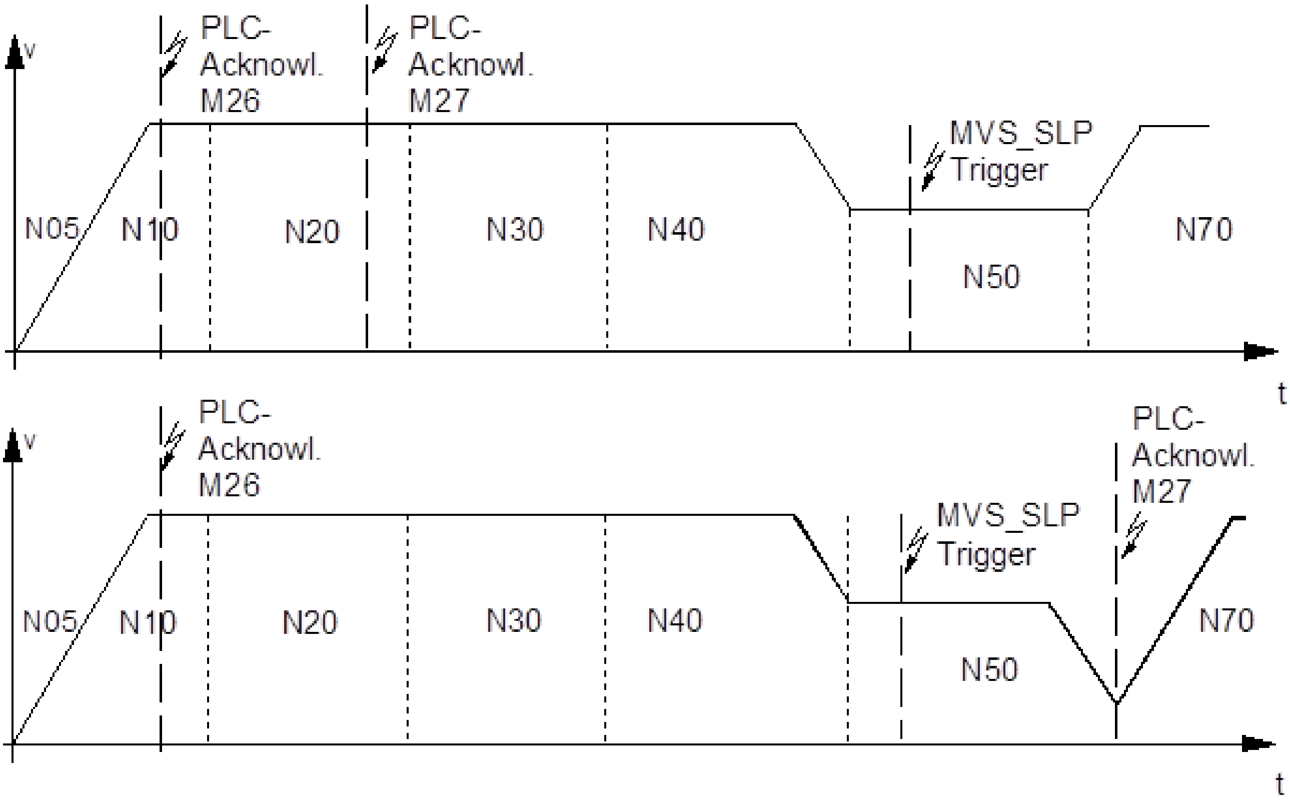


Fig. 9: Synchronisation type MVS_SLP

### Without synchronisation MOS_TS

```
N10 G01 X25 G90 F5000
N20 X50
N30 M25 (M25 MOS_TS)
N40 X100
N50 X200
M30
```

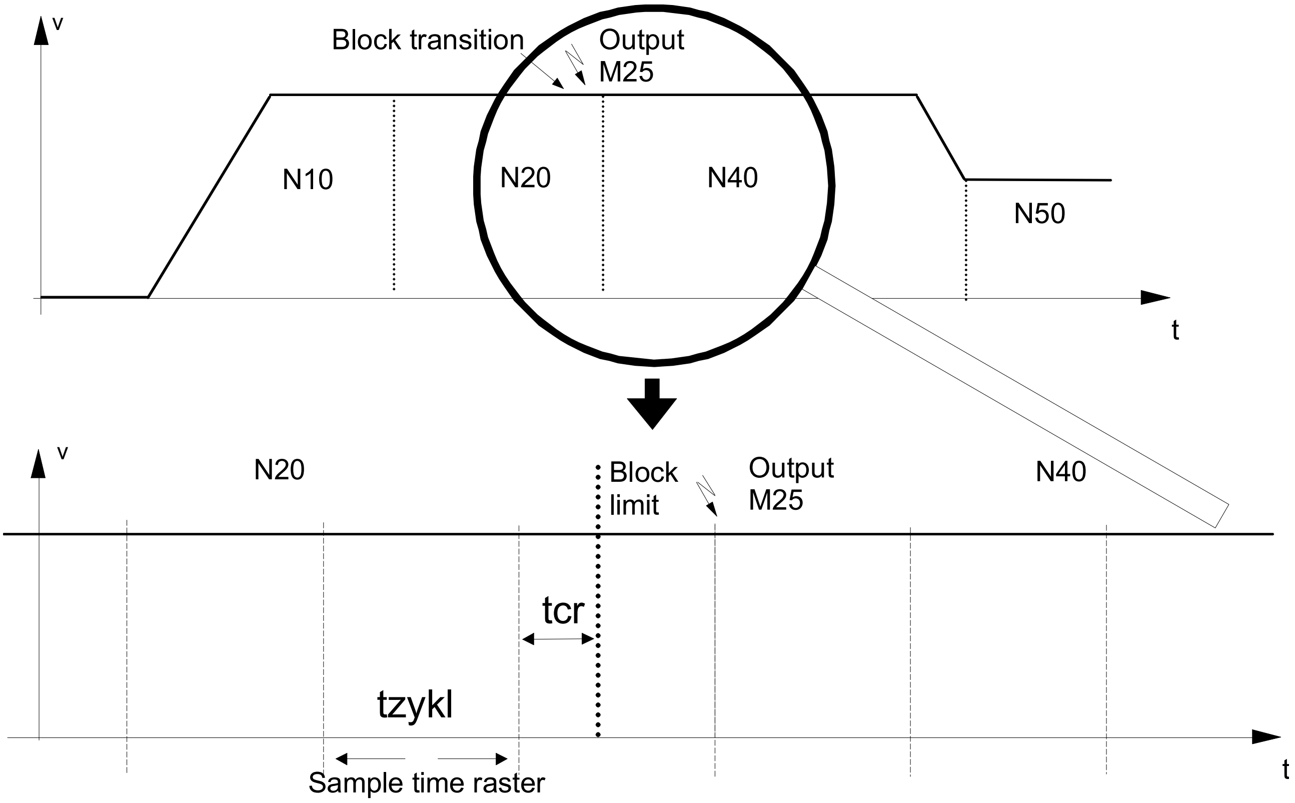


Fig. 10: Without synchronisation MOS_TS

**NOTICE**

The PLC interface must be cyclically read for the correct entry of the time offset. Only this ensures the exact output of following M functions of type MOS_TS and a correct assignment within the time-based process.

**NOTICE**

The sampling time offset, calculated by the CNC, is not passed to the HLI if the synchronisation type MOS_TS is used in axis-specific M function programming (see example).

```
N10 G01 X25 G90 F5000
N20 X50
N30 X[M25] → axis-specific output M25 (MOS_TS)
N40 X100
N50 X200
M30
```

### Synchronisation types MET_SVS, MEP_SVS

Synchronisation types with synchronisation before block with time or path-related pre-output. The associated pre-output values are set in P-CHAN-00070 [▸ 81] (m_pre_outp[i]):

```
N10 G01 X10 G90 F5000
N20 X20
N30 X30
N40 X40
N50 M96 (M96 MEP_SVS m_pre_outp = 250000)
(or MET_MOS m_pre_outp = 300000µs)
N55 X80
N60 X0
M30
```

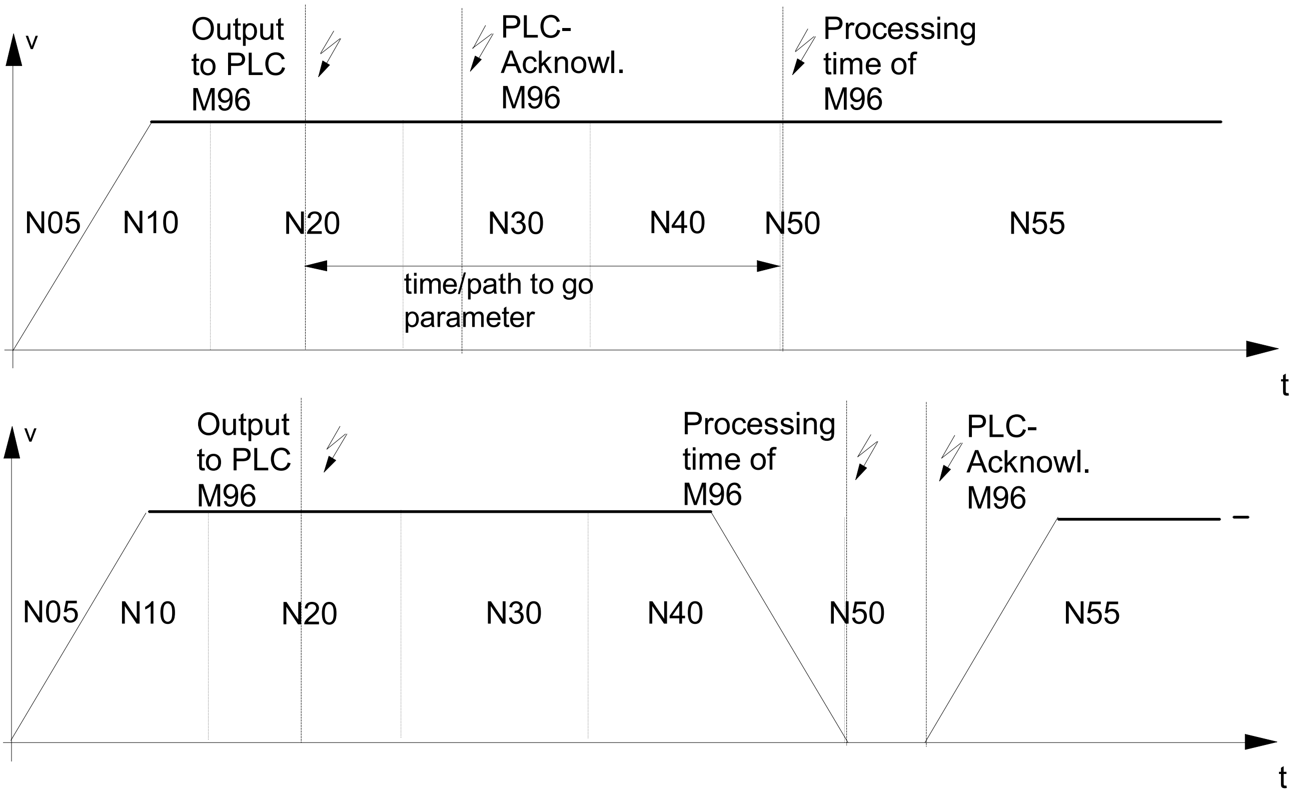


Fig. 11: Synchronisation types MET_SVS, MEP_SVS

If several M functions of this synchronisation type are programmed overlapped in relation to their effective ranges and positions in the NC program, the first programmed M function in the NC block sequence defines the earliest time of output for all following M functions (see examples below).

This may cause an offset or change in order of the output times. Therefore, avoid range overlaps as far as possible.

**NOTICE**

Please note with MET_SVS codes that the pre-output time is planned due to the required synchronisation with path motions based on a feed profile with end velocity 0. As a result, there may be deviations between the planned and the actual motion times up to the M code (block limit). This is shown in the figure below.

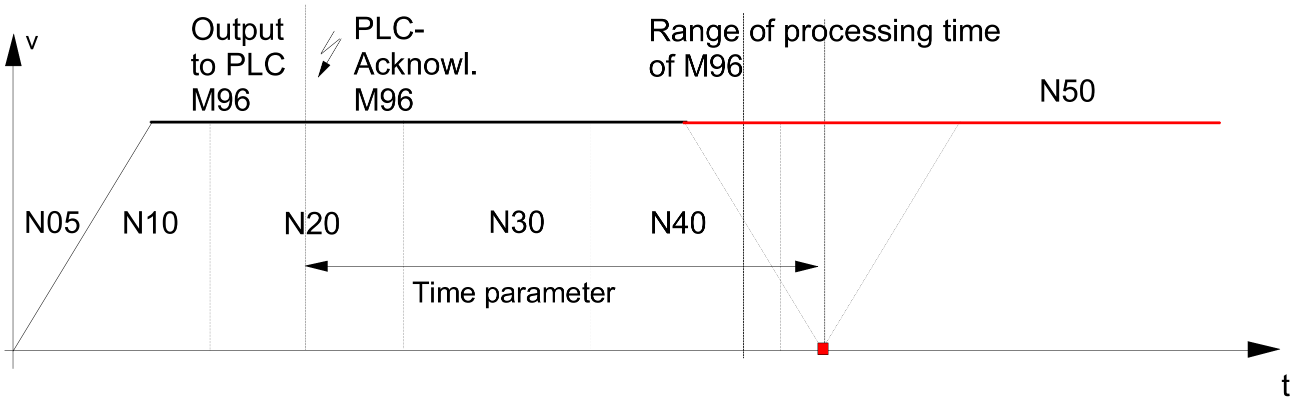
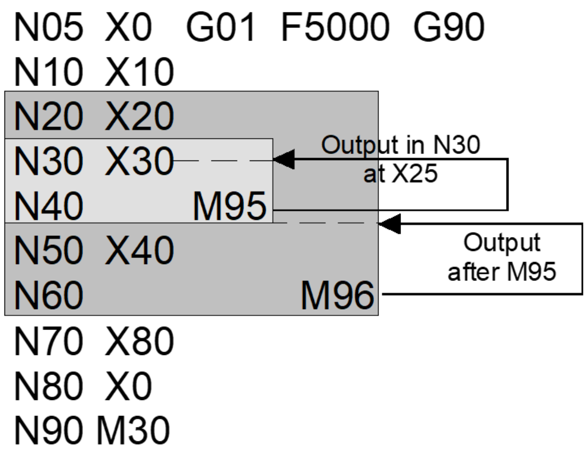


Fig. 12: Planned output time with MET_SVS

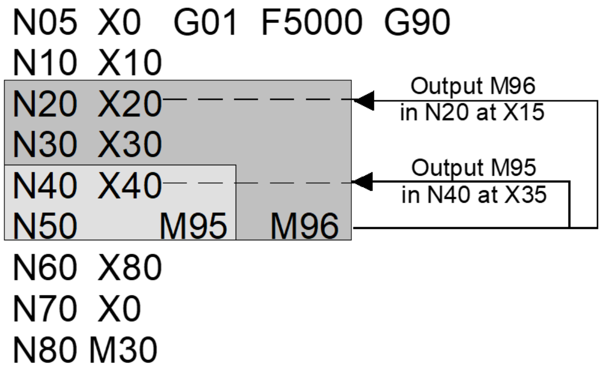
### Range overlap and output with MEP_SVS, MET_SVS

```
M functions in different NC blocks, range overlap causes changed output times:

(M95 MEP_SVS m_pre_outp = 50000)
(M96 MEP_SVS m_pre_outp = 250000)
```



```
N05  X0   G01  F5000  G90
N10  X10
N20  X20
N30  X30
N40           M95
N50  X40
N60                M96
N70  X80
N80  X0
N90 M30
```

```
M functions in the same NC block, range overlap has no effect on output times:


(M95 MEP_SVS m_pre_outp = 50000)
(M96 MEP_SVS m_pre_outp = 250000)
```



```
N05  X0   G01  F5000  G90
N10  X10
N20  X20
N30  X30
N40  X40
N50       M95    M96
N60  X80
N70  X0
N80 M30
```

## Synchronisation types MET_MOS, MEP_MOS

Synchronisation types without synchronisation with time or path-related pre-output. The associated pre-output values are set in P-CHAN-00070 [▶ 81] (m_pre_outp[i]):

```
N10 G01 X10 G90 F5000
N20 X20
N30 X30
N40 X40
N50 M96 (M96 MEP_MOS m_pre_outp = 250000)
(or MET_MOS m_pre_outp = 300000µs)
N55 X80
N60 X0
M30
```

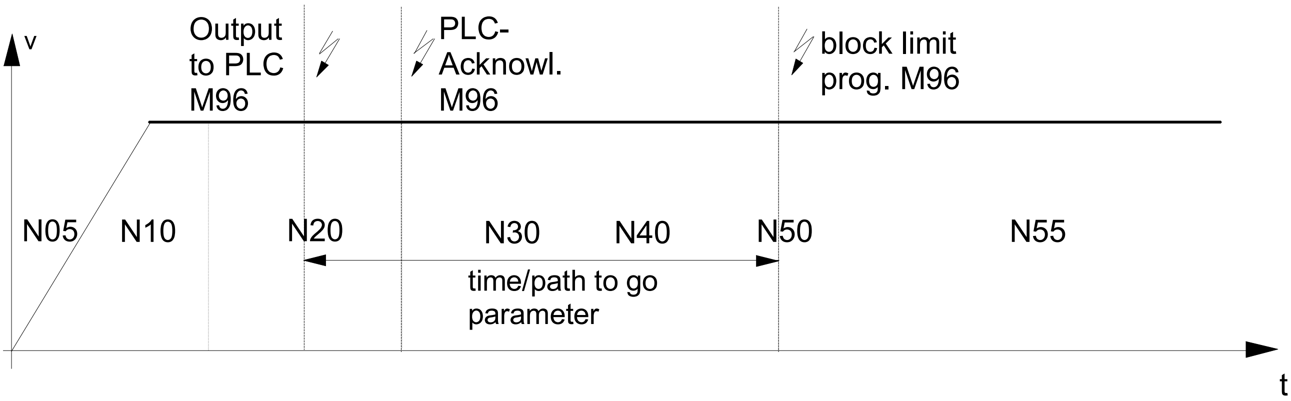


Fig. 13: Synchronisation types MET_MOS, MEP_MOS

MEP_MOS and MET_MOS require no acknowledgement; they must still be deleted in the PLC.

The same constraints apply to overlap programming as for synchronisation types MEP_SVS and MET_SVS.

### Synchronisation types MOS, MOS_TS, MVS_SVS, MVS_SNS, MNS_SNS in combination with the 'microjoints' function

See Pre-output of M functions (MicroJoint)

Use of this function requires a license for the “Cutting” option. It is not included in the scope of the standard license.

In addition, a path-related pre-output can be specified for synchronisation types MOS, MOS_TS, MVS_SVS, MVS_SNS and MNS_SNS. This path can be defined in the channel parameter list or in the NC program.

In this case the the M function is output by the specified distance relative to the current block start. M functions brought forward are then treated as if they were programmed without any motion.

With synchronisation types MVS_SVS, MVS_SNS and MNS_SNS an acknowledgement is also expected at this point (MOS and MOS_TS require no acknowledgement but they must still be deleted in the PLC).

With synchronisation type MCS_SNS: if it is necessary to split blocks due to the pre-output since the pre-output fails to occur at an existing block limit, the split block is synchronised at the end and not at the output point.

```
N01 V.G.M_FCT[11].PRE_OUTP_PATH = 25 (* in [mm] *)
N02 V.G.M_FCT[11].SYNCH = 2 (* MVS_SVS *)
N10 G01 X10 G90 F5000
N20 X20
N30 X30
N40 X40
N50 M11 X80 (M11 as MVS_SVS m_pre_outp = 250000, X40 -25 = X15)
N60 X0
M30


The above example has the same effect as the following programming of M11 without a pre-output:

N01 V.G.M_FCT[11].PRE_OUTP_PATH = 0 (* in [mm] *)
N02 V.G.M_FCT[11].SYNCH = 2 (* MVS_SVS *)
N10 G01 X10 G90 F5000
N20 X15
N20 M11 (M11 output as MVS_SVS)
N20 X20
N30 X30
N40 X40
N50 X80
N60 X0
M30
```

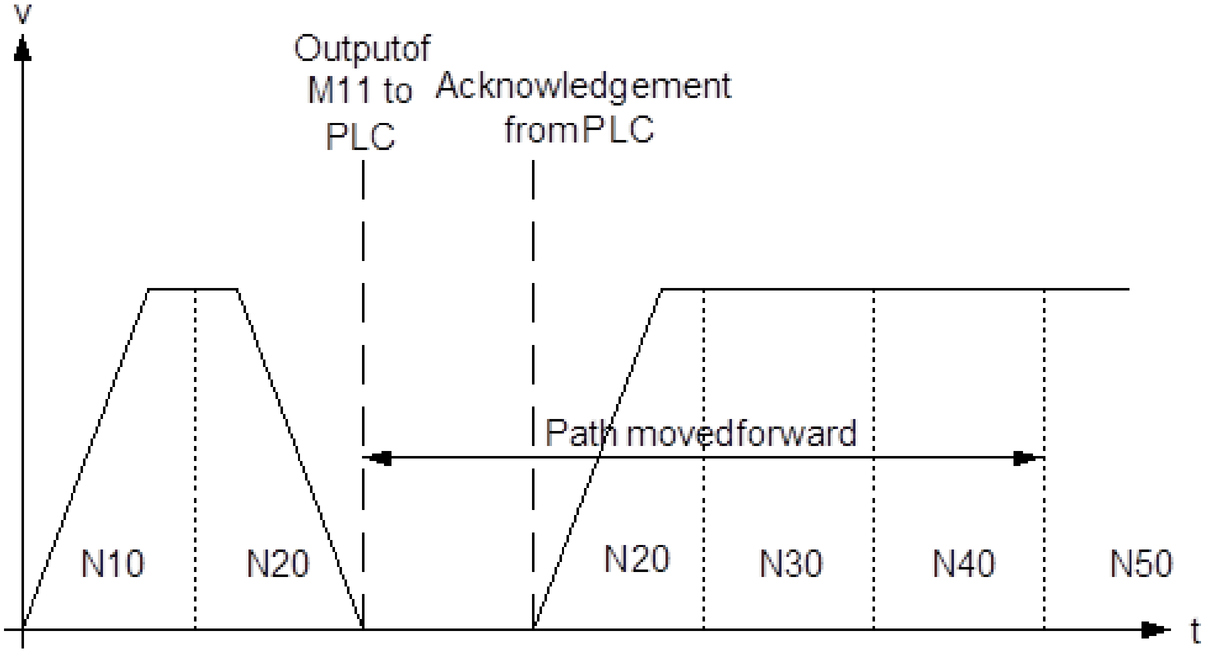


Fig. 14: Pre-output and acknowledgement of an M function with microjoints

This functionality is available as of Build **V2.10.1507.05**.
The single NC channel is enabled by P-CHAN-00600 [▸ 54] (alternatively P-STUP-00060).

```
configuration.path_preparation.function FCT_DEFAULT | FCT_M_PRE_OUTPUT   ( P-CHAN-00600 )
```

### 1.2.1.2 Path or time-related pre-output of M functions (P-CHAN-00070)

| **P-CHAN-00070** | **Path or time-related pre-output of M functions** |
|---|---|
| Description | This parameter is used in connection with M functions<br>• of synchronisation types MET_SVS, MET_MOS and MEP_SVS, MEP_MOS. The lead time is specified for MET_SVS, MET_MOS and the lead distance for the MEP_SVS or MEP_MOS type.<br>• of synchronisation types MOS, MVS_SVS, MVS_SNS, MNS_SNS, MOS_TS and the ‘microjoints’ function which requires a separate license (see [FCT-C1]). Path-related parameters make sense here.<br>The field index 'i' defines the number of associated M functions. The value of m_pre_outp[i] defines the path and time-related output point before actual processing of the M function on the path. |
| Parameter | m_pre_outp[i] where i = 0 999 (maximum number of M functions, application-specific) |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | 0.1 µm or µs |
| Default value | 0 |
| Remarks | The pre-output value can also be defined in the NC program [PROG//section V.G. variables].<br>**Caution:**<br>If one of these synchronisation types is later changed into one which requires no pre-output value, P-CHAN-00070 (m_pre_outp[i]) must be assigned to 0. Otherwise, a license error is generated in case of microjoints if this function is not licensed or not enabled (see P-CHAN-00600 [▸ 54] alternatively P-STUP-00060).<br>**Parameterisation example:** |

| | |
|---|---|
| | The user-specific M functions M96 and M98 must be output to the SPS 10 mm before reaching the synchronisation position in the block sequence.<br>The user-specific M functions M97 and M99 to SPS must be output 40 milliseconds before reaching the time of synchronisation in the block sequence.<br><br>*# Definition of M functions and synchronisation types*<br>*# ==============================================*<br>*m_synch[96]            0x01000000   MEP_SVS*<br>*m_synch[97]            0x02000000   MET_SVS*<br>*m_synch[98]            0x00100000   MEP_MOS*<br>*m_synch[99]            0x00200000   MET_MOS*<br>*#*<br>*# Definition of pre-output path, pre-output time*<br>*#========================================*<br>*m_pre_outp[96]         100000   in 0.1µm*<br>*m_pre_outp[97]          40000   in µs*<br>*m_pre_outp[98]         100000   in 0.1µm*<br>*m_pre_outp[99]          40000   in µs* |

### 1.2.1.3 Axis-specific M functions (P-CHAN-00039)

| **P-CHAN-00039** | **Axis-specific M functions** |
|---|---|
| Description | User-specific M functions programmed in DIN syntax are processed and executed channel-specific.<br>If the user forces axis-specific processing for specific M functions, it is possible to configure them using this parameter so that they have an axis-specific effect. An axis name can be assigned to each M function on which it should act. Both path axes and spindle axes are permissible. |
| Parameter | m_default_outp_ax_name[i] where i = 0 ... 999 (maximum number of M functions, application-specific) |
| Data type | STRING |
| Data range | Maximum 16 characters (length of axis name, application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameterisation example:<br>The user-specific M function M10 is to act on the Z axis when programmed in the DIN syntax.<br>The user-specific M function M11 is act on the S2 spindle when programmed in the DIN syntax<br>*m_default_outp_ax_name[10]   Z*<br>*m_default_outp_ax_name[11]   S2*<br>* Note: The default value of variables is a blank string. |

### 1.2.1.4 Time-out and process times of M functions (P-CHAN-00040)

| **P-CHAN-00040** | **Timeout / process times of M functions for machining time calculation** |
|---|---|
| Description | The timeout times of M functions are set in the array ‘m_prozess_zeit[i]’. The field index ‘p’ defines the number of the M function. If the calculation of machining time is activated, the process times of the M functions are specified in this array. |

| | |
|---|---|
| Parameter | m_prozess_zeit[i] where i = 0 999 (maximum number of M functions, application-specific) |
| Data type | UNS32 |
| Data range | 0 ≤ m_prozess_zeit ≤ MAX(UNS32) |
| Dimension | μs |
| Default value | 0 |
| Remarks | This element is currently only used to calculate machining time.<br>Parameterisation example: The timeout and process time of M function 'M15' to specified as 0.5 s.<br>*m_prozess_zeit[15]     500000* |

## 1.2.2 Settings for H functions

### 1.2.2.1 Synchronisation type of H functions (P-CHAN-00027)

| **P-CHAN-00027** | **Synchronisation type of H functions** |
|---|---|
| Description | The synchronisation type of the corresponding H function is defined in the array ‘h_synch[i]’. Here, the field index ‘i’ defines the number of the H function. This value indicates the synchronisation type of the H function, i.e. when a check is made for the presence of a PLC acknowledgement. A motion is not executed, or is stopped at the latest towards the end of the block, if no acknowledgement has arrived from the PLC. The synchronisation type is defined as a string constant or a hexadecimal value. |
| Parameter | h_synch[i] where i = 0 999 (maximum number of H functions, application-specific) |
| Data type | STRING |
| Data range | See figure below |
| Dimension | ---- |
| Default value | NOT_VALID |
| Remarks | H functions are consumption information and they must be fetched (read) from the PLC. This also applies to H functions of the type MOS, MEP_MOS and MET_MOS. Otherwise, this results in a blocked interface to the HLI in the CNC and as a consequence to an unexpected processing stop.<br>**Caution:**<br>The following applies to synchronisation types with associated time and path-related pre-output (MET_SVS, MET_MOS, MEP_SVS, MEP_MOS):<br>If one of these synchronisation types is later changed into one which requires no pre-output value, P-CHAN-00107 [▸ 84] (m_pre_outp[i]) must be assigned to 0. Otherwise, a license error is generated in case of microwebs if this function is not licensed or not enabled (see P-CHAN-00600 [▸ 54] Alternatively: P-STUP-00060) is inactive.<br>Example<br>h_synch[12]     MVS_SVS     0x00000002<br>Note: Programming a UNS32 variable is permissible for downward compatibility reasons.<br>Example: m_synch[12]     0x00000002 |

| **Constant** | **Value** | **Meaning** |
|---|---|---|
| NOT_VALID | -1 | No valid H function |

| | | |
|---|---|---|
| NO_SYNCH | 0x00000000 | No output of H function to PLC |
| MOS | 0x00000001 | Output of H function to PLC without synchronisation.<br>Fetch H function from PLC! |
| MVS_SVS | 0x00000002 | Output of H function to PLC before motion block, synchronisation before motion block |
| MVS_SNS | 0x00000004 | Output of H function to PLC before motion block, synchronisation after motion block |
| MNS_SNS | 0x00000008 | Output of H function to PLC after motion block, synchronisation after motion block |
| MNE_SNS | 0x00000020 | Output of H function to PLC after (measurement) event, synchronisation after motion block<br>(for edge banding option only) |
| MVS_SLM | 0x00004000 | Late synchronisation, output of H function to PLC within the block, synchronisation during transfer to G01/G02/G03 (implicit synchronisation) |
| MVS_SLP | 0x00008000 | Late synchronisation, output of H function to PLC within the block, synchronisation by NC command #EXPL SYN (explicit synchronisation) |
| MOS_TS | 0x00040000 | Output of H function to PLC before motion block without synchronisation.<br>CNC calculates sampling time offset for high-precision time output in the PLC.<br>H function must be fetched from PLC! |
| MEP_MOS | 0x00100000 | Pre-output of H function with specified path, without synchronisation.<br>Fetch H function from PLC! |
| MET_MOS | 0x00200000 | Pre-output of H function with specified time, without synchronisation.<br>Fetch H function from PLC! |
| BWD_SYNCH | 0x00400000 | Synchronisation of H function during backward motion with MVS_SVS |
| FWD_SYNCH | 0x00800000 | Synchronisation of H function during 'Simulated forward motion' with associated synchronisation type |
| MEP_SVS | 0x01000000 | Output of H function with specified path, synchronisation before next block |
| MET_SVS | 0x02000000 | Output of H function with specified time, synchronisation before next block |
| FAW_SYNCH | 0x10000000 | Decoding stop (Flush and Wait): Output of H function to PLC and stop of program decoding at block end until all NC blocks in NC channel are completed. FAW_SYNCH can be set in addition to other synchronisation types. H functions with FAW_SYNCH may not be used during active tool radius compensation (TRC), polynomial contouring and HSC mode. |

Application: see section Examples of synchronisation of M functions [▶ 68]

### 1.2.2.2 Path or time-related pre-output of H functions (P-CHAN-00107)

| | |
|---|---|
| **P-CHAN-00107** | **Path or time-related pre-output of H functions** |

| | |
|---|---|
| Description | This parameter is used in connection with H functions<br>• of synchronisation types MET_SVS, MET_MOS and MEP_SVS, MEP_MOS. The lead time is specified for MET_SVS, MET_MOS and the lead distance for the MEP_SVS or MEP_MOS type.<br>• of synchronisation types MOS, MVS_SVS, MVS_SNS, MNS_SNS, MOS_TS and the ‘microwebs’ function which requires a separate license (see [FCT-C1]). Path-related parameters make sense here.<br>The field index 'i' defines the number of the associated H function. The value of h_pre_outp[i] defines the path and time-related output point before actual processing of the H function on the path. |
| Parameter | h_pre_outp[i] where i = 0 999 (maximum number of H functions, application-specific) |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | 0.1µm or µs |
| Default value | 0 |
| Remarks | The pre-output value can also be defined in the NC program [PROG//Chapter V.G. variables].<br>**Caution:**<br>If the synchronisation type of an H function is later changed into one which requires no pre-output value, h_pre_outp[i] must be assigned to 0. Otherwise, a license error is generated in case of microwebs if this function is not licensed or not enabled (see P-STUP-00060).<br>**Parameterisation example:**<br>User-specific H functions H96 and H98 must be output to the PLC 10 mm before reaching the synchronisation position in the block sequence.<br>The output of the user specific H functions H97 and H99 to the PLC must be executed 40 milliseconds before reaching the time of synchronisation in the block sequence.<br><br>*# Definition of H functions and synchronisation types*<br>*# ==========================================*<br>*h_synch[96]    0x01000000   MEP_SVS*<br>*h_synch[97]    0x02000000   MET_SVS*<br>*h_synch[98]    0x00100000   MEP_MOS*<br>*h_synch[99]    0x00200000   MET_MOS*<br>*#*<br>*# Definition of pre-output path, pre-output time*<br>*#=====================================*<br>*h_pre_outp[96]    100000    in 0.1µm*<br>*h_pre_outp[97]     40000    in µs*<br>*h_pre_outp[98]    100000    in 0.1µm*<br>*h_pre_outp[99]     40000    in µs* |

**For further examples see section** Synchronisation types of M functions (P-CHAN-00041) [▶ 65]**.**

### 1.2.2.3 Axis-specific H functions (P-CHAN-00025)

| P-CHAN-00025 | Axis-specific H functions |
|---|---|

| | |
|---|---|
| Description | User-specific H functions programmed in DIN syntax are processed and executed channel-specific.<br>If the user wishes to force axis-specific handling for specific H functions, it is possible to configure them using this parameter so that they have an axis-specific effect. Each H function can be assigned an axis name on which it is to act. Both path axes and spindle axes are permissible. |
| Parameter | h_default_outp_ax_name[i] where i = 0... 999 (maximum number of H functions, application-specific) |
| Data type | STRING |
| Data range | Maximum 16 characters (length of axis name, application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameterisation example:<br>The user-specific H function H10 is to be programmed in DIN syntax to act on the Z axis.<br>The user-specific H function H11 is to be programmed in DIN syntax to act on the S2 spindle axis.<br>*h_default_outp_ax_name[10]    Z*<br>*h_default_outp_ax_name[11]    S2*<br>* Note: The default value of variables is a blank string. |

### 1.2.2.4 Time-out / process times of H functions (P-CHAN-00026)

| P-CHAN-00026 | Timeout / process times of H functions for machining time calculation |
|---|---|
| Description | The timeout times of H functions are specified in the array ‘h_prozess_zeit[i]’. The field index ‘i’ defines the number of the H function. If the calculation of machining time is activated, the process times of H functions are specified in this array. |
| Parameter | h_prozess_zeit[i] here i = 0... 999 (maximum number of H functions, application-specific) |
| Data type | UNS32 |
| Data range | 0 < P-CHAN-00026 < MAX(UNS32) |
| Dimension | [µs] |
| Default value | 0 |
| Remarks | This element is currently only used to calculate machining time.<br>Parameterisation example: The following example specifies the timeout and process time of H function 'H1' to 20 ms.<br>*h_prozess_zeit[15]    20000* |

## 1.2.3 Settings for tool change (tool.* or werkzeug.*)

### 1.2.3.1 Synchronisation type of tool request (P-CHAN-00086)

| P-CHAN-00086 | Synchronisation type of tool request (T function) |
|---|---|
| Description | This parameter defines the synchronisation type of a tool request (T function). The synchronisation type is defined as a string constant or a hexadecimal value. |
| Parameter | tool.synch |
| Data type | STRING |
| Data range | See figure below |
| Dimension | ---- |
| Default value | NO_SYNCH |

| | |
|---|---|
| Remarks | *werkzeug.synch (Old syntax)*<br>*Example*<br>*tool.synch* MNS_SNS 0x00000008<br>Note: Programming a UNS32 variable is permissible for downward compatibility reasons.<br>Example: *tool.synch* 0x00000008 |

| Constant | Value | Meaning |
|---|---|---|
| NO_SYNCH | 0x00000000 | No output of T function to PLC |
| MOS | 0x00000001 | Output of T function to PLC without synchronisation |
| MVS_SVS | 0x00000002 | Output of T function to PLC before motion block, synchronisation before motion block |
| MVS_SNS | 0x00000004 | Output of T function to PLC before motion block, synchronisation after motion block |
| MNS_SNS | 0x00000008 | Output of T function to PLC after motion block, synchronisation after motion block |
| MNE_SNS | 0x00000020 | Output of T function to PLC after (measurement) event, synchronisation after motion block<br>(for edge banding option only) |
| BWD_SYNCH | 0x00400000 | Synchronisation of T function during backward motion with MVS_SVS |
| FWD_SYNCH | 0x00800000 | Synchronisation of T function during 'Simulated forward motion' with the associated synchronisation type |
| FAW_SYNCH | 0x10000000 | Stop of decoding (Flush and Wait): Output of T function to PLC and stop of program decoding at block end until all NC blocks in NC channel are completed. FAW_SYNCH can be set in addition to other synchronisation types. T functions with FAW_SYNCH may not be used during active tool radius compensation (TRC), polynomial contouring and HSC mode. |

**For examples, see section** Synchronisation types of M functions [▶ 65]**.**

### 1.2.3.2 Timeout / process time of tool request (P-CHAN-00072)

| P-CHAN-00072 | Timeout / process time of tool request for machining time calculation |
|---|---|
| Description | This parameter specifies the timeout time of a tool request. If the calculation of machining time is activated, this element specifies the process time of the tool request. |
| Parameter | tool.prozess_zeit |
| Data type | UNS32 |
| Data range | 0 < prozess_zeit < MAX(UNS32) |
| Dimension | µs |
| Default value | 0 |
| Remarks | *werkzeug.prozess_zeit (old syntax)*<br>This element is currently only used to calculate machining time.<br>Parameterisation example: The following example specifies the timeout and process time of a tool request to 2 s.<br>*tool.prozess_zeit 2000000* |

### 1.2.3.3 Minimum permissible tool length (P-CHAN-00156)

| P-CHAN-00156 | Minimum permissible tool length |
|---|---|
| Description | If this parameter is unequal to zero, the length of the new tool to be replaced is checked for this minimum value and if required, the error message P-ERR-21426 'Tool length is smaller than minimum limit value' is output. |
| Parameter | tool.minimum_length |
| Data type | UNS32 |
| Data range | 0 ≤ minimum_length ≤ MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | |

### 1.2.3.4 Minimum permissible tool radius (P-CHAN-00157)

| P-CHAN-00157 | Minimum permissible tool radius |
|---|---|
| Description | If this parameter is unequal to zero, the radius of the new tool to be changed is checked for this minimum value and if required, the error message P-ERR-21427 'Tool radius is smaller than minimum limit value' is output. |
| Parameter | tool.minimum_radius |
| Data type | UNS32 |
| Data range | 0 ≤ minimum_radius ≤ MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | |

## 1.2.4 Process time of homing (P-CHAN-00077)

| P-CHAN-00077 | Process time of homing for machining time calculation |
|---|---|
| Description | A process time for homing must be specified by this parameter to calculate machining time. |
| Parameter | rpf_prozess_zeit |
| Data type | UNS32 |
| Data range | 0 < rpf_prozess_zeit < MAX(UNS32) |
| Dimension | µs |
| Default value | 0 |
| Remarks | This element is currently only used to calculate machining time.<br>Parameterisation example: The following example specifies the process time for homing to 3 s.<br>*rpf_prozess_zeit 3000000* |

## 1.2.5 Calculation model for M/H pre-output time (P-CHAN-00209)

| P-CHAN-00209 | Calculation model for M/H pre-output time |
|---|---|

| | |
|---|---|
| Description | The time calculation function for the pre-output time can be controlled by means of this parameter for the M/H synchronisation type MET_SVS.<br>If the parameter is set to 0, the pre-output time is calculated independently of the actually active slope profile with a linear profile model. If a non-linear slope profile is active, the profile time is then only estimated.<br>If the parameter is set to 1, the pre-output time is calculated depending on the active slope profile. Very precise time values are obtained by taking the ramp time into account in the non-linear profile.<br>Disadvantage: The more complex algorithms in the time calculation function require considerably more computing time in the CNC real-time part. |
| Parameter | m_h_pre_outp_time_calc_mode |
| Data type | BOOLEAN |
| Data range | 0: Time calculation model based on linear slope profile (default).<br>1: Time calculation model based on active slope profile |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.2.6 Tolerance for deviation with pre-output position of M/H function (P-CHAN-00760)

| P-CHAN-00760 | Tolerance for the deviation between the planned and real pre-output positions of an M/H function |
|---|---|
| Description | With M and H functions with pre-output, this parameter defines the maximum permissible deviation between the planned and actual pre-output positions.<br>If the actual deviation is greater than P-CHAN-00760, Error ID 120815 is output and program processing is aborted.<br>With deviations within the permissible range, the M or H function is output at the corresponding point without a warning.<br>If P-CHAN-00760 is assigned 0, the tolerance check is inactive. |
| Parameter | pre_output_tolerance |
| Data type | UNS32 |
| Data range | 0 ≤ P-CHAN-00760 ≤ MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | The parameter is available as of V3.1.3079.40<br>The smallest possible tolerance check is reached by the value 1. A maximum deviation of 0.1µm is permitted. |

## 1.2.7 Residual path/time calculation with M/H pre-output (P-CHAN-00212)

| P-CHAN-00212 | Activate residual path/time calculation with M/H code look ahead |
|---|---|
| Description | This parameter activates the calculation and provision of residual path or time for the M/H synchronisation types MEP_SVS and MET_SVS. If the parameter is set to 1, the residual path or time is calculated after output of all M functions of synchronisation types MEP_SVS and MET_SVS relative to the synchronisation point. CNC objects* support access to the values. The look-ahead function waits until the current active synchronisation point is crossed before it changes to the next synchronisation point. |
| Parameter | m_h_pre_outp_calc_value_to_go |
| Data type | BOOLEAN |
| Data range | 0: No calculation of residual path or time. As soon as all M functions of one synchronisation point are output, the function changes to the next synchronisation point (default).<br>1: M code look ahead with calculation of residual path or time. The function only changes to the next synchronisation point after the current active synchronisation point is crossed. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | * Accesses to CNC objects:<br>Path to synchronisation point: [0.1 µm] Index Group: 0x21301 Offset: 0x27<br>Time to synchronisation point: [1 µs] Index Group: 0x21301 Offset: 0x28 |

```
N10 G01 X10 G90 F5000
N20 X20
N30 X30
N40 X40
N45 M96
N50 X80
N60 X0
```

```
N70 M30
```

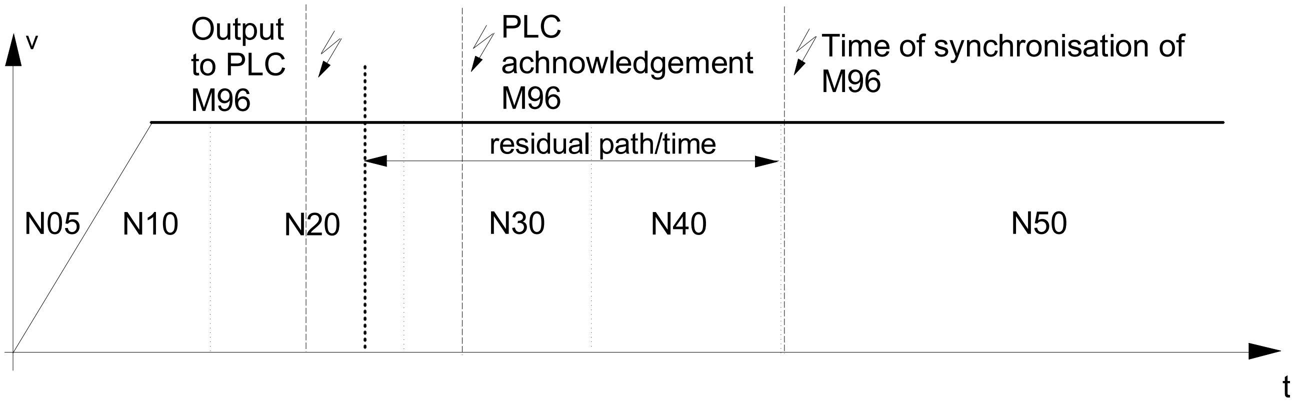


Fig. 15: Activation of residual path/time calculation with M/H pre-output

## 1.2.8 Number of NC blocks at M/H code look ahead (P-CHAN-00274)

| **P-CHAN-00274** | **Number of NC blocks at M/H code look ahead** |
|---|---|
| Description | This parameter increases the number of NC blocks for look ahead for M/H synchronisation types MEP_SVS and MET_SVS. By default the number of blocks is 50.<br>Due to the additionally required block buffer, the parameter P-CHAN-00653 [▶ 63] and possibly P-CHAN-00650 [▶ 61] (or alternatively start-up parameter P-STUP-00071 and possibly P-STUP-00070must be adapted for values > 70.<br>The increased number of blocks causes an increased cycle time in the real-time part of CNC. |
| Parameter | m_h_pre_outp_nbr_block |
| Data type | UNS32 |
| Data range | 50 ≤ P-CHAN.00274 ≤ 200 |
| Dimension | ---- |
| Default value | 50 |
| Remarks | Configuration example:<br>• Channel parameter list:<br>`m_h_pre_outp_nbr_block     100      #100 blocks M code look ahead`<br>`# P-CHAN-00653 – Large Look-Ahead buffer`<br>`configuration.interpolator.parameter  100`<br>`# P-CHAN-00655 – Activate customer-specific`<br>`# setting for Look-Ahead`<br>`configuration.interpolator.fct_enable[0]  FCT_LOOK_AHEAD_CUSTOM`<br>• Alternatively - in the start-up parameter list, modify instead of P-CHAN-00653 [▶ 63] and P-CHAN-00650 [▶ 61]<br>`configuration.channel[0].interpolator.parameter  100`<br>`configuration.channel[0].interpolator.function   FCT_LOOK_AHEAD_CUSTOM` |

# 1.3 Dimensional units and resolutions

## 1.3.1 Dimensional unit for translatory axes (P-CHAN-00054)

| **P-CHAN-00054** | **Dimensional unit for translatory axes in NC program** |
|---|---|
| Description | This parameter defines the dimensional unit in which the movements of all translatory axes are programmed in the NC program. |
| Parameter | mass_einh |
| Data type | UNS08 |
| Data range | 0: Dimensional unit is millimeter<br>1: Dimensional unit is inch |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: Following example defines that all translatory axes are specified in the dimensional unit 'millimeter' [mm].<br>*mass_einh 0* |

## 1.3.2 Dimensional unit for rotary axes (P-CHAN-00079)

| **P-CHAN-00079** | **Dimensional unit for rotary axes in NC program** |
|---|---|
| Description | This parameter defines the dimensional unit in which the motions of all rotary axes are programmed in the NC program. |
| Parameter | rund_mass_einh |
| Data type | UNS08 |
| Data range | 0: Unit is degrees (full circle 360°)<br>1: Unit is gradians (full circle 400°) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.3.3 Scaling factor of axis resolution (P-CHAN-00315)

| **P-CHAN-00315** | **Scaling factor of axis resolution** |
|---|---|
| Description | The value of this parameter represents the conversion factor between the motion paths or positions programmed in the NC program and the internal representation for positional values of the axes. |
| Parameter | resolution_factor |
| Data type | SGN32 |
| Data range | 0 ≤ resolution_factor < MAX(SGN32) |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | Not to be confused with the resolution of the drive encoder P-AXIS-00234/P-AXIS-00235.<br>When a value is unequal to zero, P-CHAN-00315 as of CNC Build V2.11.2026.09 and higher replaces the parameters lin_aufloes, rund_aufloes and spind_aufloes (P-CHAN-00034 [▸ 93], P-CHAN-00035 [▸ 140] and P-CHAN-00036 [▸ 145]). It is recommended to use only P-CHAN-00315 as of this CNC Build.<br>Parameterisation example: All translatory axes are specified in the dimension 'millimeter' [mm] and all rotary or spindle axes are specified in the dimension ‘degree’ [°] in the NC program.<br>*resolution_factor 10000* |

| Axis type<br>P-AXIS-00016 | Internal representation in [0.1 µm] or [0.0001°] | Typical value for resolution | Programmed value in the NC program in [mm] or [°] |
|---|---|---|---|
| Translatory axis type | 10 000 [0.1µm] | 10 000 | 1 [mm] |
| Rotator/spindle axis type | 10 000 [0.0001°] | 10 000 | 1 [°] |

## 1.3.4 Resolution of linear axes (P-CHAN-00034)

| **P-CHAN-00034** | **Resolution of linear axes** |
|---|---|
| Description | This value of this parameter represents the conversion factor between the motion paths or positions programmed in the NC program and the internal representation of positional values for linear axes. |
| Parameter | lin_aufloes |
| Data type | REAL64 |
| Data range | 0 < lin_aufloes < MAX(REAL64) |
| Dimension | ---- |
| Default value | 10000 |
| Remarks | Not to be confused with the resolution of the drive encoder P-AXIS-00234/P-AXIS-00235.<br>As of CNC Build V2.11.2026.09 P-CHAN-00315 [▸ 92] replaces the parameters lin_aufloes, rund_aufloes and spind_aufloes (P-CHAN-00034, P-CHAN-00035 [▸ 140] and P-CHAN-00036 [▸ 145]). It is recommended to use only P-CHAN-00315 [▸ 92] as of this CNC Build.<br>Parameterisation example: All translatory axes are specified in the dimension 'millimeter' [mm] in the NC program.<br>*lin_aufloes 10000* |

| Internal representation in [0.1 µm] | Typical value for liner resolution | Programmed value in the NC program in [mm] |
|---|---|---|
| 10 000 [0.1µm] | 10 000 | 1 [mm] |

## 1.3.5 Resolution of rotary axes (P-CHAN-00078)

| **P-CHAN-00078** | **Resolution of rotary axes** |
|---|---|
| Description | This value represents the conversion factor between the path motions or the positions programmed in the NC program and the internal representation for position values with rotary axes. |
| Parameter | rund_aufloes |
| Data type | REAL64 |
| Data range | 0 < rund_aufloes < MAX(REAL64) |
| Dimension | ---- |

| | |
|---|---|
| Default value | 10000 |
| Remarks | Not to be confused with the resolution of the drive encoder P-AXIS-00234 [▶ 93]/P-AXIS-00235.<br>As of CNC Build V2.11.2026.09 P-CHAN-00315 [▶ 92] replaces the parameters lin_aufloes, rund_aufloes and spind_aufloes (P-CHAN-00034 [▶ 93], P-CHAN-00035 [▶ 140] and P-CHAN-00036 [▶ 145]). It is recommended to use only P-CHAN-00315 [▶ 92] as of this CNC Build.<br>Parameterisation example: All rotary axes are specified in the dimension 'Degree' [°] in the NC program.<br>*rund_aufloes 10000* |

| Internal representation in [0.0001°] | Typical value for rotary axis resolution | Programmed value in the NC program in [°] |
|---|---|---|
| 10 000 [0.0001 °] | 10 000 | 1 [°] |

## 1.3.6 Resolution of spindles (P-CHAN-00083)

| **P-CHAN-00083** | **Resolution of spindles** |
|---|---|
| Description | This value represents the conversion factor between the positions programmed in the NC program and the internal representation of positional values for spindles (M19). |
| Parameter | spind_aufloes |
| Data type | REAL64 |
| Data range | 0 < spind_aufloes < MAX(REAL64) |
| Dimension | ---- |
| Default value | 10000 |
| Remarks | This resolution is used for regulated and controlled spindles. It is used only for positional values (M19) and not for speeds.<br>Not to be confused with the resolution of the drive encoder P-AXIS-00234 [▶ 93]/P-AXIS-00235.<br>As of CNC Build V2.11.2026.09 P-CHAN-00315 [▶ 92] replaces the parameters lin_aufloes, rund_aufloes and spind_aufloes (P-CHAN-00034 [▶ 93], P-CHAN-00035 [▶ 140] and P-CHAN-00036 [▶ 145]). It is recommended to use only P-CHAN-00315 [▶ 92] as of this CNC Build.<br>Parameterisation example: All spindles are specified in the dimension 'degree' [°] in the NC program.<br>*spind_aufloes 10000* |

| Internal representation in [0.0001°] | Typical value for spindle resolution | Programmed value in the NC program in [°] |
|---|---|---|
| 10 000 [0.0001 °] | 10 000 | 1 [°] |

# 1.4 Settings for tool management

## 1.4.1 Implicit execution of D word with T (P-CHAN-00014)

| **P-CHAN-00014** | **Implicit execution of D word with T** |
|---|---|
| Description | If a T command is programmed, the D command is implicitly executed with the same number as the T command. This mechanism can be selected both for a list-based internal and for an external tool management. |

| | |
|---|---|
| | This parameter has no effect on the programming of #TOOL PREP (corresponds to the T command). Afterwards, a #TOOL DATA must always be programmed. |
| Parameter | einrechnen_mit_t |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: The parameter 'einrechnen_mit_t' is assigned the value 1. The programming of 'T80' then corresponds to the programming command 'T80 D80'.<br>*einrechnen_mit_t 1* |

## 1.4.2 Tool data is located in an external tool management (P-CHAN-00016)

| P-CHAN-00016 | Tool data is requested from an external tool management |
|---|---|
| Description | If the system works with external tool management (e.g. integrated in PLC), this parameter is set to 1. |
| Parameter | ext_wzv_vorhanden |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: The system works with an external tool management.<br>*ext_wzv_vorhanden 1* |

### 1.4.2.1 Information of the external tool management (P-CHAN-00087)

| P-CHAN-00087 | Information of the external tool management in case of a T function |
|---|---|
| Description | Programming a T function generates a message which is sent to the external tool management to prepare the magazine for a tool change.<br>Assignment is only executed if an external tool management is used ((P-CHAN-00016 [▶ 95]). |
| Parameter | t_info_to_wzv |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: If a T function is programmed, the external tool management prepares a tool change.<br>*t_info_to_wzv 1* |

### 1.4.2.2 Saving and deleting of tool data (P-CHAN-00103)

| P-CHAN-00103 | Save and delete tool data in an external tool management |
|---|---|
| Description | Tool data of the current tool (e.g. wear parameters) can be modified via variable access (V.G.WZ_AKT...) in the NC program.<br>For saving, this data is written back to external tool management each time a new tool is selected. If a tool was programmed in the previous NC program, D0 (tool deselection) is implicitly requested at Program restart. With this request the data of the tool last used is transmitted back at the same time to external tool management. This ensures that no modified tool data is lost. |

| | |
|---|---|
| | This saving by implicitly initiating a D0 at program start can be achieved by setting the parameter to 1. Assignment is only executed if an external tool management is used ((P-CHAN-00016 [▶ 95]). |
| Parameter | d_clear_to_wzv |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is only used in TwinCAT systems.<br>On reset, no implicit D0 is initiated. This means that the data of the tool last used is only saved on the next program start. |

## 1.4.3 Initiating FLUSH mechanisms

During communication with an external tool management, there may be waiting times for processing during T and D programming for synchronisation reasons. Due to the buffering of the NC channel a number of motion-related NC blocks can be held back before these commands, i.e. execution of the movement is also delayed. The execution of the delayed motion blocks can be forced by configuring implicit FLUSH mechanisms.

### 1.4.3.1 Releasing motion blocks during T-selection (P-CHAN-00106)

| **P-CHAN-00106** | **Enabling motion blocks during T selection** |
|---|---|
| Description | By setting the parameter to 1, all motion blocks up to the T word are executed. This is an implicit '#FLUSH CONTINUE' initiated by the T word[PROG// Flushing NC channel].<br>Assignment only makes sense if an external tool management is used (P-CHAN-00016 [▶ 95]). |
| Parameter | t_with_implicit_flush |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.4.3.2 Flushing NC channel by implicit '#FLUSH WAIT' (flush.*)

#### 1.4.3.2.1 '#FLUSH WAIT' on selection of D word (P-CHAN-00271)

| **P-CHAN-00271** | **Implicit '#FLUSH WAIT' on selection of D word** |
|---|---|
| Description | When this parameter is set, the implicit initiation of a FLUSH mechanism (cf. [PRG]) can be configured. The FLUSH mechanism is initiated before or after the D word depending on the identification (ID). |
| Parameter | flush.d_cmd |
| Data type | STRING |
| Data range | D or not assigned: without implicit FLUSH mechanism (default).<br>FAW_D: Initiation of an implicit '#FLUSH WAIT' before the D word.<br>*D_FAW: Not available at the moment.* |
| Dimension | ---- |
| Default value | D |
| Remarks | The implicit FLUSH mechanisms of the D command are always executed, even in combination with parameters of the T command P-CHAN-00014 [▶ 94] (einrechnen_mit_t) or P-CHAN-00106 [▶ 97] (t_with_implicit_flush).<br>In addition P-CHAN-00271 is also effective if #TOOL DATA is used. |

## 1.4.4 Tool life data recording

Assigning the parameters for tool life recording only makes sense when external tool management is used (P-CHAN-00016 [▶ 95]).

### 1.4.4.1 Send tool life data (P-CHAN-00076)

| **P-CHAN-00076** | **Send tool life data to external tool management** |
|---|---|

| | |
|---|---|
| Description | If an external tool management system is used, only tool data are requested from it by default.<br>This parameter must also be set to send the tool life data of a tool determined in the NC kernel to the external tool management. |
| Parameter | tool_life_to_wzv |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.4.4.2 Suppress output of tool life data zero (P-CHAN-00243)

| **P-CHAN-00243** | **Suppress output of tool life parameters 0** |
|---|---|
| Description | When parameter P-CHAN-00076 [▶ 97] is set, the tool life data is sent to the external tool management on every tool change, even if this is zero.<br>This parameter can be set if no tool life data is sent in the event that service distance and service life = zero . |
| Parameter | tool_life_to_wzv_suppress_zero |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is available as of CNC Build V2.11.2802.00 and higher. |

### 1.4.4.3 Tool life data recording for rapid traversing movements (P-CHAN-00255)

| **P-CHAN-00255** | **Tool life data recording for rapid traverse motions** |
|---|---|
| Description | With tool life data recording, no rapid traverse motions G00 are recorded by default. This parameter achieves that rapid traverse motions are also considered for tool life data recording. |
| Parameter | tool_life_capture_all_movements |
| Data type | BOOLEAN |
| Data range | 0: Rapid traverse motions G00 are not included in tool life data recording (default).<br>1: Rapid traverse motions G00 are also included in tool life data recording. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.4.4.4 Tool life data recoding with D word (P-CHAN-00482)

| P-CHAN-00482 | Tool life data recording with D word |
|---|---|
| Description | Programming the T word does not imply that a mechanical tool change takes place. Nevertheless, tool lie data recording is already activated for this tool in the basic setting. If the D word (P-CHAN-00014 [▸ 94] = 1) is also implicitly executed with the T word, the detected tool life data indicates a precise value.<br>However, if T and D are programmed separately (P-CHAN-00014 [▸ 94] = 0) and if other motion blocks are located between the T word and the transfer of tool data by the D word, the detected tool life of the currently active tool is too short and the detected tool life of the new tool is too long.<br>To minimise this error, this parameter can be used to change the activation of the tool life data recording to the programming of the D word as trigger point. The detected tool life data is then more precise since it includes all the motions executed by a tool. |
| Parameter | tool_life_with_d |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is available as of CNC Build V2.11.2045.00 and higher. |

## 1.5 Settings for edge banding

Edge bending is described in more detail in [PROG].

### 1.5.1 Residual path (P-CHAN-00030)

| P-CHAN-00030 | Residual path on arrival of measurement signal at edge banding |
|---|---|
| Description | The residual path is the path not yet traversed with edge banding after receipt of the measurement signal. The dimensional units correspond to the settings for units and resolutions. |
| Parameter | kasto_residual_path |
| Data type | SGN32 |
| Data range | 0 ≤ kasto_residual_path ≤ MAX(SGN32) |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | Please consider that for curved contours the residual path must be specified based on the centre point path of the tool.<br>Parameterisation example: After the arrival of the measurement signal, the motion continues for another 100 units.<br>*kasto_residual_path 100* |

### 1.5.2 Block global edge banding (P-CHAN-00029)

| P-CHAN-00029 | Block global edge banding |
|---|---|
| Description | For contours that are described by many short motion blocks (e.g. generated by CAD/CAM tool), it is difficult to match the measurement signal of edge banding on a defined motion block. When this datum is set to 1, edge banding can be executed for several blocks after start of the measurement travel [PROG]. |
| Parameter | kasto_multi_block |
| Data type | BOOLEAN |

| | |
|---|---|
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: Block global edge banding is enabled.<br>*kasto_multi_block 1* |

### 1.5.3 Reject rest block (P-CHAN-00149)

| **P-CHAN-00149** | **Reject residual block on arrival of the measurement signal and continue with next NC block at edge banding** |
|---|---|
| Description | After output and acknowledgement of the specific edge banding M function, motion is not continued to the target point of the original measurement block. After an automatic position request, machining is continued with the next NC block. |
| Parameter | kasto_reject_rest_block |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | It only makes sense to use this parameter if P-CHAN-00029 [▶ 99] (kasto_multi_block) is assigned FALSE.<br>Parameterisation example: Reject the residual block and the following position request is enabled.<br>*kasto_reject_rest_block 1* |

*kasto_reject_rest_block = 0:*

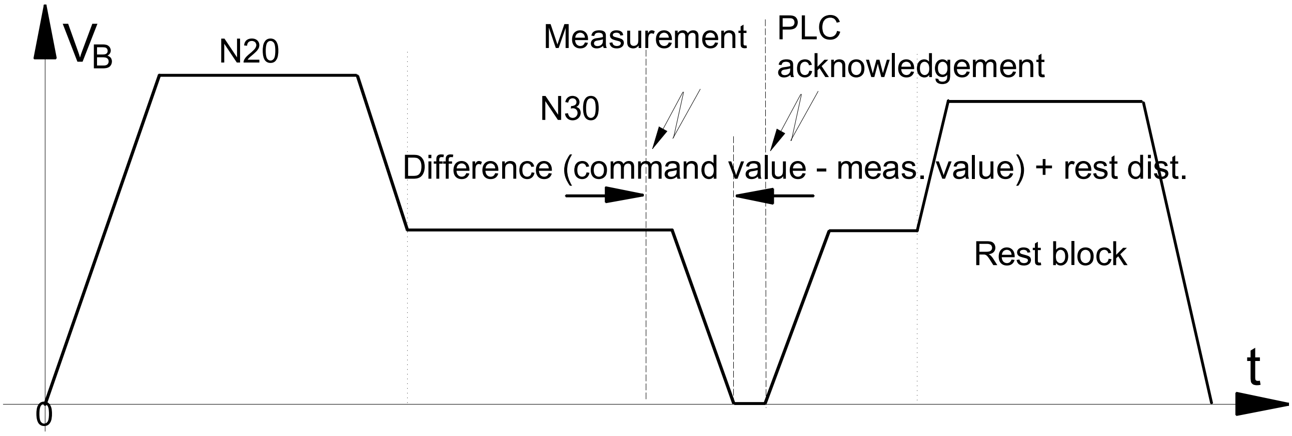


Fig. 16: After output of the M function, motion is continued up to the measurement block (default)

*kasto_reject_rest_block = 1:*

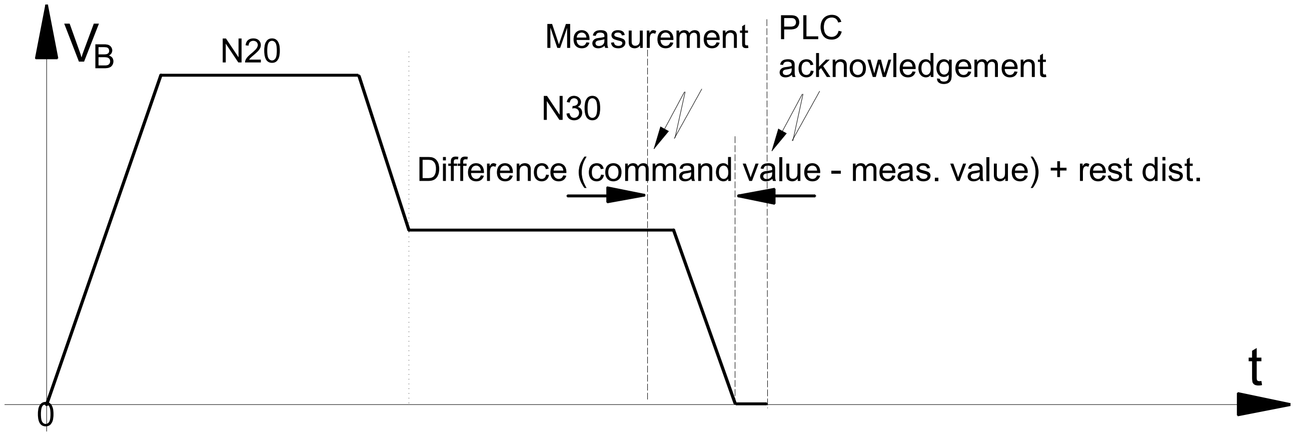


Fig. 17: After output of the M function, reject the residual measurement block

## 1.5.4 Modal active measurement (P-CHAN-00153)

| **P-CHAN-00153** | **Modal active measurement with edge bending** |
|---|---|
| Description | A measurement process requires the programming of G108 with a following MNE_SNS M function. This parameter activates rapid measurement in combination with SERCOS drives.<br>The measurement cycles then start with the programmed MNE_SNS- M function in a machining loop.<br>If P-CHAN-00153 is set, edge banding must be deactivated before program end. If the parameter is not deactivated, the error ID 50573 is output at program end. |
| Parameter | kasto_remain_active |
| Data type | BOOLEAN |
| Data range | 0: G108 must be programmed with a following MNE_SNS M function for <u>every</u> measurement cycle (default).<br>Programming example: M33 with synchronisation type MNE_SNS<br>`N10 $WHILE V.E.WAIT != 1`<br>`N10 G108`<br>`N20 G01 G90 X50`<br>`N30 G01 G90 X100 F2000 M33`<br>`N40 X0 Y20`<br>`N50 $ENDWHILE`<br>`M30`<br>1: For several measurement cycles G108 need only be programmed <u>once</u>. The measurement process is activated in the CNC with the MNE_SNS- M function.<br>Programming example: M33 with synchronisation type MNE_SNS<br>`N05 G108`<br>`N10 $WHILE V.E.WAIT != 1`<br>`N20 G01 G90 X50`<br>`N30 G01 G90 X100 F2000 M33`<br>`N40 X0 Y20`<br>`N50 $ENDWHILE`<br>`N60` **`G107`** `(End of measuring edge banding)`<br>`M30` |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | It only makes sense to use this parameter if P-CHAN-00029 [▸ 99] (kasto_multi_block) is assigned FALSE.<br>Parameterisation example: Rapid measurement for measurement cycles is activated with G108.<br>*kasto_remain_active 1* |

## 1.5.5 Suppress motion path check (P-CHAN-00227)

| **P-CHAN-00227** | **Suppress motion path check with edge banding** |
|---|---|
| Description | When the measurement signal is received, the CNC checks whether there is enough motion path in the block to traverse the residual path defined in the channel parameter and to stop. An error message is issued if this is not the case. With some applications, it may make sense not to run this check until deselection via G107 or at the end of the last travel movement. In this case, it is possible to interrupt reduction of the residual path, e.g. by means of programmed dwell times (see figure below). This is achieved by setting the channel parameter to 1. If the motion path available in the block is not sufficient, an error reaction or issuing an error message takes place at the end of the movement or on deselection of the function via G107. |
| Parameter | kasto_suppress_path_check |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | It only makes sense to use this parameter if P-CHAN-00029 [▸ 99] (kasto_multi_block) is assigned FALSE.<br>Parameterisation example:<br>*kasto_suppress_path_check 1* |

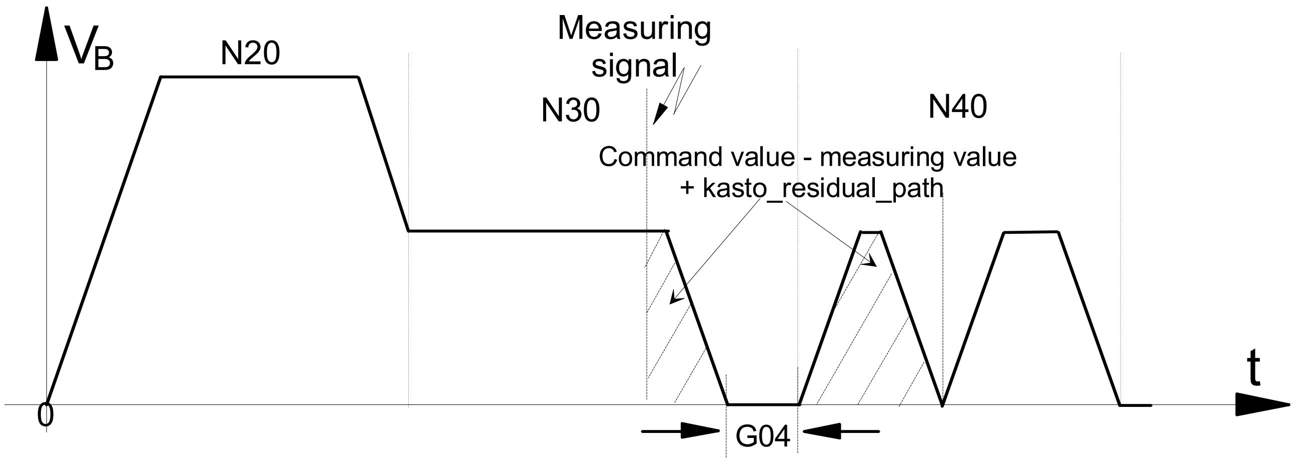


Fig. 18: Suppress motion path check with edge banding

## 1.5.6 Programmed measurement signal (P-CHAN-00257)

| **P-CHAN-00257** | **Switch over measurement logic to a programmed measuring signal for edge bending** |
|---|---|
| Description | Typically, the measurement signal for the CNC is routed either via the drive bus or is set by the PLC via the HLI interface.<br>When the channel parameter P-CHAN-00257 is set to 1, as an alternative you can switch over to a 'programmed measurement trigger signal' that is active in the path interpolator.<br>The measurement trigger signal is then triggered by the NC command #MEAS [TRIGGER] within the G108/G107 range. Activation of the measurement logic in the drive and the response to the measurement signals from the drive or the PLC are then deactivated. |
| Parameter | kasto_prog_meas_trigger |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | It only makes sense to use this parameter if P-CHAN-00029 [▶ 99] (kasto_multi_block) is assigned FALSE.<br>Parameterisation example:<br>*kasto_prog_meas_trigger 1* |

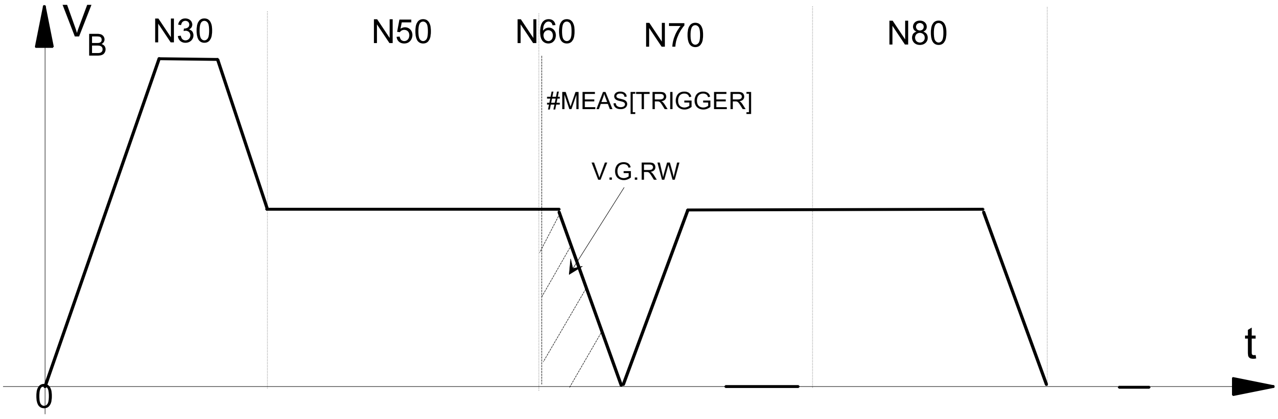


Fig. 19: Switch over measurement logic to a programmed measurement signal at edge banding

### Switch over measurement logic to a programmed measurement signal for edge banding

```
%meas
N10 V.G.RW= 20              ;Residual path after measuring signal 20mm
N20 G00 G90 X0 Y0
N30 G01 X50 F5000
N40 G108 F2500              ;Activate edge banding function
N50 G01 G90 X100 Y100 M33   ;M function 33 of type MNE_SNS
N60 #MEAS [TRIGGER]         ;Execute measurement signal at current position
N70 X150 Y150
N80 X200 Y200
N90 G107                    ;Deselect edge banding
M30
```

## 1.5.7 Residual path related to contour (P-CHAN-00258)

| **P-CHAN-00258** | **Switch over to contour-related residual path for edge banding** |
|---|---|
| Description | By default the defined residual path (P-CHAN-00030 [▶ 99]) or the programmed residual path (V.G.RW) is interpreted as a value related to the centre point path of the milling tool. After response by the probe, the controller moves the residual path to the tool centre point path, stops and then outputs the corresponding M function.<br>The channel parameter must be set to 1 if the residual path is to refer to the contour when tool radius compensation is active. The controller then calculates the motion path on the tool centre point path so that the required residual path moves along the contour.<br>A rotary simulation axis must be configured to use this function. Set the operation mode 0x04000004 (P-AXIS-00015) for this axis in the axis parameters. |
| Parameter | kasto_relate_to_prog_contour |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*kasto_relate_to_prog_contour 1* |

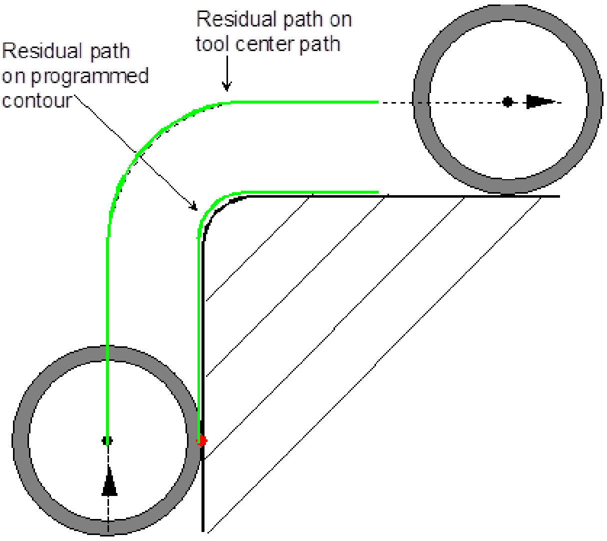

Residual path on tool center path
Residual path on programmed contour

Fig. 20: Switch over to contour-related residual path for edge banding

## 1.5.8 Warning on non-detection of probing signal (P-CHAN-00298)

| P-CHAN-00298 | Output a warning message if probing signal is not detected at edge banding |
|---|---|
| Description | By default the CNC generates the error message P-ERR-50063if no measuring signal is detected for edge banding G107. By setting the parameter to 1, the error message can be changed to a warning message. The program run is then continued without measuring signal. The faulty measurement must then be rectified by the user, e.g. in the PLC. |
| Parameter | kasto_warning_no_signal |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter can only be used in the ‘Reject rest block’ mode (P-CHAN-00149 [▶ 100] must therefore be set to the value 1).<br>Parameterisation example:<br>`# Reject rest block`<br>`kasto_reject_rest_block      1`<br>`# Generate a warning message not an error message if measurement signal is not detected.`<br>`kasto_warning_no_signal      1` |

## 1.5.9 Warning message if residual motion path is too small (P-CHAN-00299)

| P-CHAN-00299 | Output of a warning message if residual motion path is too small at edge banding |
|---|---|
| Description | After the measurement signal is received, the CNC executive plausibility checks for the residual path at edge banding, e.g. whether the residual path is greater than the deceleration distance. By default the CNC generated error messages (P-ERR-50443, P-ERR-50498 or P-ERR-50065) if the expected conditions are not fulfilled. By setting the parameter to 1, the error message can be changed into warning messages. Program processing is then continued and the axes move to the target point of the measurement travel. The faulty measurement must be treated properly by the user e.g. in the PLC |
| Parameter | kasto_warning_residual_path |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter can only be used in the ‘Reject rest block’ mode (P-CHAN-00149 [▶ 100] must therefore be set to the value 1).<br>Parameterisation example:<br>`# Reject rest block`<br>`kasto_reject_rest_block       1`<br>`# Raise warning messages, when remaining residual path is too small.`<br>`kasto_warning_residual_path   1` |

## 1.5.10 Outputting an MNE_SNS technology function at measuring event (P-CHAN-00435)

| **P-CHAN-00435** | **Output an MNE_SNS technology function at measurement event** |
|---|---|
| Description | If an M function of synchronisation type MNE_SNS is also programmed in a measurement block, this channel parameter configures the earliest time of output to the PLC after the measurement event. |
| Parameter | kasto_techno_output_with_signal |
| Data type | BOOLEAN |
| Data range | 0: After the measuring signal is received, the residual path is removed and a motion stop occurs, the MNE_SNS technology function is output. The system then waits for acknowledgement of the M function.<br>1: After the measuring signal is received, the MNE_SNS technology function is output immediately. After the residual path is removed, the system stops and waits for acknowledgement of the M function. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter can only be used in “Reject rest block” mode (P-CHAN-00149 [⏵ 100]) and must therefore be set to the value 1).<br>Parameterisation example:<br>`# Reject rest block`<br>`kasto_reject_rest_block        1`<br>`# Output an MNE_SNS technology function immediately after measuring signal`<br>`kasto_techno_output_with_signal    1` |

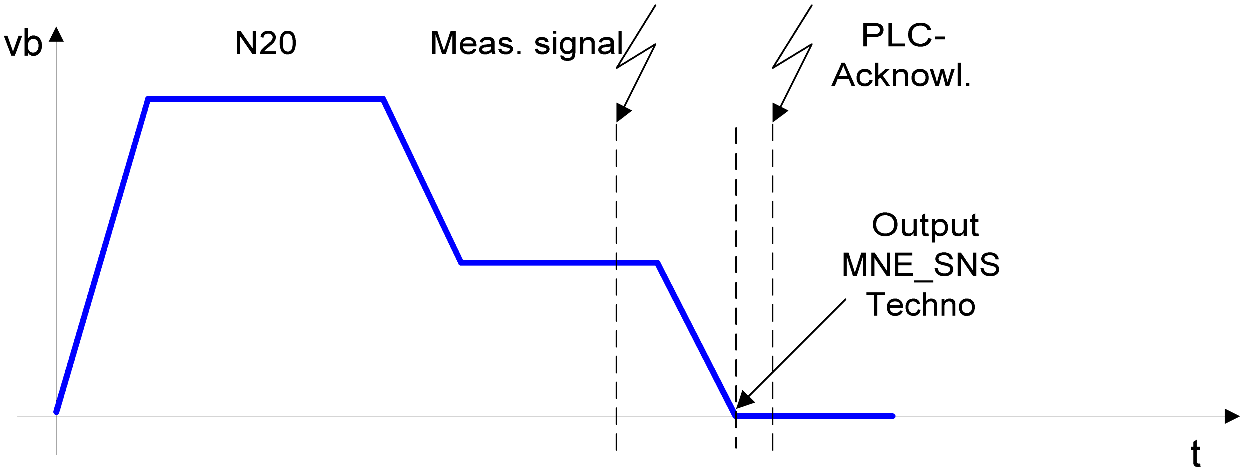


Fig. 21: Output M function after measuring signal and remove residual path (P-CHAN-00435 = 0)

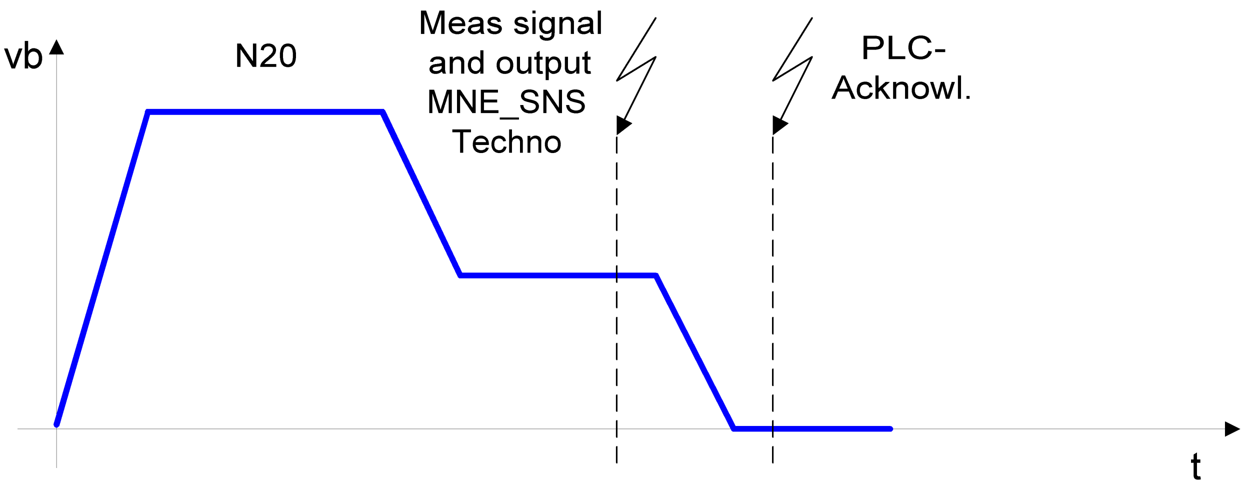


Fig. 22: Output M function directly after measuring signal (P-CHAN-00435 = 1)

# 1.6 Settings for measurement

## 1.6.1 Measurement type (P-CHAN-00057)

| **P-CHAN-00057** | **Predefine measurement type** |
|---|---|
| Description | Seven different measurement types are available. This element sets the required measurement type. |
| Parameter | messtyp |
| Data type | UNS16 |
| Data range | 1*: Measurement travel with at least one axis,<br>Measurement feed programmable by F word.<br>2*: Measurement travel with exactly one axis.<br>Measurement feed is specified in the axis data list.<br>3: Measurement travel with at least one axis,<br>Measurement feed programmable by F word,<br>optionally continue motion up to the target point.<br>4: Measurement travel only with the maximum of 3 main axes,<br>Measurement feed programmable by F word.<br>5: Interruptible measurement travel with at least one axis,<br>Measurement feed programmable by F word.<br>6: Interruptible measurement travel with at least one SERCOS axis,<br>Measurement feed programmable by F word.<br>7*: Measurement travel (G100) by moving to a fixed stop with at least one axis,<br>Measurement feed programmable by F word. |
| Dimension | ---- |
| Default value | 1 |

| | |
|---|---|
| Remarks | * for these measurement types a measurement travel is also possible with independent axes.<br>This measurement type can be changed at any time in the NC program with #MEAS MODE or #MEAS [TYPE..]. Further information is described in detail in [PROG].<br>Parameterisation example: Select measurement type 3 for a measurement travel with two axes and then continue motion up to the programmed target point.<br>*Measurement type 3* |

## 1.6.2 Error reaction with measurement type 1 (P-CHAN-00176)

| **P-CHAN-00176** | **Error reaction with measurement type 1** |
|---|---|
| Description | With measurement type 1, this parameter can influence the error reaction when the measurement signal is missing from the measurement block. |
| Parameter | meas_error_no_signal |
| Data type | BOOLEAN |
| Data range | 0: No output of an error message if no measurement signal is received (default).<br>1: Output of an error message if no measurement signal is received. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*meas_error_no_signal 1* |

## 1.6.3 Active delay with measurement signal (P-CHAN-00214)

| **P-CHAN-00214** | **Active delay with measurement signal** |
|---|---|
| Description | In general, the effective feedhold deceleration is selected in all motion blocks by the channel parameter P-CHAN-00097 [▶ 136] . The controller uses this deceleration to brake motion by default, even after the measurement signal is activated (e.g. measuring probe). Set the parameter to 1 if braking is required on activation of the measurement signal with rapid traverse deceleration. |
| Parameter | meas_deceleration_mode |
| Data type | UNS16 |
| Data range | 0: When the measurement signal is received, the motion is braked with feedhold deceleration (P-AXIS-00053 for non-linear slope) (default).<br>1: When the measurement signal is received, the motion is braked with rapid traverse deceleration (P-AXIS-00004 for non-linear slope). |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*meas_deceleration_mode 1* |

## 1.6.4 Error reaction with measurement type 7 (P-CHAN-00266)

| **P-CHAN-00266** | **Error reaction with measurement type 7** |
|---|---|
| Description | With measurement type 7 (measuring with motion to fixed stop), this parameter influences the error response when the fixed stop is not detected in the measurement block.<br>If no error message is output when the fixed stop is not found, the CNC still goes to the position of the current axis actual value at the end of the measurement travel in order to eliminate a possible position lag (e.g. if the specified position lag is not reached completely). |

| Parameter | meas_fixed_stop_no_error |
|---|---|
| Data type | BOOLEAN |
| Data range | 0: Output of an error message if the fixed stop is not detected (default).<br>1: No output of an error message if the fixed stop is not detected. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*meas_fixed_stop_no_error     1* |

## 1.6.5 Automatic travel range limit (P-CHAN-00268)

| **P-CHAN-00268** | **Automatic motion path limiting** |
|---|---|
| Description | The programmed target positions of a measurement block (G100) must be inside the software limit switches (SLS), otherwise the measurement motion is not started or executed. With specific measurement processes, the position of the PCS target positions is unknown. Movement should continue along the programmed direction until the probe responds. The parameter P-CHAN-00268 can activate the automatic limit of the measurement process. If no probe signal is detected during the measurement travel, the CNC stops the movement before the software limit switch or IMCS rectangular workspace or protection area (as of V3.1.3079.22). An error message is output with ID 50706 .<br>This function is available with measuring methods 1, 2, 3, and 4. Measurement offsets set in the axis parameters P-AXIS-00114 or P-AXIS-00467 are only effective if the CNC does not execute any limiting of the measurement travel. |
| Parameter | meas_soft_limit_move_path |
| Data type | BOOLEAN |
| Data range | 0: Programmed measurement motions with target position outside software limit switches are not started; a software limit switch error message is output.<br>1: Programmed measurement motions with target outside software limit switches are started without error message. If the measurement is not executed by the time the end of the motion path (SLS) is reached, the CNC stops the movement and error message ID 50706 is output. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*meas_soft_limit_move_path 1* |

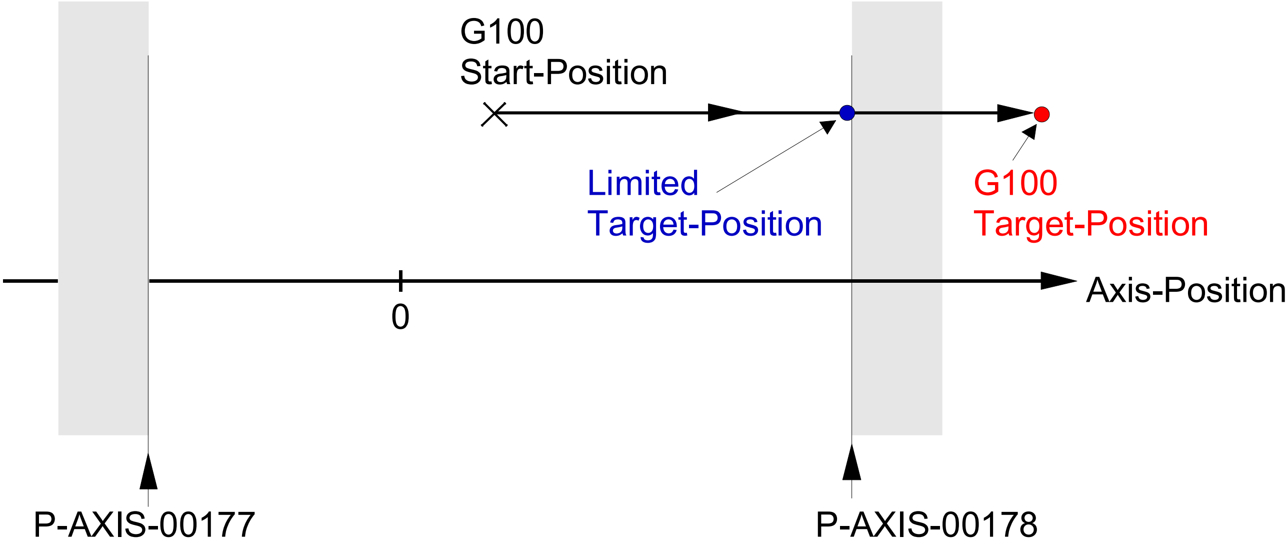
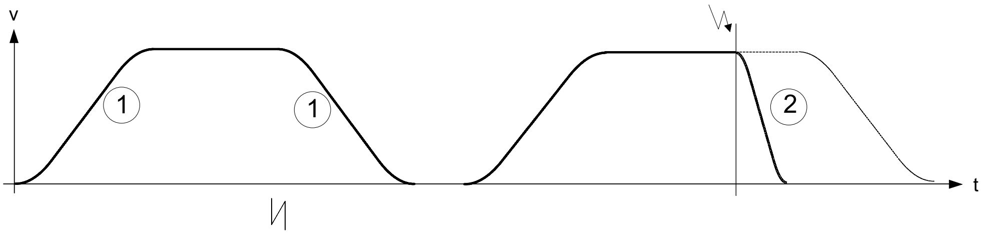


Fig. 23: Motion path limiting during measurement run

## 1.6.6 Active dynamic for measurement run (P-CHAN-00296)

| **P-CHAN-00296** | **Active dynamic for measurement run** |
|---|---|

| | |
|---|---|
| Description | The measurement travel profile planning is executed based on G00 dynamic parameters. In general, this ensures that the system can stop the motion in good time if the probe deflection is limited.<br>If the measurement travel profile planning must be executed with G01 values, set the parameter to 1.<br>The deceleration ramp used when the measuring signal is received is always dependent on P-CHAN-00097 [▶ 136] and P-CHAN-00214 [▶ 108]. |
| Parameter | meas_use_std_dynamic |
| Data type | BOOLEAN |
| Data range | 0: The measurement travel profile planning is executed based on rapid traverse acceleration values (G00) dependent on P-CHAN-00097 [▶ 136] and P-CHAN-00214 [▶ 108] . The CNC functions for acceleration and ramp time weighting are not effective here.<br>1: The dynamics of the measurement travel is dependent on P-CHAN-00097 [▶ 136] and P-CHAN-00214 [▶ 108] and is executed based on the dynamics of feed blocks (G01). CNC functions for acceleration and ramp time weighting can be used here.<br>The deceleration ramp used when the measurement signal is received is always a_feedh. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*meas_use_std_dynamic 1*<br>The effective dynamics dependent on the parameters P-CHAN-00097 [▶ 136] and P-CHAN-00214 [▶ 108] are shown in the table below.<br> |

### Dynamics for measurement travel with non-linear slops

| Active dynamic for measurement run<br>P-CHAN-00296 [▶ 110] | Active delay with measuring signal<br>P-CHAN-00214 [▶ 108] | Valid deceleration ramp at FEED-HOLD<br>P-CHAN-00097 [▶ 136] | Profile dynamic (1) (limiting algorithm) measurement signal inactive | Deceleration ramp (2) (limiting algorithm) measurement signal active |
|---|---|---|---|---|
| 0 | 0 | 0 | Acceleration in rapid traverse<br>P-AXIS-00004 | Acceleration in rapid traverse<br>P-AXIS-00004 |
| 0 | 0 | 1 | Acceleration in rapid traverse<br>P-AXIS-00004 | Deceleration at feed-hold<br>P-AXIS-00053 |
| 0 | 1 | X | Acceleration in rapid traverse<br>P-AXIS-00004 | Acceleration in rapid traverse<br>P-AXIS-00004 |

| | | | | |
|---|---|---|---|---|
| 1 | X | X | Acceleration at machining feedrate<br>P-AXIS-00001 or delay at machining feedrate P-AXIS-00002 | Deceleration at feedhold<br>P-AXIS-00053 |

**Dynamics for measurement travel with non-linear slops**

| Active dynamic for measurement run<br>P-CHAN-00296 [▶ 110] | Active delay with measuring signal<br>P-CHAN-00214 [▶ 108] | Valid deceleration ramp at FEEDHOLD<br>P-CHAN-00097 [▶ 136] | Profile dynamic (1) measurement signal inactive | Deceleration ramp (2) measurement signal active |
|---|---|---|---|---|
| 0 | 0 | 0 | Acceleration in rapid traverse<br>P-AXIS-00005/ P-AXIS-00006<br>or delay at machining feedrate<br>P-AXIS-00280/ P-AXIS-00281 | Acceleration in rapid traverse<br>P-AXIS-00005/ P-AXIS-00006<br>or delay at machining feedrate<br>P-AXIS-00280/ P-AXIS-00281 |
| 0 | 0 | 1 | Acceleration in rapid traverse<br>P-AXIS-00005/ P-AXIS-00006<br>or delay at machining feedrate<br>P-AXIS-00280/ P-AXIS-00281 | Deceleration at feedhold<br>P-AXIS-00024 |
| 0 | 1 | X | Acceleration in rapid traverse<br>P-AXIS-00005/ P-AXIS-00006<br>or delay at machining feedrate<br>P-AXIS-00280/ P-AXIS-00281 | Acceleration in rapid traverse<br>P-AXIS-00004 |
| 1 | X | X | Acceleration at machining feedrate<br>P-AXIS-00011/ P-AXIS-00012<br>or delay at machining feedrate<br>P-AXIS-00282/ P-AXIS-00283 | Deceleration at feedhold<br>P-AXIS-00024 |

## 1.6.7 Including ball radius in PCS measurement value (P-CHAN-00311)

| **P-CHAN-00311** | **Include ball radius in calculation of PCS measured value** |
|---|---|

| | |
|---|---|
| Description | This parameter controls the inclusion of the touch probe ball radius in the PCS measured value (V.A.MEAS.PCS.VALUE.<*Axis*>), see also the chapter on Programming in [FCT-C4]). |
| Parameter | meas_pcs_value_no_probe_radius |
| Data type | BOOLEAN |
| Data range | 0: The ball radius of the probe is included in the PCS measured value (default). *<br>1: The ball radius of the touch probe is not included in the PCS measured value. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | * The ball radius can only be considered if the probe is defined as a separate tool and is selected by the T/D word before the measurement travel. Additionally the ball radius must be entered as 'tool radius' in the tool data record. |

## 1.6.8 Allow already actuated probes when starting a measurement run (P-CHAN-00312)

| P-CHAN-00312 | Allowing already-actuated measuring probes when starting a measurement run |
|---|---|
| Description | If this parameter is initialised with 0 (default value), the error message ID 50052 is output at the start of a measurement travel with the actuated probe. The measurement travel is cancelled and the CNC assumes the error state.<br>When this parameter is initialised with 1 and the measuring probe is already actuated, the measurement travel is ended immediately and without error.<br>With Measuring travel to fixed stop combined with a parameterised minimum distance before detecting the fixed stop (P-AXIS-00776, P-AXIS-00777), this parameter suppresses the output of error message ID 51046 . The measuring travel is then terminated but the decoder variable V.A.MERF.* Then has the value 0. |
| Parameter | meas_allow_actuated_probe |
| Data type | BOOLEAN |
| Data range | 0: Actuated probe is not allowed at the start of the measurement travel (default).<br>1: Actuated probe is allowed at the start of the measurement travel. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.6.9 Measurement path without motion path (P-CHAN-00313)

| P-CHAN-00313 | Measurement run without programmed motion path |
|---|---|
| Description | During a measurement travel G100, a check is made along the path whether path movement is programmed in the measurement block. If not, the CNC generates the error message P-ERR-50067. This plausibility check can be disabled by the parameter.<br><br>Example:<br>N010 G01 X100 F100<br>N020 G100 X100 (Error P-ERR-50067 ) |
| Parameter | meas_error_no_move_path |
| Data type | BOOLEAN |
| Data range | 0: No error message output.<br>1: Error message output when a measurement block contains no path motion (default). |
| Dimension | ---- |

| | |
|---|---|
| Default value | 1 |
| Remarks | This parameter is available as of CNC Build V2.11.2025.03 and higher. |

## 1.6.10 Switch over between physical and clone axes during a measurement run (P-CHAN-00326)

| **P-CHAN-00326** | **Block switch-over between physical and clone axes during a measurement run.** |
|---|---|
| Description | This parameter blocks a switch-over between physical and clone axes during a measurement travel. |
| Parameter | meas_dont_allow_suspend_axis_output |
| Data type | BOOLEAN |
| Data range | 0: Switch-over can also be commanded during a measurement travel. However, switch-over only takes place after the measurement travel ends (default).<br>1: An error message is output if a switch-over command occurs during a measurement travel. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.6.11 Axes, measurement values and offsets in CS (P-CHAN-00327)

| **P-CHAN-00327** | **Axes, measurement values and offsets in CS** |
|---|---|
| Description | During 2.5D measurements, measurement values are only provided and measurement offsets calculated for the axes programmed in the measurement block.<br>Regardless of programming for measurement in a coordinate system (CS), measurement values are always provided for the first 3 machine axes, but measurement offsets are only calculated for programmed axes.<br>This parameter can extend measurement in the coordinate system (CS) so that measurement values and also the calculation of the offsets are always provided for all axes in the NC channel |
| Parameter | meas_all_axes |
| Data type | BOOLEAN |
| Data range | 0: Provide measurement values for the first 3 machine axes, calculate measurement offsets only for axes programmed in measurement block (default).<br>1: Provide measurement values and calculate measurement offsets for all axes in the channel. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.7 Global program effectiveness of P parameters (P-CHAN-00067)

| P-CHAN-00067 | **Global program effectiveness of P parameters** |
|---|---|
| Description | If the parameter is set to 1, the P parameters are globally effective for all programs, i.e. they are not deleted at program start. |
| Parameter | p_param_haltend |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: All P parameters are globally effective.<br>*p_param_haltend 1* |

## 1.8 Single-step operating mode (P-CHAN-00015)

| P-CHAN-00015 | **Define single-step operating mode** |
|---|---|
| Description | This parameter defines the basic behaviour of single-step mode after it is activated in the HMI.<br>All operating modes have in common that they stop before NC blocks with axis movements. In addition, you can define the behaviour with regard to further NC blocks.<br>If motion blocks are inserted by specific NC functions (TRC, SPLINE, polynomial contouring, etc.) and block boundaries are altered, these new block boundaries also determine stopping in single-step mode. |
| Parameter | einzelschrittmodus |
| Data type | SGN16 |
| Data range | -1: Single-step mode for<br>• NC motion blocks<br>0: (Default) - Single-step mode for<br>• NC motion blocks and<br>• relevant control blocks<br>1: Single-step mode for<br>• NC motion blocks and<br>• relevant control blocks and<br>• all other NC blocks * |
| Dimension | ---- |
| Default value | 0 |
| Remarks | * No stop is executed with comment lines, comment areas and hidden NC blocks (Skip). |

## 1.9 Settings for diameter programming (P-CHAN-00091)

| P-CHAN-00091 | **Set the offset scaling for diameter programming** |
|---|---|
| Description | This parameter must be set to 1 if zero offsets and reference point offsets are specified for the face turning axis in diameter coordinates. |
| Parameter | versch_im_durchm |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | Diameter programming is described in detail in [PROG].<br>Parameterisation example: Zero and reference point offsets for the face axis are specified in diameter coordinates. This results in only half the motion path in each case.<br>*versch_im_durchm     1* |

## 1.10 Settings for circle centre point correction and radius compensation

A circle that is determined in the NC program by starting point, end point and middle point is geometrically overdetermined. With imprecisely programmed circle coordinates, the radius starting and end points of the circle are different. This problem is solved by correcting the centre point and radius coordinates of the circle. The algorithm for circle centre point correction is described in greater detail in [PROG]. The correction algorithm provides a usable result even for geometrically meaningless programmed circle coordinates. In some cases the deviations between programmed and corrected centre point coordinates may be correspondingly large. Therefore, the new calculated centre point coordinates are subjected to a plausibility check.

It is verified whether the centre point offset Δm is greater than the maximum centre point offset 'mittelpkt_diff' defined as the absolute value in the channel parameters

Δm > sda_mds.mittelpkt_diff ?

and whether the centre point offset Δm is greater than the product of 'mittelpkt_faktor/1000' and the corrected radius 'radius'.

Δm > sda_mds.mittelpkt_faktor/1000 * radius ?

If this is the case, an error message is output. Otherwise, the corrected centre point coordinates are adopted. Here, 'mittelpkt_diff' must be defined as the absolute value and 'mittelpkt_faktor' as the factor (unit 0.1%).

The programmed centre point coordinates must therefore lie in a circumcircle about the corrected centre point of the circle. The radius of this circumcircle corresponds to the permissible centre point offset Δm which can be set using the two parameters 'mittelpkt_diff' and 'mittelpkt_faktor':

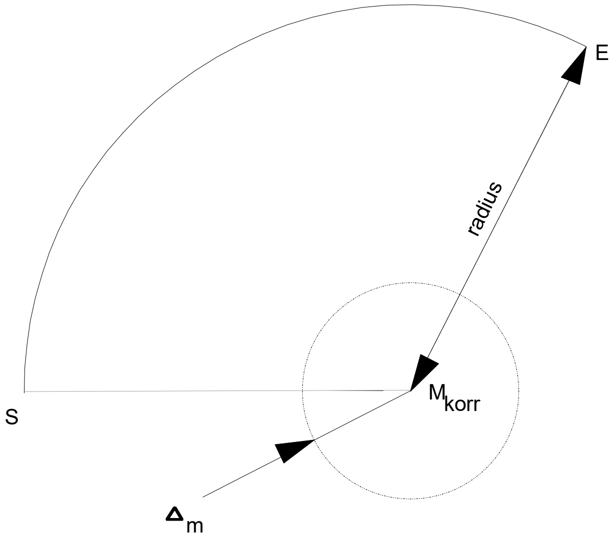


Fig. 24: Zone of permissibly programmed centre points

### 1.10.1 Permissible absolute centre point offset (P-CHAN-00059)

| **P-CHAN-00059** | **Permissible absolute centre point offset during circle programming** |
|---|---|
| Description | This parameter is used to define the absolute limit for the accuracy of centre point programming. |
| Parameter | mittelpkt_diff |
| Data type | REAL64 |
| Data range | 0 ... MAX(REAL64) |
| Dimension | 0.1µm |
| Default value | 20000 |

| | |
|---|---|
| Remarks | Parameterisation example: Permissible absolute centre point offset 0.1 mm<br>*mittelpkt_diff 1000* |

## 1.10.2 Permissible relative centre point offset (P-CHAN-00060)

| **P-CHAN-00060** | **Permissible relative centre point offset during circle programming** |
|---|---|
| Description | This parameter can be used to define the permissible relative centre point offset. |
| Parameter | mittelpkt_faktor |
| Data type | UNS16 |
| Data range | 0 < mittelpkt_faktor < MAX(UNS16) |
| Dimension | 0.1% |
| Default value | 50 |
| Remarks | Parameterisation example: The centre point offset may be 0.5% of the corrected circle radius.<br>*mittelpkt_faktor 5* |

## 1.10.3 Maximum permissible absolute radius difference (P-CHAN-00171)

| **P-CHAN-00171** | **Maximum permissible absolute radius difference during circle programming** |
|---|---|
| Description | This parameter defines the permissible absolute radius difference of the circle radii between the starting point and the target point. If the radius difference of a programmed circle is below the limit defined by the parameter, the circle radius is transferred linearly to the target radius starting from the starting radius; otherwise an error message is output. If the parameter is not assigned or is assigned the value 0, the function is inactive |
| Parameter | max_radius_diff_circle |
| Data type | UNS32 |
| Data range | 0 < max_radius_diff_circle < 10000 |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | CHAN-00171 is only active only when circle centre point compensation is inactive (G164).<br>Parameterisation example: The circle radius error may be 0.01 mm.<br>*max_radius_diff_circle 100* |

## 1.10.4 Maximum permissible percentage radius difference (P-CHAN-00172)

| **P-CHAN-00172** | **Maximum permissible percentage radius difference during circle programming** |
|---|---|
| Description | This parameter specifies the permissible relative radius difference referred to the circle radius per mil. If the relative radius difference of a programmed circle is below the limit defined by the parameter, the circle radius is transferred linearly to the target radius starting from the starting radius, otherwise an error message is output.<br>The effectively active circle radius difference error is considered by P-CHAN-00171 [▶ 118] . |
| Parameter | max_proz_radius_diff_circle |
| Data type | UNS32 |
| Data range | 0 < max_proz_radius_diff_circle < 500 |
| Dimension | 0.1% |

| | |
|---|---|
| Default value | 0 |
| Remarks | P-CHAN-00172 is only effective in combination with G164 and the channel parameter P-CHAN-00171 [▶ 118] .<br>Parameterisation example: The percentage circle radius error may amount to maximum 1%.<br>Circle radius 100 mm -> permissible relative circle radius difference 1 mm.<br>*max_proz_radius_diff_circle 10* |

## 1.11 Settings for RESET

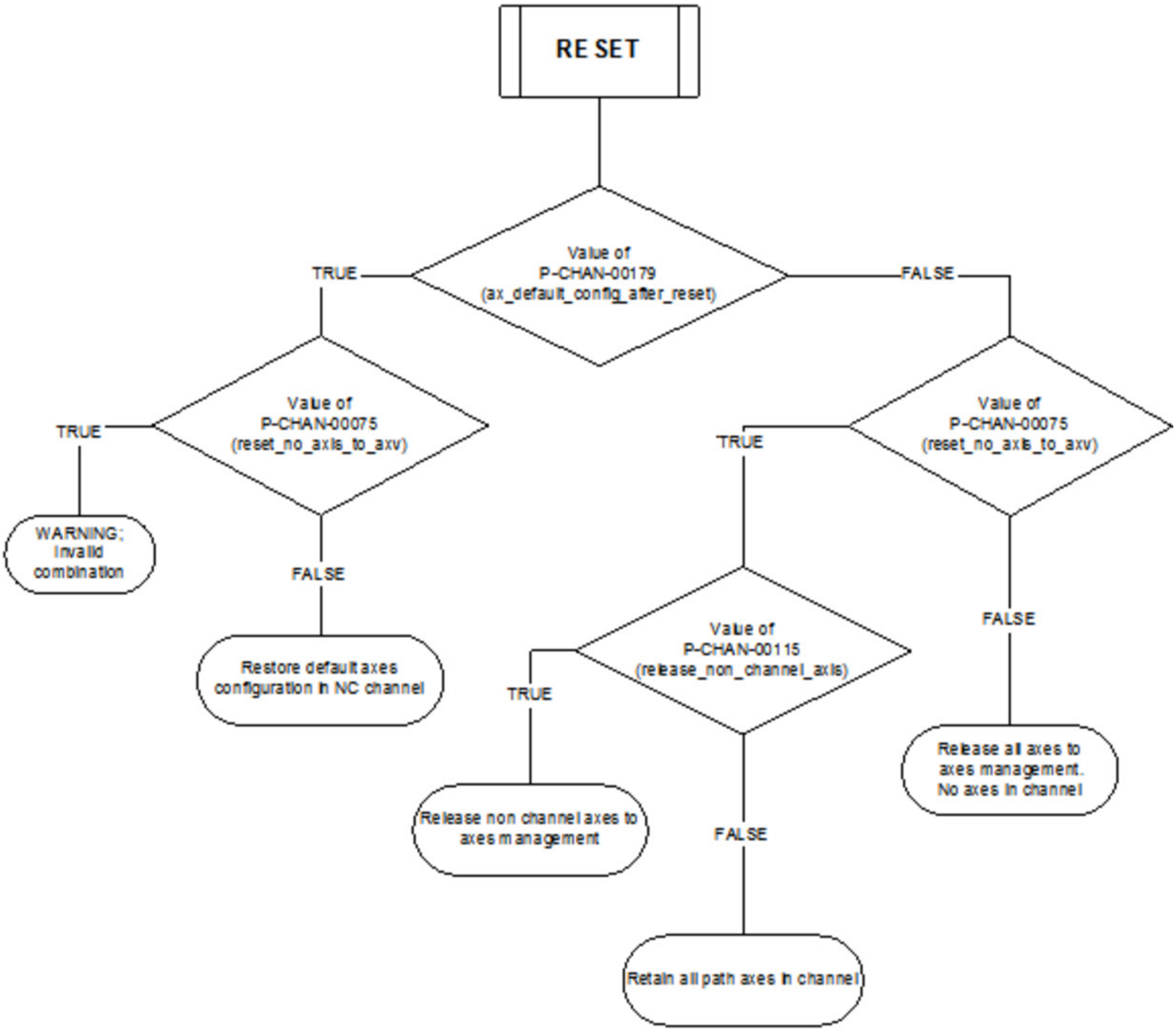

RESET
Value of P-CHAN-00179 (ax_default_config_after_reset)
TRUE
FALSE
Value of P-CHAN-00075 (reset_no_axis_to_axv)
TRUE
FALSE
WARNING: invalid combination
Restore default axes configuration in NC channel
Value of P-CHAN-00075 (reset_no_axis_to_axv)
TRUE
FALSE
Value of P-CHAN-00115 (release_non_channel_axis)
TRUE
FALSE
Release all axes to axes management. No axes in channel
Release non channel axes to axes management
Retain all path axes in channel

Fig. 25: Restore axis configuration after RESET depending on influencing parameters

### 1.11.1 Default axis configuration in the channel after reset (P-CHAN-00179)

| P-CHAN-00179 | Restore default axis configuration in the channel after reset |
|---|---|
| Description | This parameter defines whether the axis configuration is to be restored according to the channel parameter list after a reset in the NC channel. |
| Parameter | ax_default_config_after_reset |
| Data type | BOOLEAN |
| Data range | 0: Handling path axes as per P-CHAN-00075 [▸ 121] and P-CHAN-00115 [▸ 121] (default).<br>1: Restore basic configuration of axes after RESET according to channel parameter list. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | It only makes sense to use this parameter if P-CHAN-00075 [▸ 121] (reset_no_axis_to_axv) is assigned FALSE.<br>See also P-AXIS-00251 |

## 1.11.2 Axis release after Reset (P-CHAN-00075)

| P-CHAN-00075 | Axis release after reset |
|---|---|
| Description | This parameter defines whether all axes of a channel are released to axis management at RESET. |
| Parameter | reset_no_axis_to_axv |
| Data type | BOOLEAN |
| Data range | 0: All axes are released to axis management at reset (default).<br>1: All axes remain in the channel at reset. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | See also P-AXIS-00251 |

## 1.11.3 Release of non-channel axes after Reset (P-CHAN-00115)

| P-CHAN-00115 | Release of non-channel axes after reset |
|---|---|
| Description | This parameter defines whether all path axes which do not belong to the basic channel configuration are released to the axis management at RESET. This does not apply to SAI axes which are always released to the axis management at reset. |
| Parameter | release_non_channel_axis |
| Data type | BOOLEAN |
| Data range | 0: Non-channel axes remain in the channel at reset (default).<br>1: Non-channel axes are released to the axis management at reset. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | It only makes sense to use this parameter if P-CHAN-00075 [▸ 121] (reset_no_axis_to_axv) is assigned TRUE.<br>See also P-AXIS-00251 |

# 1.12 Retention of the last axis configuration (P-CHAN-00460)

| P-CHAN-00460 | Retention of the last axis configuration in the channel |
|---|---|
| Description | When this parameter is set, the default axis configuration is only activated at initial program start according to the channel parameter list and after controller start-up. At every subsequent main program start, the last active configuration of the previous NC program is retained.<br>Actions such as a change in operation mode, RESET or program end (M30) have no further implicit influence on the current axis configuration. SAI axes and spindle axes exchanged in the axis group also remain in the channel (P-CHAN-00251). After an initial program start, the user must ensure that a consistent axis configuration exists by programming accordingly.<br>When P-CHAN-00460 is set, the parameters P-CHAN-00075 [▸ 121], P-CHAN-00115 [▸ 121] and P-CHAN-00179 [▸ 120] have no effect in conjunction with RESET or program end. |
| Parameter | retain_last_ax_config |
| Data type | BOOLEAN |

| | |
|---|---|
| Data range | 0: Restore the default axis configuration according to the channel parameter list after program start taking into account P-CHAN-00075 [▸ 121] , P-CHAN-00115 [▸ 121] and P-CHAN-00179 [▸ 120] (default).<br>1: Cross-program retention of the last axis configuration in the channel. Not influenced by program start, M30 or RESET. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | An attempt to restore the default axis configuration according to the channel parameter list after a program error should only be made by the explicit programming of #AX DEF DEFAULT. If this is not successful, the controller can only be returned to its initial state by a repeated start-up (cold start). |

## 1.13 Maximum channel override (P-CHAN-00056)

| **P-CHAN-00056** | **Limiting maximum channel override** |
|---|---|
| Description | This parameter limits the maximum override in the channel. |
| Parameter | max_vb_override |
| Data type | UNS16 |
| Data range | 0 ≤ max_vb_override ≤ 2000 (maximum value of channel override, application-specific) |
| Dimension | 0.1% |
| Default value | 1000 |
| Remarks | Parameterisation example: The limit of the maximum channel override is 150%.<br>*max_vb_override 1500* |

## 1.14 Sampling of override (P-CHAN-00065)

| **P-CHAN-00065** | **Override scanning** |
|---|---|
| Description | To avoid exciting oscillations in the machine, the override value is only adopted after the first change to calculate a new velocity. This may also occur in the same cycle.<br>The next value is only adopted after the time expires in the parameter. This avoids excitations in the machine caused by an override change. |
| Parameter | override_delay |
| Data type | UNS32 |
| Data range | 10 * cycle time < override_delay < 100 s |
| Dimension | μs |
| Default value | 10 |
| Remarks | Parameterisation example: The delay time for override scanning is set to 50 ms.<br>*override_delay 50000*<br>The smallest value that can be set in override_delay is 10 * cycle time. Therefore, a cycle time of 2 ms is equal to 20000 μs.<br>If a smaller value is set for P-CHAN-00065, the value is corrected to the minimum value. |

# 1.15 Mode of the override

## 1.15.1 Influence on feed (P-CHAN-00066)

| **P-CHAN-00066** | **Influence of override on feed** |
|---|---|
| Description | This parameter defines how feed is calculated in conjunction with path feed override.<br>This means, for example, the parameter sets whether the override bandwidth is applied to the programmed feed or to the minimum programmed feed and the maximum permissible feed in the block ($v_{max}$). The CNC calculates the feed based on the permissible axis dynamics.<br>**Value 0 for P-CHAN-00066:**<br>v = MIN ($v_{prog}$,$v_{max}$) * Override<br>If this parameter is set to 0, a change of override always leads to a change of velocity in the machine (linear correlation).<br>**Value 1 for P-CHAN-00066:**<br>v = MIN ($v_{prog}$* Override, $v_{max}$)<br>If this parameter is set to 1, the override always refers to the programmed feed. If the maximum feed in the block is smaller than the programmed feed multiplied by the current override, the current path feed is not changed when there is a change in override.<br>**Value 2 for P-CHAN-00066:**<br>Calculation is executed<br>• as for the value 0 provided there are no active functions that reduce velocity. For example, REDUCED_SPEED, REDUCED_SPEED_ZONE, IPO_ACTIVATE_TCP_VEL_LIMIT<br>• as for the value 1 provided there are no active functions that reduce velocity<br>If velocity drops are not desired for technological reasons, this parameter must be set to 1. |
| Parameter | override_weight_prog_feed |
| Data type | UNS16 |
| Data range | 0: Change in override causes a change of velocity.<br>1: An override change only causes a velocity change if the programmed feed * override < maximum permissible velocity in the block<br>2: Override effect in conjunction with reduced velocity (available as of Build V3.1.3079.42) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.15.2 Influence on acceleration (P-CHAN-00170)

| **P-CHAN-00170** | **Influence of override on acceleration** |
|---|---|

| | |
|---|---|
| Description | This parameter defines that the feed override also influences path acceleration.<br>If the parameter is set to 1, the path acceleration is influenced up to a minimum value of 10% of the override value, i.e. the path dynamics decrease as the override value decreases (motions become 'softer').<br>Acceleration is changed by the square of the override speed factor. An override value of 50% therefore results in an acceleration weighting factor of 0.25.<br>The influence acts on feed and rapid traverse blocks. If necessary, acceleration is limited as values become higher for override changes in dynamic movement phases. With non-linear profiles and short ramp times, limiting must consider the ratio of the cycle time to the ramp time.<br>If the parameter is set to 0, the value of the feed override has no influence on path acceleration. |
| Parameter | override_weight_acc |
| Data type | BOOLEAN |
| Data range | 0: An override change does not cause a change in path acceleration.<br>1: Besides producing a change in the path feed rate, an override change also causes a change in path acceleration. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Active acceleration with feedhold is not changed. |

The function is not active with an active rapid traversing interface according to P-CHAN-00181.

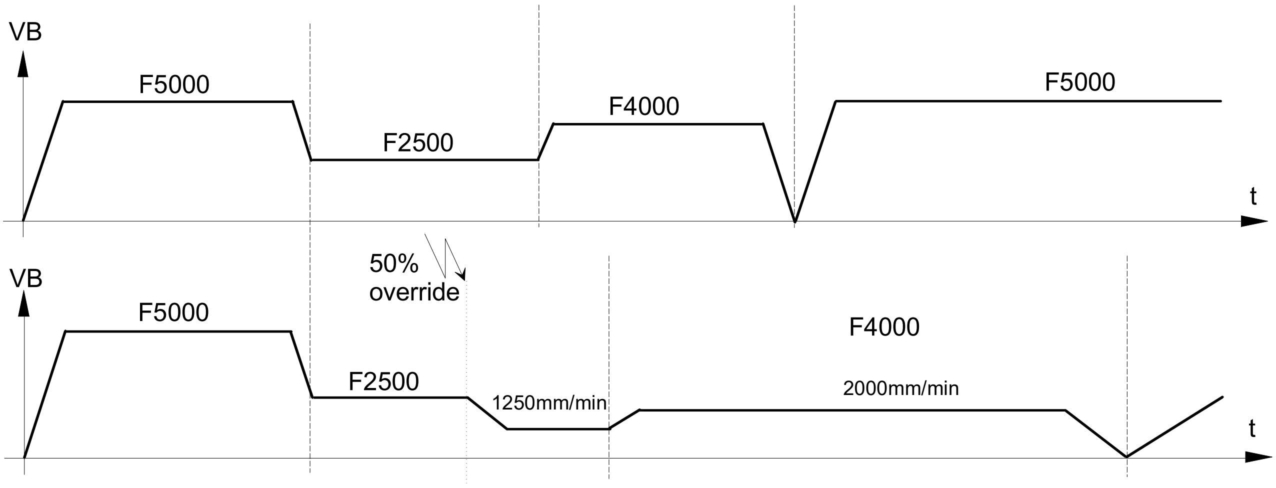


Fig. 26: Feed profile for override_weight_acc = 1 and override change

When this parameter is set, direct weighting of path acceleration via the PLC acceleration override interface is inactive.

## 1.15.3 Rapid traverse override (P-CHAN-00181)

| P-CHAN-00181 | Define and activate rapid traverse override mode |
|---|---|

| | | |
|---|---|---|
| Description | This parameter is used to define whether the override interface for influencing rapid traverse blocks is active and how rapid traverse override acts. Activating this interface deactivates the influence of acceleration as per P-CHAN-00170 [▶ 123].<br>The parameter also is active for continuous jog mode (manual mode). | |
| Parameter | g00_override_mode | |
| Data type | STRING | |
| Data range | NOT_USED or 0: | Rapid traverse override is inactive. Rapid traverse override acts on feed and rapid traverse blocks. |
| | MINIMUM_OF_G0_G1 or 1: | Rapid traverse override is active. Feed override acts on feed blocks; the minimum of feed and rapid traverse override acts on rapid traverse blocks. |
| | ACTIVE or 2: | Rapid traverse override is active. Feed override acts on feed blocks; only rapid traverse override acts on rapid traverse blocks. |
| | ACTIVE_EXTENDED or 3: | Same as under ACTIVE, rapid traverse override also acts exclusively on homing G74, measurement G100 and manual mode controlled by the CNC. |
| Dimension | ---- | |
| Default value | 0 | |
| Remarks | The channel parameter acts on the channel-specific override interface.<br>Please note in ACTIVE and ACTIVE_EXTENDED modes that G0 movements can only be influenced by the rapid traverse override. | |

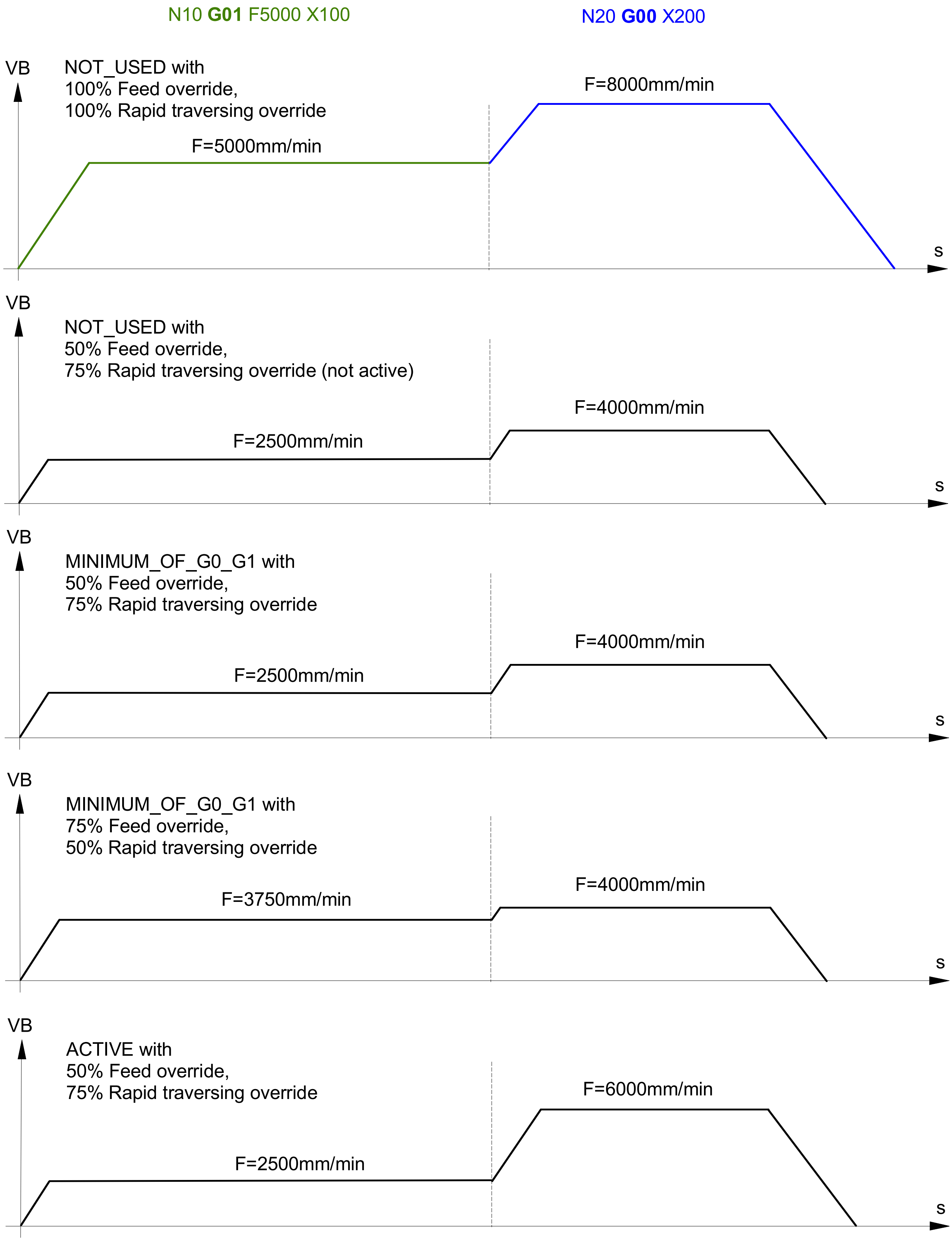


Fig. 27: Examples of rapid traverse override modes

## 1.15.4 Influence of time override on dwell time (P-CHAN-00111)

| P-CHAN-00111 | Influence of time override on dwell time |
|---|---|

| | |
|---|---|
| Description | This parameter defines whether the dwell time function is influenced by the time override real-time function. |
| Parameter | time_override_weight_dwell_time |
| Data type | BOOLEAN |
| Data range | 0: A change of override has no influence on dwell time.<br>1: The duration of a programmed dwell time changes in relation to the time override specified by the PLC. For example, this means that at 50% time override, the programmed dwell time of 1s is increased by a factor of 2 to 2s |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.15.5 Influence of time override on dwell time (P-CHAN-00353)

| **P-CHAN-00353** | **Influence of time override on dwell time** |
|---|---|
| Description | This parameter defines whether the dwell time function is influenced by the time override real-time function. |
| Parameter | feed_override_weight_dwell_time |
| Data type | BOOLEAN |
| Data range | 0: A change of override has no influence on dwell time.<br>1: The duration of a programmed dwell time changes in relation corresponding to the time override specified by the PLC. For example, this means that at 50% time override, the programmed dwell time of 1s is increased by a factor of 2 to 2s. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.16 Operating principle of PLC feed (P-CHAN-00102)

| **P-CHAN-00102** | **Operating principle of PLC feed** |
|---|---|
| Description | This parameter defines whether the PLC command feed is only active with feed blocks or also with rapid traverse blocks. |
| Parameter | plc_command_rapid_feed |
| Data type | BOOLEAN |
| Data range | 0: Feed commanded by PLC is only active with feed blocks (G01, G02; G03)<br>1: Feed commanded by PLC is also active with rapid traverse blocks (G00) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.17 Settings for vector dynamic limits (vector.*)

The maximum permissible velocity, acceleration and deceleration on the path are dependent on the dynamic values in the axis-specific parameter list and the programmed contour. The following parameters initialise path velocity and acceleration/deceleration default limits. Activation/deactivation and change of limits are executed by NC command (see also [PROG//Command #VECTOR LIMIT..]).

### 1.17.1 Vector acceleration limit (P-CHAN-00002)

| **P-CHAN-00002** | **Path acceleration limit** |
|---|---|

| | |
|---|---|
| Description | If no change is made to the parameter values by NC programming, this value limits path acceleration after it is activated in the NC program. |
| Parameter | vector.acceleration |
| Data type | REAL64 |
| Data range | 0 ... Maximum acceleration, application-specific |
| Dimension | mm/min$^2$ or mm/s$^2$ * |
| Default value | 100000000 |
| Remarks | * The dimension used depends on P-CHAN-00351 [▶ 128].<br>Parameterisation example:<br>`vector.acceleration 1800000`<br>The specified default value is in mm/min$^2$. |

## 1.17.2 Path deceleration limit (P-CHAN-00208)

| **P-CHAN-00208** | **Path deceleration limit** |
|---|---|
| Description | If no change is made to the parameter values by NC programming, this value limits path deceleration after it is activated in the NC program. |
| Parameter | vector.deceleration |
| Data type | REAL64 |
| Data range | 0 ... Maximum deceleration, application-specific |
| Dimension | mm/min$^2$ or mm/s$^2$ * |
| Default value | 100000000 |
| Remarks | * The dimension used depends on P-CHAN-00351 [▶ 128].<br>Parameterisation example:<br>`vector.deceleration 2000000`<br>The specified default value is in mm/min$^2$. |

## 1.17.3 Radial acceleration limit (P-CHAN-00361)

| **P-CHAN-00361** | **Radial acceleration limit** |
|---|---|
| Description | In basic setting, radial acceleration on curved paths (circles, polynomials) is limited in the NC program by the parameter value after #VECTOR LIMIT ON [..]. The value can be changed at any time in the NC program by the command #VECTOR LIMIT ON [RADIAL_ACC..]. |
| Parameter | vector.radial_acceleration |
| Data type | REAL64 |
| Data range | 0 ... Maximum acceleration, application-specific |
| Dimension | mm/min$^2$ or mm/s$^2$ * |
| Default value | 100000000 |
| Remarks | * The dimension used depends on P-CHAN-00351 [▶ 128].<br>Parameterisation example:<br>`vector.radial_acceleration 1000000`<br>The specified default value is in the unit mm/s$^2$ |

## 1.17.4 Unit for path acceleration/deceleration and jerk (P-CHAN-00351)

| **P-CHAN-00351** | **Unit for path acceleration and path deceleration** |
|---|---|

| | |
|---|---|
| Description | This channel parameter switches over the unit of acceleration/deceleration in the #VECTOR LIMIT command. The default unit is mm/min$^2$ or mm/min$^3$. This unit is switched over to mm/s$^2$ by setting the parameters to MM_S2. The arguments ACC, DEC, RADIAL_ACC and RADIAL_JERK in the NC command are then interpreted in the unit mm/s$^2$ or mm/s$^3$. |
| Parameter | vector.acc_dec_unit |
| Data type | STRING |
| Data range | MM_M2: Unit is mm/min$^2$ or mm/min$^3$<br>MM_S2: Unit is mm/sec$^2$ or mm/sec$^3$ |
| Dimension | ---- |
| Default value | MM_M2 |
| Remarks | This parameter cannot be set using the NC command #VECTOR LIMIT.<br>Parameterisation example:<br>`vector.acc_dec_unit MM_S2` |

## 1.17.5 Path velocity limit (P-CHAN-00090)

| P-CHAN-00090 | Path velocity limit |
|---|---|
| Description | If no change is made to the parameter values by NC programming, this value limits path velocity after it is activated in the NC program. |
| Parameter | vector.velocity |
| Data type | REAL64 |
| Data range | 0 ... Maximum acceleration, application-specific |
| Dimension | mm/min |
| Default value | 2000000000 |
| Remarks | Parameterisation example:<br>`vector.velocity 1500` |

## 1.17.6 Dynamic limit in the circle (P-CHAN-00350)

| P-CHAN-00350 | Implicit dynamic limit in the circle |
|---|---|
| Description | This parameter can also extend the effect of the configured or programmed limit value of path deceleration P-CHAN-00002 [▸ 127] onto circular blocks. |
| Parameter | vector.cir_radial_acc_limit |
| Data type | BOOLEAN |
| Data range | 0: Limit value of path acceleration only acts on linear blocks<br>1: Limit value of path acceleration acts on linear and circular blocks |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter cannot be set using the NC command #VECTOR LIMIT.<br>Parameterisation example:<br>`vector.cir_radial_acc_limit 1` |

# 1.18 Configuration of effector coordinate system (ECS)

When an effector coordinate system (ECS) is defined, the Z (effector) axis is determined first based on tool orientation. The positions of the X and Y (effector) axes are based on the following parameter settings. Please note that the ECS system is defined so that the Z component of the direction vector of the resulting third axis always points in a positive Z direction of the Cartesian machine coordinate system.

### 1.18.1 Select the second axis of ECS (P-CHAN-00031)

| **P-CHAN-00031** | **Select the second axis of effector coordinate systems (ECS)** |
|---|---|
| Description | This parameter defines whether the second axis for ECS is the X or Y axis. |
| Parameter | kind_of_2nd_ecs_ax |
| Data type | STRING |
| Data range | 'X', 'x', 'Y' or 'y' |
| Dimension | ---- |
| Default value | X |
| Remarks | |

### 1.18.2 Plane selection (P-CHAN-00050)

| **P-CHAN-00050** | **Select the plane for the second axis of effector coordinate systems (ECS)** |
|---|---|
| Description | This parameter selects the plane in the machine coordinate system in which this second ECS axis is located. |
| Parameter | mach_plane_of_2nd_ecs_ax |
| Data type | STRING |
| Data range | 'XY', 'xy', 'ZX', 'zx', 'YZ', 'yz' |
| Dimension | ---- |
| Default value | XY |
| Remarks | The direction of the second and third axes is calculated to obtain a right-handed system. In this case the third axis is always on the same side of the plane as the Z (effector) axis. |

### 1.18.3 Examples

The following 2 examples show the assignment of P-CHAN-00031 and P-CHAN-00050.

| | |
|---|---|
| kind_of_2nd_ecs_ax | X |
| mach_plane_of_2nd_ecs_ax | XY |

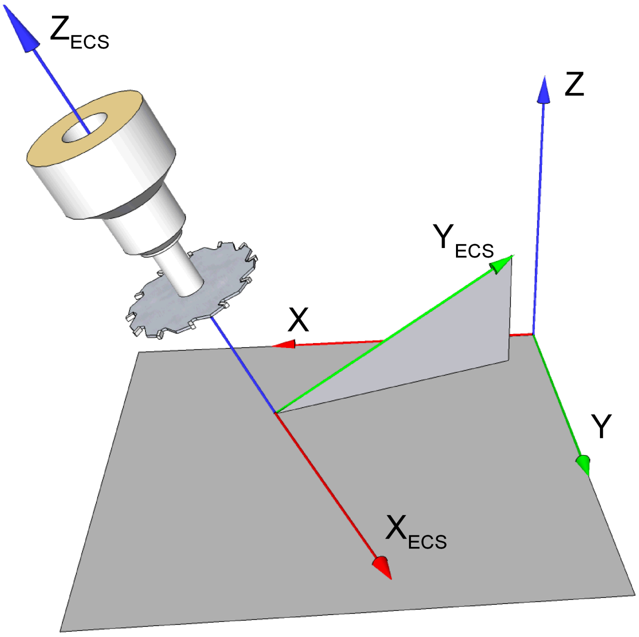


Fig. 28: Definition of effector coordinate system (ECS) via the X axis

| | |
|---|---|
| kind_of_2nd_ecs_ax | Y |
| mach_plane_of_2nd_ecs_ax | YZ |

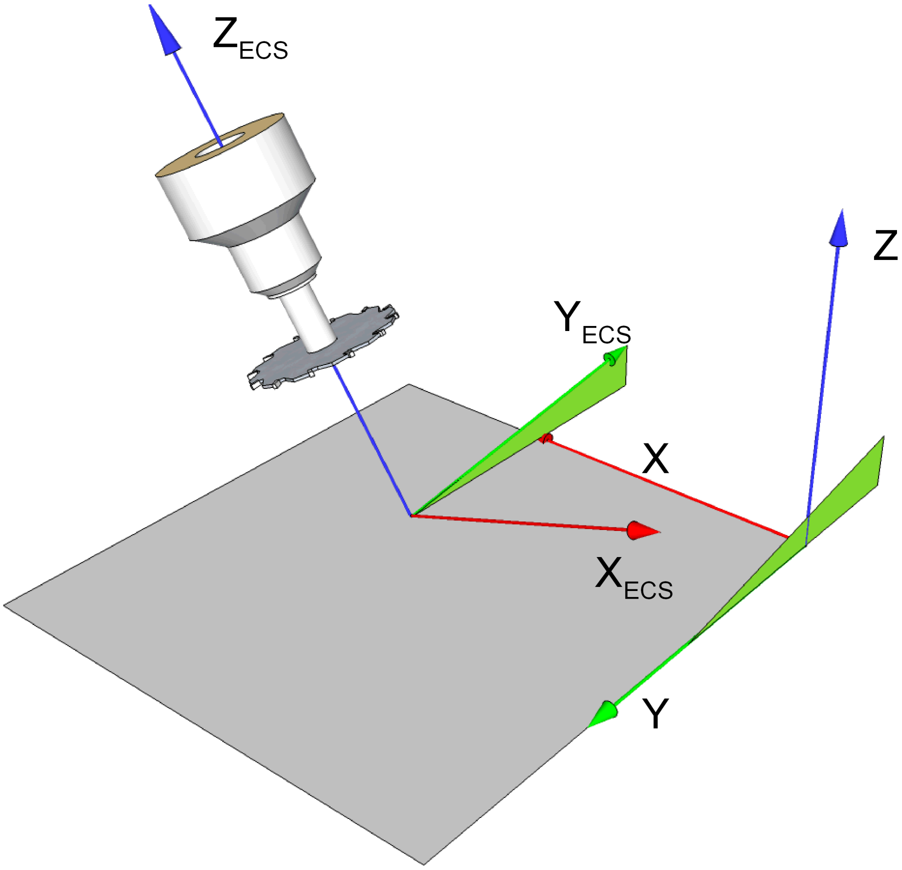


Fig. 29: Definition of effector coordinate system (ECS) via the Y axis

# 1.19 Tracking a rotary axis

The following parameters must be assigned for processes which require the tracking of the tool along the path contour at constant orientation.

## 1.19.1 Definition of the tracking axis (P-CHAN-00095)

| **P-CHAN-00095** | **Define the tracking axis (C axis tracking)** |
|---|---|
| Description | This parameter defines the axis which is orientated to the tracked path after automatic axis tracking (called C axis tracking) is selected in the NC program. |
| Parameter | log_number_tracking_axis |
| Data type | UNS16 |
| Data range | 1 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: In tracking mode the axis with the logical axis number 5 is automatically orientated at a tangential angle to the path contour.<br>*log_number_tracking_axis 5* |

ℹ **The tracking axis used must be an additional axis. This may not lie on a main axis index.**

## 1.19.2 Automatic orientation of a tracking axis (P-CHAN-00101)

| **P-CHAN-00101** | **Automatic orientation of a tracking axis** |
|---|---|
| Description | When axis tracking is used (see P-CHAN-00095 [▶ 133]), this parameter orientates the tracking axis automatically at a tangential angle to the first contour element. With modulo axes the orientation motion is executed on the shortest possible path. The motion on the path contour is always stopped during the orientation movement. |
| Parameter | auto_align_tracking_axis |
| Data type | BOOLEAN |
| Data range | 0: During the tracking operation the offset of the orientation of the tracking axis to the first contour element is retained at the time of selection. This orientation movement must be programmed to achieve a tangential orientation of the tracking axis to the first contour element.<br>1: When selected the tracking axis is orientated automatically tangentially to the first contour element. No orientation movement must be programmed. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.19.3 Selection of tracking mode without stopping (P-CHAN-00109)

| **P-CHAN-00109** | **Select tracking mode without stopping** |
|---|---|
| Description | If the tracking axis (see P-CHAN-00095 [▶ 133]) already has the required orientation to the first contour element when tracing mode is selected and this parameter is set, the motion is smoothed in the following block. |
| Parameter | fast_tracking_transition |
| Data type | BOOLEAN |

| | |
|---|---|
| Data range | 0: When tacking mode is selected, the motion is stopped and, if necessary, the orientation movement is executed.<br>1: If the orientation of the tracking axis is correct, the motion is smoothed in the following block without stopping. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.19.4 Suppress a rotary CS offset (P-CHAN-00144)

| **P-CHAN-00144** | **Suppress a rotary CS offset in tracking mode** |
|---|---|
| Description | When the coordinate system (#CS ON[]) is active, a rotation about the Z axis is included as offset for the tracking axis for automatic tracking and orientation. If this is not required, this parameter suppresses the inclusion of the rotary coordinate system offset in the calculation. |
| Parameter | suppress_cs_tracking_offset |
| Data type | BOOLEAN |
| Data range | 0: Rotary coordinate system (#CS) offset is included in the tracking axis.<br>1: Rotary coordinate system (#CS) offset is not included in the tracking axis. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.19.5 Tracking axis in the workpiece (P-CHAN-00185)

| **P-CHAN-00185** | **Define the position of the tracking axis in the workpiece** |
|---|---|
| Description | In the default setting, the tracking axis is positioned in the tool, i.e. the tracking axis rotates the tool. This parameter can adjust the tracking axis for a machine structure where the workpiece is rotated through a C axis in conjunction with kinematic transformation. The tool axis must then be vertical to the XY machining plane. |
| Parameter | tracking_axis_rot_wc |
| Data type | BOOLEAN |
| Data range | 0: The tracking axis is an axis in the tool (default).<br>1: The tracking axis is an axis in the workpiece. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.19.6 Including a contour rotation (P-CHAN-00215)

| **P-CHAN-00215** | **Including a contour rotation in the tracking axis** |
|---|---|
| Description | For a contour rotated in the plane with #ROTATION ON[], the rotation angle about the Z axis is not included in the tracking axis. If this is required, this parameter activates the inclusion of the rotational offset in the calculation.<br>This function is only useful if automatic tracking (P-CHAN-00101 [▶ 133]) is inactive for technological reasons. |
| Parameter | consider_rot_tracking_offset |
| Data type | BOOLEAN |
| Data range | 0: Contour rotation (#ROTATION) is not included in the tracking axis (default).<br>1: Contour rotation (#ROTATION) is included in the tracking axis. |
| Dimension | ---- |

| | |
|---|---|
| Default value | 0 |
| Remarks | |

## 1.19.7 Suppress vector dynamic limitation (P-CHAN-00265)

| P-CHAN-00265 | Suppress vector dynamic limitation in aligning blocks in tracking mode |
|---|---|
| Description | The programmable dynamic limitation functions are also effective via #VECTOR LIMIT[...] for inserted aligning blocks. In combination with kinematic transformations, this can lead to undesirable delays caused by slow alignment movements when a virtual C tracking axis is used. This response can be avoided by the following parameter. |
| Parameter | suppress_tracking_vec_limit |
| Data type | BOOLEAN |
| Data range | 0: Vector dynamic limitation is active for aligning blocks (default).<br>1. Vector dynamic limitation via #VECTOR LIMIT[...] is not active for aligning blocks. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.19.8 PCS smoothing function for tracking axis (P-CHAN-00281)

| P-CHAN-00281 | PCS smoothing function for tracking axis |
|---|---|
| Description | The path velocity must be as constant as possible to ensure a favourable removal process with certain technologies. When axis tracking is used (see P-CHAN-00095 [▸ 133]), the controller must reduce the path velocity at block transitions due to the required tracking axis movement.<br>This parameter activates a smoothing function for the tracking axis. This function is effective for all movements of the tracking axis and results in a greater angular deviation of this axis. In general, this results in a path velocity profile with a lower speed reduction due to discontinuous block transitions. The dynamic loading of the tracking axis is higher and the parameterised dynamic limits of *v* and *a* are maintained. |
| Parameter | filter_tracking_axis |
| Data type | BOOLEAN |
| Data range | 0: Smoothing function for C axis tracking not active (default).<br>1: Smoothing function for C axis tracking active. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

# 1.20 Switch the default feed axis (P-CHAN-00096)

| P-CHAN-00096 | Define the weakest axis as the default feed axis |
|---|---|
| Description | This parameter defines the valid default setting for feed axes.<br>If the parameter is set to 0, the default feed axes in the the axis-specific settings (gruppe [i].*) apply after the controller is started.<br>If the parameter is set to 1, the axes with the longest traverse time ('weakest axis') moves automatically at the programmed feed rate (F word). All the other axes are treated as tracking axes. |
| Parameter | feed_to_weakest_axis |
| Data type | BOOLEAN |

| | |
|---|---|
| Data range | 0: Default feed axes of the axis-specific setting are valid.<br>1: Weakest axis is default feed axis. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.21 Valid deceleration ramp at FEEDHOLD (P-CHAN-00097)

| P-CHAN-00097 | Valid deceleration ramp at FEEDHOLD |
|---|---|
| Description | This parameter defines the deceleration ramp used when FEEDHOLD is active. |
| Parameter | use_drive_curr_limit |
| Data type | BOOLEAN |
| Data range | 0: When FEEDHOLD occurs, deceleration takes place using the currently valid deceleration rate.<br>1: With FEEDHOLD, the parameterised deceleration in P-AXIS-00024 and the ramp time set in P-AXIS-00081 are used for deceleration. If these two parameters are not set, deceleration takes place at the current deceleration value (P-AXIS-00053). |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.22 Switch over the meaning of M functions (P-CHAN-00098)

| P-CHAN-00098 | Switching the meaning of M functions M3/M4/M5/M19 |
|---|---|
| Description | According to DIN66025, certain M functions have different meanings depending on the machining method and the type of machine.<br>With metal-cutting machines (e.g. milling/turning/drilling), M3/M4/M5/M19 are permanently assigned to the spindle functions (direction of rotation, stopping, positioning).<br>With machining methods such as plasma/laser cutting or wire erosion, the mentioned M functions are used to control other technology functions. To ensure free assignment, the meaning of the M functions M3/M4/M5/M19 can be switched over with this parameter. |
| Parameter | spindle_m_fct_free |
| Data type | BOOLEAN |
| Data range | 0: M3/M4/M5/M19 are permanently assigned to the spindle M functions (default when spindles are configured).<br>1: M3/M4/M5/M19 are freely available for other technology functions. They must be defined in m_synch[i] in the channel parameters. Then the M functions are not spindle functions any more. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | When machining and cutting processes are combined on the same machine, it is possible to switch over the meaning of the M functions M3/M4/M5/M19 in the NC program using the variable V.SPDL.M_FCT_FREE [PROG]! |

## 1.23 Effectiveness of tool compensation data (P-CHAN-00100)

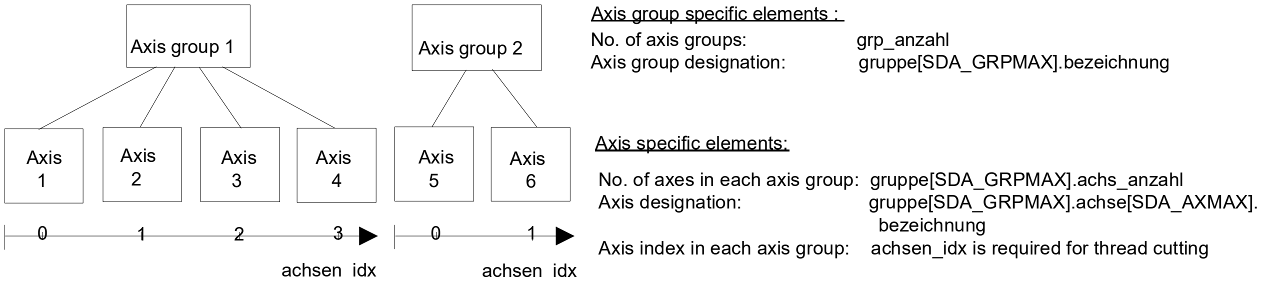
| P-CHAN-00100 | Time of effectiveness of tool compensation data |
|---|---|
| Description | This parameter defines the time of the effectiveness of the tool compensation data (tool length compensation, tool position compensation) programmed by the D word |
| Parameter | move_tool_offsets_directly |
| Data type | BOOLEAN |
| Data range | 0: The compensation movement to consider the new tool compensation data only takes place after the D word for an axis when the next traverse information **absolute** (G90) is programmed (default).<br>1: The compensation movement in the corresponding axes is executed immediately with the D word without programming an absolute path condition (in compliance with DIN 66025). |
| Dimension | ---- |
| Default value | 0 |
| Remarks | The parameter has no effect if the kinematic transformation is active (#TRAFO ON). New tool compensation data are only considered with the next absolute travel information. |

# 1.24 Configuration of path axes

This section describes the elements for configuring path axes. Path axes are organised into axis groups in the channel.

The example below depicts an axis configuration with two axis groups. In addition, it shows all the relevant elements in the channel parameters list.

> Currently, only path axis configurations are permitted with one axis group.
>
> **Axes of the 'spindle' type may not be specified as path axes since they are located in a spindle interpolator on start-up.**



Axis group specific elements :

| | |
|---|---|
| No. of axis groups: | grp_anzahl |
| Axis group designation: | gruppe[SDA_GRPMAX].bezeichnung |

Axis specific elements:

| | |
|---|---|
| No. of axes in each axis group: | gruppe[SDA_GRPMAX].achs_anzahl |
| Axis designation: | gruppe[SDA_GRPMAX].achse[SDA_AXMAX].bezeichnung |
| Axis index in each axis group: | achsen_idx is required for thread cutting |

Fig. 30: Example of axis configuration with two axis groups.

The example in the figure above results in the following assignment of elements in channel parameters:

| Axis | Axis group index<br>'achsen_idx' | Axis identifier<br>' gruppe[j].achse[i].bezeichnung' | Logical axis number<br>'log_achs_nr' |
|---|---|---|---|
| 1 | 0 | **X** | **3** |
| 2 | 1 | **Y_ACHSE** | **1** |
| 3 | 2 | **Z** | **5** |
| 4 | 3 | **W** | **2** |
| 5 | 0 | **C1** | **7** |
| 6 | 1 | **B** | **6** |

| | | |
|---|---|---|
| **Number of axis groups** | **'grp_anzahl'** | **2** |
| **Axis group designation** | **'gruppe[0].bezeichnung'**<br>**'gruppe[1].bezeichnung'** | **GRUPPE1**<br>**GRUPPE2** |
| **Number of axes in each axis group** | **'gruppe[0].achs_anzahl'**<br>**'gruppe[1].achs_anzahl'** | **4**<br>**2** |

## 1.24.1 Number of axis groups (P-CHAN-00023)

| **P-CHAN-00023** | **Number of axis groups in the NC channel** |
|---|---|
| Description | This parameter defines the maximum number of axis groups. |
| Parameter | grp_anzahl |
| Data type | UNS16 |
| Data range | 0 ... 1 |
| Dimension | ---- |
| Default value | 1 |
| Remarks | Currently, only axis configurations are permitted with one axis group. This parameter must therefore always be assigned with '1'. |

## 1.24.2 Axis group-specific elements

### 1.24.2.1 Axis group structure (gruppe[i].*)

This structure contains the axis group designation, the number of axes and the axis structure.

| Structure name | Index |
|---|---|
| gruppe[i] | i = 0 (only one group permitted, group 0) |

### 1.24.2.2 Axis group designation (P-CHAN-00005)

| P-CHAN-00005 | Name of the axis group in the NC channel |
|---|---|
| Description | This element defines the designation of the axis groups in the channel. This is a string. |
| Parameter | gruppe[i].bezeichnung |
| Data type | STRING |
| Data range | Maximum 16 characters (length of axis group designation, application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameterisation example: The axis group [0] receives the designation GRUPPE_1.<br>*gruppe[0].bezeichnung GRUPPE_1*<br>* Note: The default value of variables is a blank string. |

### 1.24.2.3 Number of axes in each axis group (P-CHAN-00003)

| P-CHAN-00003 | Number of axes in each axis group |
|---|---|
| Description | This parameter defines the number of axes in an axis group. Only defined axes are counted; gaps in the axis configuration are not considered. |
| Parameter | gruppe[i].achs_anzahl |
| Data type | UNS16 |
| Data range | 0 ... 32 (maximum number of axes, application-specific) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Only axis configurations with one axis group (gruppe[0].*) are permitted.<br>Parameterisation example:<br>*gruppe[0].achs_anzahl 3* |

## 1.24.3 Axis structure (gruppe[i].achse[j].*)

This structure contains axis-specific elements.

| Structure name | Index |
|---|---|
| achse[j] | j = 0 ... 31 (maximum number of axes 32, application-specific) |

### 1.24.3.1 Axis designation (P-CHAN-00006)

| P-CHAN-00006 | Name of an axis in the NC channel |
|---|---|
| Description | This variable specifies the axis designations of all axes in the interpolator. The related corresponding logical axis number is defaulted by the parameter P-CHAN-00035 [▶ 140]. |
| Parameter | gruppe[i].achse[j].bezeichnung |

| | |
|---|---|
| Data type | STRING |
| Data range | Maximum 16 characters (length of axis designation, application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | The axis designations must begin with the letters A, B, C, U, V, W, X, Y, Z or Q. After that, all letters and digits are possible. The axis designations must be unique.<br>Caution: The axis identifiers ‘A1‘ to ‘A32‘ may not be used if the parameter P-CHAN-00253 [▶ 262] is active. In this case, these identifiers are explicitly permitted in the axis lists (P-AXIS-00297) only.<br>Parameterisation example:<br>An axis group with 3 axes is to be configured. The first axis is addressed with ‘X_ACHSE1’ in the NC program. The second axis is designated ‘A’. The third axis is addressed with ‘W1’.<br>`gruppe[0].achse[0].bezeichnung X_ACHSE1`<br>`gruppe[0].achse[1].bezeichnung A`<br>`gruppe[0].achse[2].bezeichnung W1+`<br>* Note: The default value of variables is a blank string. |

### 1.24.3.2 Logical axis number (P-CHAN-00035)

| **P-CHAN-00035** | **Logical number of an axis in the NC channel** |
|---|---|
| Description | The logical axis number is unique in the entire system. The logical axis number is used to assign the axis designation in the NC program to the axis data (axis parameter lists, see [AXIS]) supplied by axis management. Only these logical axis numbers are significant if they are known to the axis management.<br>The logical axis number "0" is not permitted. |
| Parameter | gruppe[i].achse[j].log_achs_nr |
| Data type | UNS16 |
| Data range | 0 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | - |
| Remarks | A logical axis number may not be assigned several times. A logical axis number may not be simultaneously configured as path axis and spindle. If this is the case, the plausibility check generates an error message.<br>Axes of the 'spindle' type may not be specified as path axes since they are located in a spindle interpolator on start-up.<br>Parameterisation example:<br>An axis group with 3 axes is to be configured. The first axis is assigned the axis data with the logical axis number '2', the second axis is assigned the data with the logical axis number '7' and the third axis is assigned the data with the logical axis number '1' :<br>`gruppe[0].achse[0].log_achs_nr 2`<br>`gruppe[0].achse[1].log_achs_nr 7`<br>`gruppe[0].achse[2].log_achs_nr 1` |

### 1.24.3.3 Default feed axis (P-CHAN-00011)

| **P-CHAN-00011** | **Assign an axis to the default feed group** |
|---|---|

| | |
|---|---|
| Description | This parameter defines which axes form the default feed group (effect of the F word). If no feed axes are defined, all the main axes automatically form the channel feed group (index 1, 2, 3) after start-up. The effectiveness of this setting is defined by the parameter P-CHAN-00096 [▶ 135] .<br>Feed axes are freely definable with linear interpolation. Parameterisation is restricted with circular interpolation and polynomial interpolation:<br>With circular interpolation either all the main axes must be defined as feed axes or all defined feed axes must be tracked axes.<br>With polynomial interpolation only the main axes may be feed axes. |
| Parameter | gruppe[i].achse[j].default_feed_axis |
| Data type | BOOLEAN |
| Data range | 0: Axis not in feed group<br>1: Axis in feed group |
| Dimension | ---- |
| Default value | 0 |
| Remarks | With multi-channel applications which have gaps in the configuration in the channel-specific main axis groups on start-up (e.g. index 0 -> X, index 1 -> - index 2 -> Z), it is recommended not to assign or comment the channel parameter P-CHAN-00011 in the channel axes. In this case any axis can be changed into a main axis and a complete feed axis group can be created. In other cases all changed axes are treated as tracked axes. |

## 1.25 Configuration of spindles

This section describes the elements for configuring spindles. Note that only the spindles entered in the channel parameter list can also be addressed in the part program. This applies not only to position-controlled spindles but also to speed-controlled (PLC) spindles.

The spindle configuration defined in the channel parameter list is the default setting which is provided after the controller is started up. See also the section 'Spindle programming' in [PROG].

### 1.25.1 Number of spindles (P-CHAN-00082)

| **P-CHAN-00082** | **Number of configured spindles in NC channel** |
|---|---|
| Description | This element specifies the total number of existing position-controlled and speed-controlled spindles. The number of spindles must be identical with the entered spindles (spindel[i].*). |
| Parameter | spdl_anzahl |
| Data type | UNS16 |
| Data range | 0 ≤ spdl_anzahl ≤ 6 (application-specific) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: A position-controlled and a speed-controlled spindle are to be configured.<br>*spdl_anzahl      2* |

### 1.25.2 Spindle name within a path compound (P-CHAN-00010)

| **P-CHAN-00010** | **Spindle name within a path compound** |
|---|---|
| Description | If a spindle axis is changed with specific machining modes (e.g. C axis mode) in the path compound, it can be addressed by this default name (e.g. C1). |
| Parameter | default_ax_name_of_spindle |
| Data type | STRING |
| Data range | Maximum 16 characters (length of spindle designation, application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | The designation of spindles in the path compound must start with the letters A, B, C, U, V, W, X, Y, Z or Q. After that, all letters and digits are possible. The axis designation must be unique. It may not be identical with the configured name of a channel axis (P-CHAN-00006 [▶ 139]).<br>Parameterisation example: The C axis receives the designation C1.<br>*default_ax_name_of_spindle      C1*<br>* Note: The default value of variables is a blank string. |

### 1.25.3 Machine ID with C axis face machining (P-CHAN-00008)

| **P-CHAN-00008** | **Machine ID with C axis face machining** |
|---|---|
| Description | This parameter defines the type of machine used for face machining. |
| Parameter | cax_face_id |
| Data type | UNS16 |
| Data range | 1: Lathe (automatic orientation of the rotary axis in the centre of rotation)<br>2: Milling machines (no orientation) |
| Dimension | ---- |

| | |
|---|---|
| Default value | 0 |
| Remarks | Parameterisation example: Face machining takes place on a milling machine.<br>*cax_face_id     2* |

## 1.25.4 Definition of main spindle

Programming in the default syntax requires the declaration of a spindle as the 'main spindle'. The main spindle can be programmed in the conventional DIN syntax together with specific default functionalities (e.g. tapping, gear changes). Even if there is only one spindle in the system, it must be configured as the main spindle.

### 1.25.4.1 Main spindle axis number (P-CHAN-00051)

| P-CHAN-00051 | Logical axis number of the main spindle in the NC channel |
|---|---|
| Description | This parameter assigns the definition of the main spindle. The logical axis number of one of the spindles which is configured in the channel parameters list is entered here. After the controller starts up, this spindle becomes the main spindle. However, any other spindle in the system can be declared to be the main spindle using a command in the part program (#MAIN SPINDLE [PROG]). |
| Parameter | main_spindle_ax_nr |
| Data type | UNS16 |
| Data range | 1 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: The spindle with logical axis number 6 is the main spindle.<br>*main_spindle_ax_nr     6* |

### 1.25.4.2 Designation of the main spindle (P-CHAN-00053)

| P-CHAN-00053 | Designation of the main spindle in the NC channel |
|---|---|
| Description | Besides the logical axis number, a name must be assigned to the main spindle by which it can be addressed in the subroutine. This parameter therefore assigns an axis name to the main spindle. The axis name is freely selectable, but the first character must be an 'S'. |
| Parameter | main_spindle_name |
| Data type | STRING |
| Data range | Maximum 16 characters (length of spindle designation, application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | The designation of the main spindle cannot be changed in the subroutine. However, if the main spindle is changed using #MAIN SPINDLE (see [PROG]), the new main spindle is assigned.<br>Parameterisation example: The main spindle (logical axis number 6) is programmed in the subroutine with the name 'S'.<br>*main_spindle_ax_nr      6*<br>**main_spindle_name       S**<br>* Note: The default value of variables is a blank string. |

### 1.25.4.3 Enabling gear changes of the main spindle (P-CHAN-00052)

| P-CHAN-00052 | Enable mechanical gear change of main spindle |
|---|---|
| Description | This parameter enables or disables gear changes for the main spindle. |
| Parameter | main_spindle_gear_change |

| | |
|---|---|
| Data type | BOOLEAN |
| Data range | 0: Spindle gear changes disabled for the main spindle<br>1: Spindle gear changes enabled for the main spindle |
| Dimension | ---- |
| Default value | 0 |
| Remarks | The M functions to select the gear speeds of the main spindle M40–45 are activated by the parameter P-CHAN-00052 in the channel parameter list. The M functions M40–45 can be freely used if gear changes are disabled. |

## 1.25.5 Spindle data (spindel[i].*)

This structure consists of the elements: spindle name, logical axis number, synchronisation types and timeout and process times.

| Structure name | Index |
|---|---|
| spindel[i] | i = 0 ... 5 (maximum number of position and speed-controlled spindles: 6, application-specific) |

### 1.25.5.1 Spindle name (P-CHAN-00007)

<table>
<tr><th>P-CHAN-00007</th><th>Name of a spindle in the NC channel</th></tr>
<tr><td>Description</td><td>This parameter defines the default name to address the spindle in the NC program. Please note that as long as a spindle is the main spindle, it can only be programmed using the main spindle name. The spindle name is a string.</td></tr>
<tr><td>Parameter</td><td>spindel[i].bezeichnung</td></tr>
<tr><td>Data type</td><td>STRING</td></tr>
<tr><td>Data range</td><td>Maximum 16 characters (length of spindle designation, application-specific)</td></tr>
<tr><td>Dimension</td><td>----</td></tr>
<tr><td>Default value</td><td>*</td></tr>
<tr><td>Remarks</td><td>The spindle names must start with the letter ‘S’. After that, all letters and digits are possible. Spindle names must be unique.<br>Parameterisation example: Configuration of a 1-channel system with 3 spindles. After start-up, spindle 'S1' with logical axis number 6 is the main spindle. It is addressed by the spindle name 'S'. The spindles with logical axis numbers 11 and 30 are programmed by their default names 'S2' and 'S3'.
<pre>
spdl_anzahl                          3
:
main_spindle_ax_nr                   6-> -> ->-
main_spindle_name                    S ->-       /
#                                        /       /
spindel[0].bezeichnung               S1-<-       /
spindel[0].log_achs_nr               6-< -< -<-
:
spindel[1].bezeichnung               S2
spindel[1].log_achs_nr               11
:
spindel[2].bezeichnung               S3
spindel[2].log_achs_nr               30
* Note: The default value of variables is a blank string.
</pre></td></tr>
</table>

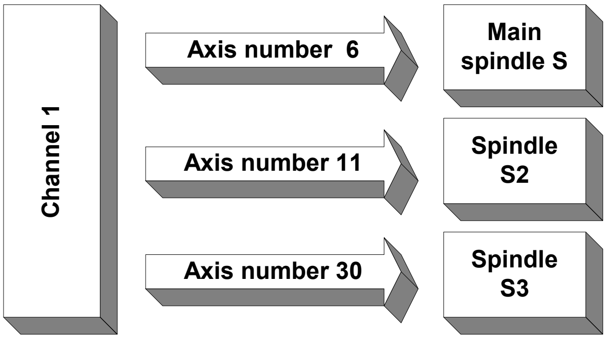


Fig. 31: Example of channel-specific spindle configuration

### 1.25.5.2 Logical axis number (P-CHAN-00036)

| P-CHAN-00036 | Logical axis number of a spindle in NC channel |
|---|---|

| | |
|---|---|
| Description | The logical axis number is entered in this parameter. The logical axis name is unique throughout the entire system. The logical axis number assigns the spindle name in the NC program to the axis data (see axis parameters lists in [AXIS]). Therefore only logical numbers make sense if they are known to the NC program.<br>The logical axis number "0" is not permitted. |
| Parameter | spindel[i].log_achs_nr |
| Data type | UNS16 |
| Data range | 1... MAX(UNS16) |
| Dimension | ---- |
| Default value | - |
| Remarks | A logical axis number may not be assigned several times. A logical axis number may not be simultaneously configured as path axis and spindle. If this is the case, the plausibility check of the channel parameters generates an error message at start-up. |

### 1.25.5.3 Channel-specific PLC spindle (P-CHAN-00069)

| **P-CHAN-00069** | **Spindle control by PLC via channel specific interface** |
|---|---|
| Description | This parameter is set to TRUE if a spindle is controlled directly by the PLC and not by a spindle interpolator in the NC channel. Note here that all synchronisations are no longer (spindle) axis-specific but are output and processed by the channel-specific HLI range. The axis-specific syntax for programming spindle commands is still allowed, but is limited to specifying the speed and the M functions M3/M4/M5/M19. |
| Parameter | spindel[i].plc_control |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

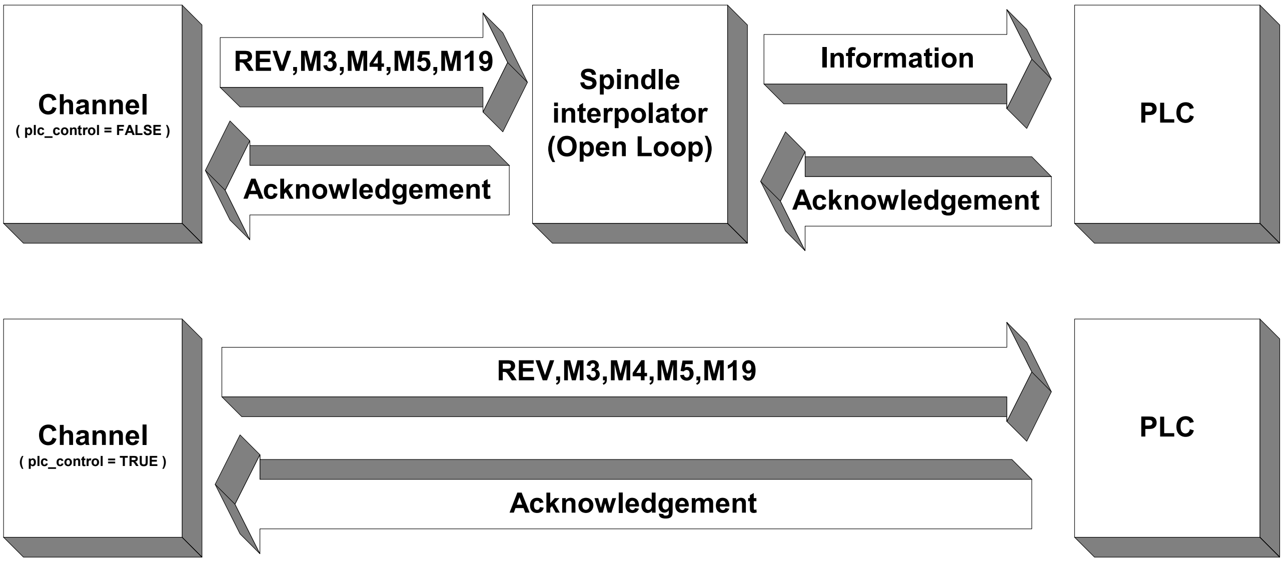


Fig. 32: Difference between spindle control by spindle interpolator or PLC

### 1.25.5.4 Spindle-specific synchronisation types

The synchronisation type of spindle function S and the spindle M functions M3, M4, M5 and M19 are defined spindle-specific. The following synchronisation types are permissible:

| Constant | Value | Meaning |
|---|---|---|
| NO_SYNCH | 0x00000000 | No output of S/M function to PLC |
| MOS | 0x00000001 | Output of S/M function to PLC without synchronisation |
| MVS_SVS | 0x00000002 | Output of S/M function to PLC before motion block,<br>synchronisation before motion block |
| MVS_SNS | 0x00000004 | Output of S/M function to PLC before motion block,<br>synchronisation after motion block |
| MNS_SNS | 0x00000008 | Output of S/M function to PLC after motion block,<br>synchronisation after motion block |
| MNE_SNS | 0x00000020 | Output of S/M function to PLC after (measurement event,<br>synchronisation after motion block<br>(for edge banding option only) |
| MVS_SLM | 0x00004000 | Late synchronisation, output of S/M function to PLC in the block,<br>synchronisation on transition to G01/G02/G03 (implicit synchronisation) |
| MVS_SLP | 0x00008000 | Late synchronisation, output of S/M function to PLC in the block,<br>synchronisation by NC command #EXPL SYN (explicit synchronisation) |
| PLC_INFO | 0x00020000 | It only makes sense to use the PLC_INFO bit with NC spindles (position-controlled spindles). In addition to the synchronisation type, the PLC_INFO bit can be set for every spindle M function. It defines whether the spindle M function must also be output to the PLC and acknowledged by the PLC.<br>If the PLC_INFO bit is not set, no output to PLC is sent to the PLC and internal synchronisation only takes place based on window monitoring for the position or speed.<br>For PLC spindles (speed-controlled spindles) please note the following:<br>In general, with every spindle M function, an output of the M function is also sent to the PLC automatically. It is therefore not necessary to additionally set the PLC_INFO bit. |
| BWD_SYNCH | 0x00400000 | Synchronisation of S/M function during backward movement with MVS_SVS |
| FWD_SYNCH | 0x00800000 | Synchronisation of S/M function during 'Simulated forward movement' with the corresponding synchronisation type |
| FAW_SYNCH | 0x10000000 | Decoding stop (Flush and Wait): Output of S/M function to PLC and stop of program decoding at block end until program run is completed.<br>FAW_SYNCH can be set in addition to other synchronisation types. S/M functions with FAW_SYNCH may not be used with active tool radius compensation (TRC), polynomial contouring or HSC mode. |
| NOT_VALID | 0x80000000 | Blocking the S/M function. Programming one of the S/M functions using this synchronisation generates an error message. This synchronisation may only be set exclusively. |

**For examples, see Section** Synchronisation type of M functions [▸ 65]**.**

The synchronisation type of the S function has no effect if a spindle M function is programmed in the NC block. Synchronisation only takes place based on the settings for the spindle M function. The following priorities apply: **M19 > M3/M4/M5 > S**

#### 1.25.5.4.1 Synchronisation type of the spindle S function (P-CHAN-00081)

| P-CHAN-00081 | Synchronisation type of the spindle S function |
|---|---|

| | |
|---|---|
| Description | This parameter defines the synchronisation type of the spindle S function. The synchronisation type is defined as a string constant or a hexadecimal value. |
| Parameter | spindel[i].s_synch |
| Data type | STRING |
| Data range | See Spindle-specific synchronisation types [▶ 147] |
| Dimension | ---- |
| Default value | NO_SYNCH |
| Remarks | Parameterisation example: For a (position-controlled) spindle 'S1' the spindle function is executed without synchronisation. The PLC is not informed.<br>`spindel[0].bezeichnung       S1`<br>`spindle[0].log_achs_no        6`<br>**`spindel[0].s_synch      MOS                     0x00000001`**<br>`spindle[0].m3_synch      PLC_INFO \| MVS_SVS     0x00020002`<br>`spindle[0].m4_synch      PLC_INFO \| MVS_SNS     0x00020004`<br>`spindle[0].m5_synch      PLC_INFO \| MVS_SVS     0x00020002`<br>`spindle[0].m19_synch     MNS_SNS                 0x00000008`<br><br>`Note: Programming a UNS32 variable is permissible for downward compatibility reasons.`<br>`Example: spindle[0].s_synch 0x00000001` |

### 1.25.5.4.2 Synchronisation type for M03 (P-CHAN-00045)

| **P-CHAN-00045** | **Synchronisation type for M03** |
|---|---|
| Description | When the M03 function is used, the synchronisation type must be defined for the spindles used. The synchronisation type is defined as a string constant or a hexadecimal value. |
| Parameter | spindel[i].m3_synch |
| Data type | STRING |
| Data range | See Spindle-specific synchronisation types [▶ 147] |
| Dimension | ---- |
| Default value | NO_SYNCH |
| Remarks | Parameterisation example: For a (position-controlled) spindle 'S1', the spindle-specific M function M03 is assigned the synchronisation type MVS_SVS. The PLC is also informed.<br>`spindel[0].bezeichnung       S1`<br>`spindel[0].log_achs_no        6`<br>`spindel[0].s_synch       MOS                     0x00000001`<br>**`spindel[0].m3_synch      PLC_INFO \| MVS_SVS     0x00020002`**<br>`spindle[0].m4_synch      PLC_INFO \| MVS_SNS     0x00020004`<br>`spindle[0].m5_synch      PLC_INFO \| MVS_SVS     0x00020002`<br>`spindle[0].m19_synch     MNS_SNS                 0x00000008`<br><br>`Note: Programming a UNS32 variable is permissible for downward compatibility reasons.`<br>`Example: spindel[0].m3_synch   0x00020002` |

### 1.25.5.4.3 Synchronisation type for M04 (P-CHAN-00047)

| **P-CHAN-00047** | **Synchronisation type for M04** |
|---|---|
| Description | When the M04 function is used, the synchronisation type must be defined for the spindles used. The synchronisation type is defined as a string constant or a hexadecimal value. |
| Parameter | spindel[i].m4_synch |
| Data type | STRING |
| Data range | See Spindle-specific synchronisation types [▶ 147] |
| Dimension | ---- |
| Default value | NO_SYNCH |

| | |
|---|---|
| Remarks | Parameterisation example: For a (position-controlled) spindle 'S1' the spindle-specific M function M04 is assigned the synchronisation type MVS_SNS. The PLC is also informed. |

```
spindel[0].bezeichnung       S1
spindel[0].log_achs_no        6
spindel[0].s_synch      MOS                    0x00000001
spindel[0].m3_synch     PLC_INFO | MVS_SVS     0x00020002
spindel[0].m4_synch     PLC_INFO | MVS_SNS     0x00020004
spindel[0].m5_synch     PLC_INFO | MVS_SVS     0x00020002
spindel[0].m19_synch    MNS_SNS                0x00000008

Note: Programming a UNS32 variable is permissible for downward compatibility
reasons.
Example: spindel[0].m4_synch   0x00020004
```

### 1.25.5.4.4 Synchronisation type for M05 (P-CHAN-00049)

| P-CHAN-00049 | Synchronisation type for M05 |
|---|---|
| Description | When the M05 function is used, the synchronisation type must be defined for the spindles used. The synchronisation type is defined as a string constant or a hexadecimal value. |
| Parameter | spindel[i].m5_synch |
| Data type | STRING |
| Data range | See Spindle-specific synchronisation types [▶ 147] |
| Dimension | ---- |
| Default value | NO_SYNCH |
| Remarks | Parameterisation example: For a (position-controlled) spindle 'S1', the spindle-specific M function M05 is assigned the synchronisation type MVS_SVS. The PLC is also informed. |

```
spindel[0].bezeichnung       S1
spindel[0].log_achs_no        6
spindel[0].s_synch      MOS                    0x00000001
spindel[0].m3_synch     PLC_INFO | MVS_SVS     0x00020002
spindel[0].m4_synch     PLC_INFO | MVS_SNS     0x00020004
spindel[0].m5_synch     PLC_INFO | MVS_SVS     0x00020002
spindel[0].m19_synch    MNS_SNS                0x00000008

Note: Programming a UNS32 variable is permissible for downward compatibility
reasons.

Example: spindel[0].m5_synch   0x00020002
```

### 1.25.5.4.5 Synchronisation type for M19 (P-CHAN-00043)

| P-CHAN-00043 | Synchronisation type for M19 |
|---|---|
| Description | When the M19 function is used, the synchronisation type must be defined for the spindles used. The synchronisation type is defined as a string constant or a hexadecimal value. |
| Parameter | spindel[i].m19_synch |
| Data type | STRING |
| Data range | See Spindle-specific synchronisation types [▶ 147] |
| Dimension | ---- |
| Default value | NO_SYNCH |

| | |
|---|---|
| Remarks | Parameterisation example: For a (position-controlled) spindle 'S1' the spindle-specific M function M19 is assigned the synchronisation type MNS_SNS. The PLC is also informed. |

```
spindel[0].bezeichnung       S1
spindle[0].log_achs_no        6
spindel[0].s_synch      MOS                          0x00000001
spindel[0].m3_synch     PLC_INFO | MVS_SVS    0x00020002
spindel[0].m4_synch     PLC_INFO | MVS_SNS    0x00020004
spindel[0].m5_synch     PLC_INFO | MVS_SVS    0x00020002
spindel[0].m19_synch    MNS_SNS               0x00000008

Note: Programming a UNS32 variable is permissible for downward compatibility
reasons.
Example: spindel[0].m19_synch   0x00000008
```

### 1.25.5.5 Timeout and process time of the spindle function S (P-CHAN-00080)

| **P-CHAN-00080** | **Timeout and process time of the spindle function S for the machining time calculation** |
|---|---|
| Description | This parameter specifies the timeout and the process time of a spindle S function when the machining time calculation is activated. |
| Parameter | spindel[i].s_prozess_zeit |
| Data type | UNS32 |
| Data range | 0 < s_prozess_zeit < MAX(UNS32) |
| Dimension | µs |
| Default value | 0 |
| Remarks | Currently, the parameter is only used to calculate machining time.<br>Parameterisation example: The process time of the spindle function is defined as 1 s for an ‘S1’ spindle.<br>`spindel[0].bezeichnung     S1`<br>`spindel[0].log_achs_nr      6`<br>`:`<br>**`spindel[0].s_prozess_zeit      1000000`**<br>`spindel[0].m3_prozess_zeit     1000000`<br>`spindel[0].m4_prozess_zeit     1000000`<br>`spindel[0].m5_prozess_zeit     1500000`<br>`spindel[0].m19_prozess_zeit    2000000` |

### 1.25.5.6 Timeout and process time of M03 (P-CHAN-00044)

| **P-CHAN-00044** | **Timeout and process time of M03 for calculation of machining time** |
|---|---|
| Description | This parameter specifies the timeout and the process time of the M function M03 when the machining time calculation is activated. |
| Parameter | spindel[i].m3_prozess_zeit |
| Data type | UNS32 |
| Data range | 0 < m3_prozess_zeit < MAX(UNS32) |
| Dimension | µs |
| Default value | 0 |
| Remarks | Currently, the parameter is only used to calculate machining time.<br>Parameterisation example: The process time for M03 is set to 1s for the spindle 'S1'.<br>`spindel[0].bezeichnung     S1`<br>`spindel[0].log_achs_nr      6`<br>`:`<br>`spindel[0].s_prozess_zeit      1000000`<br>**`spindel[0].m3_prozess_zeit     1000000`**<br>`spindel[0].m4_prozess_zeit     1000000`<br>`spindel[0].m5_prozess_zeit     1500000`<br>`spindel[0].m19_prozess_zeit    2000000` |

### 1.25.5.7 Timeout and process time of M04 (P-CHAN-00046)

| **P-CHAN-00046** | **Timeout and process time of M04 to calculate machining time** |
|---|---|
| Description | This parameter specifies the timeout and the process time of the M function M04 when the machining time calculation is activated. |
| Parameter | spindel[i].m4_prozess_zeit |
| Data type | UNS32 |
| Data range | 0 < m4_prozess_zeit < MAX(UNS32) |
| Dimension | µs |
| Default value | 0 |

| | |
|---|---|
| Remarks | Currently, the parameter is only used to calculate machining time.<br>Parameterisation example: The process time for M04 is set to 1s for the spindle 'S1'.<br>`spindel[0].bezeichnung     S1`<br>`spindel[0].log_achs_nr      6`<br>`:`<br>`spindel[0].s_prozess_zeit      1000000`<br>`spindel[0].m3_prozess_zeit     1000000`<br>**`spindel[0].m4_prozess_zeit     1000000`**<br>`spindel[0].m5_prozess_zeit     1500000`<br>`spindel[0].m19_prozess_zeit    2000000` |

### 1.25.5.8 Timeout and process time of M05 (P-CHAN-00048)

| P-CHAN-00048 | Timeout and process time of M05 to calculate machining time |
|---|---|
| Description | This parameter specifies the timeout and the process time of the M function M05 when the machining time calculation is activated. |
| Parameter | spindel[i].m5_prozess_zeit |
| Data type | UNS32 |
| Data range | 0 < m5_prozess_zeit < MAX(UNS32) |
| Dimension | µs |
| Default value | 0 |
| Remarks | Currently, the parameter is only used to calculate machining time.<br>Parameterisation example: For spindle 'S1' the process time for M05 is defined as 1 s.<br>`spindel[0].bezeichnung     S1`<br>`spindel[0].log_achs_nr      6`<br>`:`<br>`spindel[0].s_prozess_zeit      1000000`<br>`spindel[0].m3_prozess_zeit     1000000`<br>`spindel[0].m4_prozess_zeit     1000000`<br>**`spindel[0].m5_prozess_zeit     1500000`**<br>`spindel[0].m19_prozess_zeit    2000000` |

### 1.25.5.9 Timeout and process time of M19 (P-CHAN-00042)

| P-CHAN-00042 | Timeout and process time of M19 to calculate machining time |
|---|---|
| Description | This parameter specifies the timeout and the process time of the M function M19 when the machining time calculation is activated. |
| Parameter | spindel[i].m19_prozess_zeit |
| Data type | UNS32 |
| Data range | 0 < m19_prozess_zeit < MAX(UNS32) |
| Dimension | µs |
| Default value | 0 |
| Remarks | Currently, the parameter is only used to calculate machining time.<br>Parameterisation example: The process time for M19 is set to 2s for the spindle 'S1'.<br>`spindel[0].bezeichnung     S1`<br>`spindel[0].log_achs_nr      6`<br>`:`<br>`spindel[0].s_prozess_zeit      1000000`<br>`spindel[0].m3_prozess_zeit     1000000`<br>`spindel[0].m4_prozess_zeit     1000000`<br>`spindel[0].m5_prozess_zeit     1500000`<br>**`spindel[0].m19_prozess_zeit    2000000`** |

### 1.25.5.10 Spindle gear change

The following switches and variables are needed to set the spindle speed ranges and gear changes.

Gear changing is only possible for the main spindle.

#### 1.25.5.10.1 Direction of range selection (P-CHAN-00074)

| **P-CHAN-00074** | **Direction of range selection for spindle gear change** |
|---|---|
| Description | This parameter defines whether the lower or the higher gear range is selected when speed ranges overlap. If 'range_way' = 0 the search starts from the lowest speed range, if 'range_way' > 0 from the highest. The correct speed (gear) range is the one where the programmed spindle speed is first found. |
| Parameter | spindel[i].range_way |
| Data type | UNS16 |
| Data range | 0 ≤ range_way < MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: The search starts from the lowest speed range.<br>*spindel[0].range_way  0  (from bottom to top)* |

#### 1.25.5.10.2 Automatic range selection (P-CHAN-00004)

| **P-CHAN-00004** | **Automatic range selection for spindle gear change** |
|---|---|
| Description | If the spindle gear range is to be be automatically determined by the NC kernel, this parameter must be set to 1. In this case the M functions M40 to M45 need not be programmed. This means that the correct gear range is determined implicitly by the programmed speed (S word). |
| Parameter | spindel[i].autom_range |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: The automatic range selection is enabled.<br>*spindel[0].autom_range 1* |

### 1.25.5.10.3 Table of speed ranges (spindel[i].range_table[j].*)

The system allows the definition of maximum 6 speed ranges (M40 to M45) per spindle. For each range, a minimum and maximum speed is defined (unit = rpm).

#### Min. spindle speed (P-CHAN-00058)

| P-CHAN-00058 | Minimum spindle speed of a speed range (spindle gear change) |
|---|---|
| Description | The speed ranges of a spindle may be defined with or without overlap. If one range is not used, the corresponding values must be set to zero in the table. |
| Parameter | spindel[i].range_table[j].min_speed |
| Data type | UNS16 |
| Data range | 0 ≤ min_speed ≤ MAX(UNS16) |
| Dimension | rpm |
| Default value | 0 |
| Remarks | Parameterisation example: Definition of a speed range table for 6 ranges. Only the first four ranges are used.<br>`spindel[0].range_table[0].min_speed 50`<br>`spindel[0].range_table[0].max_speed 560`<br>`spindel[0].range_table[1].min_speed 400`<br>`spindel[0].range_table[1].max_speed 800`<br>`spindel[0].range_table[2].min_speed 700`<br>`spindel[0].range_table[2].max_speed 3360`<br>`spindel[0].range_table[3].min_speed 3361`<br>`spindel[0].range_table[3].max_speed 4000`<br>`spindel[0].range_table[4].min_speed 0`<br>`spindel[0].range_table[4].max_speed 0`<br>`spindel[0].range_table[5].min_speed 0`<br>`spindel[0].range_table[5].max_speed 0` |

#### Max. spindle speed (P-CHAN-00055)

| P-CHAN-00055 | Maximum spindle speed of a speed range (spindle gear change) |
|---|---|
| Description | The speed ranges of a spindle may be defined with or without overlap. If one range is not used, the corresponding values must be set to zero in the table. |
| Parameter | spindel[i].range_table[j].max_speed |
| Data type | UNS16 |
| Data range | 0 ≤ max_speed ≤ MAX(UNS16) |
| Dimension | rpm |
| Default value | 0 |
| Remarks | Parameterisation example: Definition of a speed range table for 6 ranges. Only the first four ranges are used.<br>`spindel[0].range_table[0].min_speed 50`<br>`spindel[0].range_table[0].max_speed 560`<br>`spindel[0].range_table[1].min_speed 400`<br>`spindel[0].range_table[1].max_speed 800`<br>`spindel[0].range_table[2].min_speed 700`<br>`spindel[0].range_table[2].max_speed 3360`<br>`spindel[0].range_table[3].min_speed 3361`<br>`spindel[0].range_table[3].max_speed 4000`<br>`spindel[0].range_table[4].min_speed 0`<br>`spindel[0].range_table[4].max_speed 0`<br>`spindel[0].range_table[5].min_speed 0`<br>`spindel[0].range_table[5].max_speed 0` |

### 1.25.5.11 Identification of an optional spindle (P-CHAN-00415)

| P-CHAN-00415 | Identify an optional spindle |
|---|---|

| | |
|---|---|
| Description | If this parameter is assigned the value 1, a check is made at start-up whether this spindle is known in the system.<br>If the check is positive, it is available in the NC channel after start-up.<br>If the check is negative, this spindle is removed form the configuration data of the NC channel. It is not available after start-up and no interface is created to this spindle. |
| Parameter | spindel[i].optional |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | If the main spindle is marked as optional and the check is negative, no main spindle is available in the NC channel after start-up. This is indicated by a message. |

## 1.26 Settings for synchronous mode (synchro_data.*)

The figure below shows the basic structure of a machine with two machining units with Y, Z and C axes and a mechanically coupled X axis. If the axis group (X, Y2, Z2, C2) is to be coupled to the axis group (X, Y1, Z1, C1), please specify the coupling procedures described in the following sections.

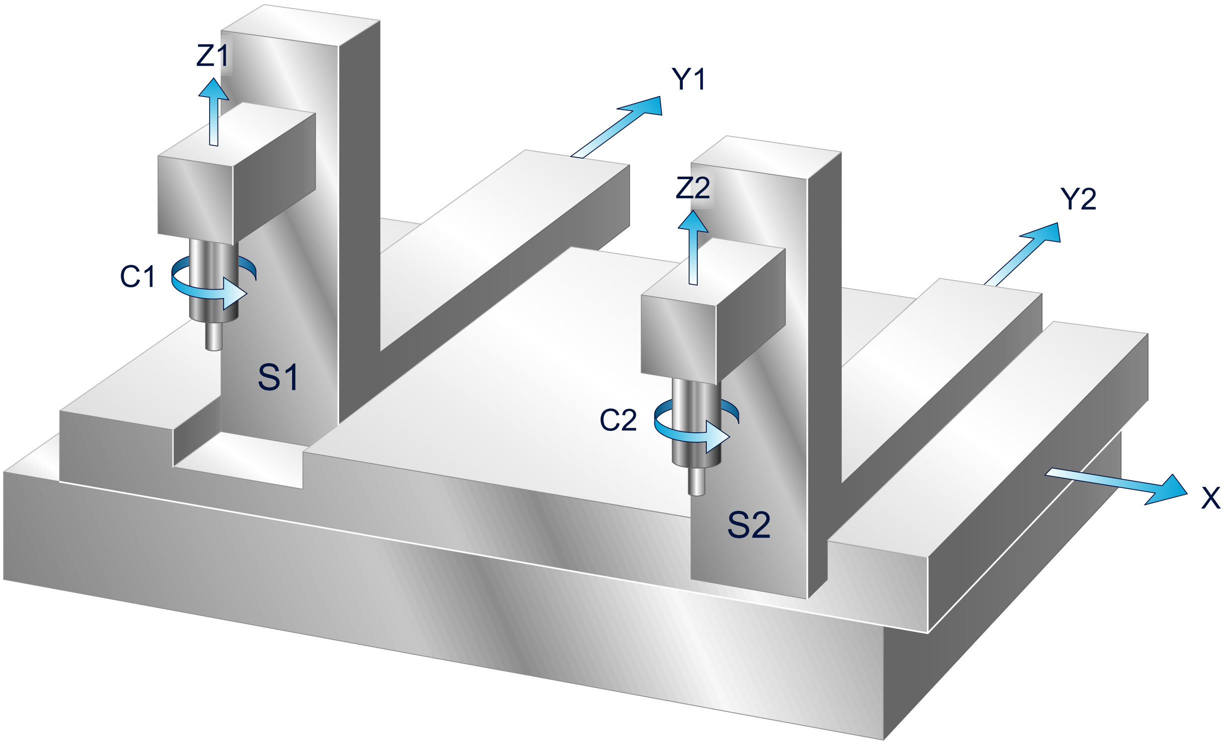


Fig. 33: Example of a machine structure for synchronous operation with spindles

The default coupling specifications can be defined for every axis group. It is listed in the NC working data under the coupling number '0'.

**NOTICE**

The synchronous operation of axes is only possible for axes in an axis group.

Synchronous mode is described in detail in [PROG].

## 1.26.1 Default coupling specifications (synchro_data.koppel_gruppe[i].*)

The structure element koppel_gruppe[j] specifies the coupling group in which the axes are coupled in pairs.

| Structure name | Index |
|---|---|
| koppel_gruppe[i] | i = 0 (only one coupling group allowed, coupling group 0) |

### 1.26.1.1 Coupled axis pairs (synchro_data.koppel_gruppe[i].paar[j].*)

The structure element 'paar[j]' specifies the coupled axis pair of the coupling specification. In a coupled axis pair, one axis is defined as the master axis and the other as the slave axis. The structure consists of the elements: logical axis number of the slave axis and the logical axis number of the master axis which are described in the following chapters. The coupled axis pairs may consist of path axes or of spindle axes. A combination of path and spindle axes within one coupled axis pair is not permitted.

**NOTICE**

The synchronous operation of path axes is only possible for axes within an axis group or only for spindle axes of known spindles within the channel.

| Structure name | Index |
|---|---|
| paar[j] | j = 0 ... 14 (maximum number of coupled axis pairs: 15, application-specific) |

#### 1.26.1.1.1 Logical axis number of the slave axis (P-CHAN-00038)

| P-CHAN-00038 | Logical axis number of the slave axis (synchronous operation) |
|---|---|
| Description | This parameter defines an axis as the slave axis. The logical axis number P-CHAN-00035 [▶ 140] must then be specified. |
| Parameter | synchro_data.koppel_gruppe[i].paar[j].log_achs_nr_slave |
| Data type | UNS16 |
| Data range | 1 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: The axis with the logical number 3 is defined in the coupled axis pair[1] within the coupled axis group[0] as slave axis.<br>*synchro_data.koppel_gruppe[0].paar[1].log_achs_nr_slave 3* |

#### 1.26.1.1.2 Logical axis number of the master axis (P-CHAN-00037)

| P-CHAN-00037 | Logical axis number of the master axis (synchronous operation) |
|---|---|
| Description | This parameter defines an axis as the master axis. The logical axis number P-CHAN-00035 [▶ 140] must then be specified. |
| Parameter | synchro_data.koppel_gruppe[i].paar[j].log_achs_nr_master |
| Data type | UNS16 |
| Data range | 1 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: The axis with the logical number 1 is defined in the coupled axis pair[1] within the coupled axis group[0] as the master axis.<br>*synchro_data.koppel_gruppe[0].paar[1].log_achs_nr_master 1* |

### 1.26.1.1.3 Mode of the coupled axis pair (P-CHAN-00061)

| P-CHAN-00061 | Mode of the coupled axis pair (synchronous operation) |
|---|---|
| Description | Since spindle axes can be coupled in addition to the synchronous operation of path axes, the type of coupling must be specified for each pair. The coupling type must therefore be entered in this parameter. |
| Parameter | synchro_data.koppel_gruppe[i].paar[j].mode |
| Data type | UNS16 |
| Data range | 0: Coupled axis pair consists of path axes<br>1: Coupled axis pair consists of spindle axes |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>The coupled axis pair [0] within the coupled axis group [0] consists of path axes (mode 0)<br>The coupled axis pair [1] within the coupled axis group [0] consists of spindle axes (mode 1).<br>`#synchro_data.koppel_gruppe[0].paar[0].log_achs_nr_slave 4`<br>`#synchro_data.koppel_gruppe[0].paar[0].log_achs_nr_master 1`<br>`#synchro_data.koppel_gruppe[0].paar[0].mode 0`<br>`#synchro_data.koppel_gruppe[0].paar[1].log_achs_nr_slave 11`<br>`#synchro_data.koppel_gruppe[0].paar[1].log_achs_nr_master 6`<br>`#synchro_data.koppel_gruppe[0].paar[1].mode 1` |

### 1.26.1.1.4 Restore coupling after RESET (P-CHAN-00104)

| P-CHAN-00104 | Restore coupling after reset (synchronous operation) |
|---|---|
| Description | An active axis coupling (synchronous operation) is deselected by NC reset. Set this parameter to 1 if the axis coupling is to be reactivated automatically in the following NC program. |
| Parameter | synchro_data.restore_coupling_after_reset |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter may only be used if all axes participating in the coupling exist in the basic configuration of the NC channel. |

### 1.26.1.1.5 Restore coupling after program end (P-CHAN-00105)

| P-CHAN-00105 | Restore coupling after program end (synchronous operation) |
|---|---|
| Description | An active axis coupling (synchronous operation) is deselected at NC program end (#DISABLE AXLINK) if it is not ended explicitly in the NC program. Set this parameter to 1 if the axis coupling is to be reactivated automatically in the following NC program. |
| Parameter | synchro_data.preserve_coupling_after_prog_end |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter may only be used if all axes participating in the coupling exist in the basic configuration of the NC channel. |

## 1.27 Parameterisation of the change logging function (aep.*)

The change logging function outputs the changes (e.g. of G functions) made to the previous status within an NC program.

### 1.27.1 Logging mode (P-CHAN-00024)

| **P-CHAN-00024** | **Enable logging of changes** |
|---|---|
| Description | The logging mode defines whether display data is provided. |
| Parameter | aep.output_mode |
| Data type | UNS08 |
| Data range | 0: Without logging<br>1: Logging is active. The display data can be evaluated via ADS objects. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: The output data is displayed:<br>*aep.output_mode 1* |

### 1.27.2 Logging P parameters (P-CHAN-00068)

| **P-CHAN-00068** | **Enable logging of P parameters** |
|---|---|
| Description | This parameter permits the logging of P parameters. |
| Parameter | aep.p_parameter |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: P parameters are logged.<br>*aep.p_parameter 1* |

### 1.27.3 Logging self-defined variables (P-CHAN-00088)

| **P-CHAN-00088** | **Logging self-defined variables** |
|---|---|
| Description | This parameter permits the logging of self-defined variables. |
| Parameter | aep.v_eigendef |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: Self-defined variables are logged.<br>*aep.v_eigendef 1* |

### 1.27.4 Logging G functions (P-CHAN-00022)

G functions are documented in the Programming Manual [PROG].

| P-CHAN-00022 | Enable logging of G functions |
|---|---|
| Description | G functions are organised into different groups. The functions that mutually exclude themselves are contained in the same group. If a group g_gruppe[i] is set to 1, the active G function of the group is logged. |
| Parameter | aep.g_gruppe[i] where i = 0 38 (maximum number of G function groups: 39, application-specific) |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: The path conditions (g_gruppe[0]) are to be logged, but not tool feed rates adjustment (g_gruppe[3]).<br>*aep.g_gruppe[0] 1*<br>*aep.g_gruppe[3] 0* |

| Group | Included G functions |
|---|---|
| g_gruppe[0] | G00, G01, G02, G03, G04, G33, G63, G74, G98, G99, G160, G331, G332, G301, G302 |
| g_gruppe[1] | G08, G193 |
| g_gruppe[2] | G09, G900, G901 |
| g_gruppe[3] | G10, G11 |
| g_gruppe[4] | G17, G18, G19 |
| g_gruppe[5] | G20, G21, G22, G23, G351 |
| g_gruppe[6] | G25, G26 |
| g_gruppe[7] | G40, G41, G42 |
| g_gruppe[8] | G51, G52 |
| g_gruppe[9] | G53,..., G59, G159 |
| g_gruppe[10] | G60, G359, G360, G61, G260, G261 |
| g_gruppe[11] | G166 |
| g_gruppe[12] | G70, G71 |
| g_gruppe[13] | G80 - G89, G800 - G819 |
| g_gruppe[14] | G90, G91 |
| g_gruppe[15] | G92 |
| g_gruppe[16] | G93, G94, G95, G194 |
| g_gruppe[17] | G96, G97, G196 |
| g_gruppe[18] | G112 |
| g_gruppe[19] | G115, G116, G117 |
| g_gruppe[20] | G130, G131, G231, G333, G334 |
| g_gruppe[21] | G135, G136, G137 |
| g_gruppe[22] | G05, G138, G139, G237, G238, G239 (G236) |
| g_gruppe[23] | G161, G162 |
| g_gruppe[24] | G163 |
| g_gruppe[25] | G164, G165 |
| g_gruppe[26] | G200, G201, G202 |
| g_gruppe[27] | not assigned |
| g_gruppe[28] | G132, G133, G134, G233, G338, G339 |
| g_gruppe[29] | G150, G151 |
| g_gruppe[30] | G100, G101, G102, G106, G107, G108 |
| g_gruppe[31] | G12, G13 |
| g_gruppe[32] | G140, G141 |
| g_gruppe[33] | not assigned |

| Group | Included G functions |
|---|---|
| g_gruppe[34] | G310 |
| g_gruppe[35] | G167 |
| g_gruppe[36] | not available (G335, G337) |
| g_gruppe[37] | G129 |
| g_gruppe[38] | G66 |

Group organisation of G functions

### 1.27.5 Logging tool data (P-CHAN-00093)

| P-CHAN-00093 | Enable the logging of tool data |
|---|---|
| Description | This parameter permits the logging of tool data. |
| Parameter | aep.wz_daten |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: Tool data is saved to a log.<br>*aep.wz_daten 1* |

## 1.28 Definition of macros (makro_def[i].*)

A macro definition consists of a macro name (alias) and a macro content in which one or more complete NC commands are stored. If a macro name is recognised in the NC program, the NC command sequence stored in the associated macro content is executed. Macros are therefore placeholders for NC syntax.

| Structure name | Index |
|---|---|
| makro_def[i] | i = 0 ... 49 (maximum number of macros in channel list: 50, application-specific) |

The use of macros is described in greater detail in [PROG].

### 1.28.1 Specification of macro name (P-CHAN-00085)

| P-CHAN-00085 | Specify a macro name |
|---|---|
| Description | This parameter specifies the macro name. |
| Parameter | makro_def[i].symbol |
| Data type | STRING |
| Data range | Maximum of 30 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string. |

### 1.28.2 Specification of macro content (P-CHAN-00062)

| P-CHAN-00062 | Specify macro content |
|---|---|
| Description | This parameter specifies the macro content (NC syntax). |
| Parameter | makro_def[i].nc_code |

| | |
|---|---|
| Data type | STRING |
| Data range | Maximum of 80 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | Example 1: The parameter calculation is replaced by the macro CALC_P5.<br>*makro_def[4].symbol CALC_P5*<br>*makro_def[4].nc_code P5 = P2 + 3*<br><br>Example 2: The programming of a motion block is replaced by the macro RELPOS1.<br>*makro_def[6].symbol RELPOS1*<br>*makro_def[6].nc_code G01 G91 X10 Y20 F100*<br>* Note: The default value of variables is a blank string. |

# 1.29 Settings for tool radius compensation (TRC)

## 1.29.1 Configuration with TRC (P-CHAN-00092)

| **P-CHAN-00092** | **NC channel configuration includes a module for tool radius compensation (TRC)** |
|---|---|
| Description | If tool radius compensation (TRC) is present in the system configuration, this parameter must be set to 1. |
| Parameter | wrk_im_kanal_vorhanden |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: Tool radius compensation is present in the system configuration. This parameter must therefore be set to 1.<br>*wrk_im_kanal_vorhanden 1* |

## 1.29.2 Suppress warnings during contour masking (P-CHAN-00021)

| **P-CHAN-00021** | **Suppress warnings with contour masking (TRC)** |
|---|---|
| Description | The machine operator is informed by a warning if a contour violation is detected during active contour masking. The warning output can be set by this parameter. |
| Parameter | create_cont_mask_warnings |
| Data type | BOOLEAN |
| Data range | 0: No output of a warning if a contour violation is detected (default).<br>1: Output of a warning if a contour violation is detected. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | It only makes sense to use this parameter if P-CHAN-00092 [▶ 161] (wrk_im_kanal_vorhanden) is assigned TRUE. |

## 1.29.3 Implicit contour masking (P-CHAN-00219)

| **P-CHAN-00219** | **Activate implicit contour masking (TRC)** |
|---|---|

| | |
|---|---|
| Description | It only makes sense to use this parameter if no G141 (contour masking on) is set in the NC program. If the parameter is assigned the value 1, the contour masking algorithms are used to detect contour violations in good time in order to output and stop an error message and so to avoid contour damage. |
| Parameter | implicit_contour_masking |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | It only makes sense to use this parameter if P-CHAN-00092 [▶ 161] (wrk_im_kanal_vorhanden) is assigned TRUE. |

## 1.29.4 Extended kerf detection for circular blocks (P-CHAN-00284)

| **P-CHAN-00284** | **Extended kerf detection for circular blocks (TRC)** |
|---|---|
| Description | This parameter can extend kerf detection for circular blocks. All combinations of linear and circular blocks are then identified (LIN-LIN, LIN-CIR, CIR-CIR, CIR-LIN) |
| Parameter | trc_circular_kerf_masking |
| Data type | BOOLEAN |
| Data range | 0: No extended kerf detection for circular blocks (default). G141/G140 are used to activate/deactivate the #TRC option KERF_MASKING implicitly.<br>1: Extended kerf detection for circular blocks is active. The implicit linking for G141/G140 is skipped. |
| Dimension | ---- |
| Default value | 1 * |
| Remarks | It only makes sense to use this parameter if P-CHAN-00092 [▶ 161] (wrk_im_kanal_vorhanden) is assigned TRUE.<br>* as of CNC Build V3.01.3038 the default value is 1; in earlier build numbers the default value is 0. |

# 1.30 Configuration of syntax check (syn_chk.*)

The syntax check function performs a check of an NC program by the NC and continues to run if an error is detected, if possible. Only actually processed program lines are checked. The NC channel continues to be supplied but no axes or drives are moved. This has the advantage that an NC program can be checked for tool radius compensation or for software limit switch monitoring and this allows a more comprehensive check.

## 1.30.1 Number of errors per row (P-CHAN-00020)

| **P-CHAN-00020** | **Number of errors per row on syntax check** |
|---|---|
| Description | This parameter defines the number of errors after which the program skips to the next program row. |
| Parameter | syn_chk.errors_per_block |
| Data type | UNS16 |
| Data range | 0 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | *syn_chk.fehler_pro_zeile (old syntax up to V2.11.2012.07)*<br>Parameterisation example: After a maximum of 2 errors the program skips to the next NC row.<br>*syn_chk.errors_per_block 2* |

## 1.30.2 Number of errors in an NC program (P-CHAN-00019)

| **P-CHAN-00019** | **Number of errors in an NC program on syntax check** |
|---|---|
| Description | This parameter defines the number of errors after which NC program decoding is aborted. |
| Parameter | syn_chk.errors_total |
| Data type | UNS16 |
| Data range | 0 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | *syn_chk.fehler_gesamt (old syntax up to V2.11.2012.07)*<br>Parameterisation example: After a maximum of 20 errors in the NC program the syntax check is aborted.<br>*syn_chk.errors_total 20* |

## 1.30.3 Specifying the operation mode (P-CHAN-00028)

| **P-CHAN-00028** | **Specify the operation mode on syntax check** |
|---|---|
| Description | The parameter defines the operation mode of the syntax check.<br>Mode 1 - Automatic operation (0):<br>After an error, decoding continues automatically. Syntax check is only aborted when the corresponding limits described in P-CHAN-00019 [▶ 163] and P-CHAN-00020 [▶ 163] are reached.<br>Mode 2 - Step (interactive) mode (1):<br>After each error, decoding is stopped. The operator decides whether the syntax check should continue or decoding should be aborted. When the corresponding limits described in P-CHAN-00019 [▶ 163] and P-CHAN-00020 [▶ 163] are reached, decoding is also aborted automatically. |
| Parameter | syn_chk.interactive |

| | |
|---|---|
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | *interactive (old syntax up to V2.11.2012.07)*<br>Parameterisation example: The syntax check runs in automatic mode.<br>*syn_chk.interactive 0* |

## 1.30.4 Write results of syntax check to file (P-CHAN-00416)

| P-CHAN-00416 | Write results of syntax check to file |
|---|---|
| Description | If this parameter is set to 1, all NC blocks and reported errors checked during the syntax check are logged to a file. The file is created in the controller root directory. |
| Parameter | syn_chk.record_result |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | The filename consists of the 'dec0' string and the NC channel number in which the syntax check was executed. The file extension is '.sc'.<br>Example:<br>Name of log file after a syntax check run in channel 1: *dec01.sc* |

# 1.31 Parameterisation of kinematic transformations

## 1.31.1 Configuration for CNC Build V2.11.2xxx and higher:

### 1.31.1.1 Selection of default kinematic type (P-CHAN-00032)

| P-CHAN-00032 | Select kinematic default transformation (kinematic type) |
|---|---|
| Description | The purpose of kinematic ID is to identify the machine or tool head specific kinematic types implemented in the controller.<br>This parameter defines the default setting for the kinematic transformation to be used. |
| Parameter | kinematik_id |
| Data type | UNS16 |
| Data range | 1 ≤ kinematik_id < 93 (Maximum number of permissible kinematic types: 92, application.specific |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>After the controller starts up, transformation with ID 2 is valid.<br>*kinematik_id 2*<br>For more details on machine kinematics see [KITRA] and [PROG]. |

### 1.31.1.2 Kinematic data (kinematik[i].*)

The specific kinematic offset values of the various tool heads are entered in this structure for the transformations. The related kinematic ID is used for the direct indexing of the associated data set.

| Structure name | Index |
|---|---|
| kinematik[i] | i = 1 ... 92 (Maximum number of permissible kinematic types: 92, application-specific) |

#### 1.31.1.2.1 Kinematic parameters (P-CHAN-00094)

| P-CHAN-00094 | Define kinematic parameters |
|---|---|
| Description | The specific kinematic offsets are entered in this structure for each transformation. |
| Parameter | kinematik[i].param[j] where j = 0... 74 (maximum number of kinematic parameters, application-specific, syntax as of V2.10.1501) |
| Data type | REAL64 |
| Data range | ---- |
| Dimension | 0.1µm or 0.0001° (for offsets) |
| Default value | 0 |
| Remarks | *kinematik[i].wz_kopf_versatz[j] (syntax up to V260)*<br>Kinematic parameters can also be entered in the tool data list P-TOOL-00009 (they are then relevant only when a tool is selected, independent of the kinematics). If a kinematic parameter is assigned in both lists, the specified values are added in the NC.<br>For more details on parameterising kinematic transformation and 5-axis machining, see [KITRA] and [PROG].<br>Parameterisation example: Offset values are specified for kinematics 1, 2 and 5. The kinematic with ID 2 is the default kinematic.<br>`kinematik_id 2 Default kinematics: 2`<br>`#`<br>`kinematik[1].param[0] 1088000`<br>`kinematik[1].param[1] 1987000`<br>`kinematik[1].param[2] 342000`<br>`#`<br>`kinematik[2].param[0] 1538000`<br>`kinematik[2].param[1] 25000`<br>`kinematik[2].param[2] 0`<br>`kinematik[2].param[5] 1800000`<br>`#`<br>`kinematik[5].param[0] 1487000`<br>`kinematik[5].param[1] 25000` |

### 1.31.1.2.2 Configuration of Universal Kinematics

The **Universal Kinematics** require special parameters. They are described in the section Parameters of Universal Kinematics [▸ 173] .

The parameters of Universal Kinematics are filed under **Kinematic ID 91** . For more information see [FCT-C27] and [PROG].

## 1.31.2 Configuration for CNC Builds as of V3.00 and higher:

As of CNC Build V300 and higher, multi-step transformations also can be defined in addition to normal single-step transformations.

**Use kinematic data from CNC Build V2.11.xxxx in CNC Build V3.xx.**

The configuration of transformations in existing channel lists from CNC Build V2.11.xxxx must be adapted if they are to be used in CNC Builds as of V3.00 and higher. Please pay special attention to the differences.

A example of simple migration of a kinematic is contained in [KITRA// Specification of kinematics ID and offset data].

The **Key innovation** is that a kinematic must be specified by the corresponding ID in P-CHAN-00262 [▸ 168] .

### 1.31.2.1 Select default kinematic types for multi-step transformations (P-CHAN-00264)

| **P-CHAN-00264** | **Select default kinematic types for multi-step transformations** |
|---|---|
| Description | A default kinematic ID can be defined for each transformation step. This is valid after the controller is started. The default kinematics must be configured in each of the transformation steps; otherwise an error message is output when the transformations (#TRAFO...) are selected. |
| Parameter | default_id_of_kin_step[i] where i = 0, 1 |
| Data type | UNS16 |
| Data range | 0 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 for i = 0, 1 |
| Remarks | Parameterisation example:<br>After the controller is started, the kinematics are valid where ID 87 applies to the first (..step[0]) and ID 51 to the second (..step[1]) transformation step.<br>*default_id_of_kin_step[0] 87*<br>*default_id_of_kin_step[1] 51*<br>The definition of the first default kinematic can be set either with *default_id_of_kin_step[0]* or in the previous syntax with *kinematik_id* . This is especially useful for the exclusive definition of single-step transformations.<br>For more details on machine kinematics and 5-axis machining, see [KITRA] and [PROG]. |

### 1.31.2.2 Kinematic data (kin_step[i].trafo[j].* / trafo[j].*)

**NOTICE**

As of Version V3.00.3012.00, the structures *kin_step[i].trafo[j].** or *trafo[j].** **replace** the definition of kinematic data programmed with *kinematik[i].**.

The previous structure *kinematik[i].** is no longer supported.

The new structures can continue to be used to configure kinematics for single-step (default) transformations (**Single-Step Transformations**) as previous kinematics.

In addition, the new structures permit the configuration of kinematics which consist of several transformation steps (**Multi-Step Transformations**). This requires parameterisation in separate data records *kin_step[i].**.

Currently, a multi-step transformation may consist of maximum 2 steps.

The transformations of the first transformation step are defined in the structure *kin_step[0]*.

The transformations of the second transformation step are defined in the structures of *kin_step[1]*.

| Structure name | Index |
|---|---|
| kin_step[i] | i = 0 (Step 1), 1 (Step 2) |

A maximum of 10 different transformations can be configured in each transformation step in the structure (*trafo[j].**). Besides the kinematic parameters, the related kinematic ID is entered in the data record as an additional parameter (see description of P-CHAN-00262 [▶ 168] and P-CHAN-00263 [▶ 169]) is inactive.

| Structure name | Index |
|---|---|
| trafo[j] | j = 0 ... 9 |

**Example:**

```
kin_step[0].trafo[j].* ;kinematics for transformation step 1
kin_step[1].trafo[j].* ;kinematics for transformation step 2
```

The definition of a single-step transformation can be created directly in trafo[j].* without specifying the element kin_step[0]. For example, the notation of kin_step[0].trafo[0].* and trafo[0].* is completely equivalent.

The definition of multi-step transformations requires the specification of kin_step[i].*. The sequence of the configuration steps takes place in the direction of forward transformation.

#### 1.31.2.2.1 Definition of kinematic ID (P-CHAN-00262)

| P-CHAN-00262 | Define kinematic ID for multi-step transformations |
|---|---|
| Description | The kinematic ID identifies the related transformation as an element of the data set of the kinematic parameters. |
| Parameter | trafo[j].id<br>kin_step[i].trafo[j].id (multistep transformations) |
| Data type | UNS16 |
| Data range | 1 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter syntax as of V300 and higher |

### 1.31.2.2.2 Kinematic parameters (P-CHAN-00263)

| **P-CHAN-00263** | **Define kinematic parameters for multi-step transformations** |
|---|---|
| Description | Enter the specific kinematic offsets in this structure for each transformation. |
| Parameter | trafo[j].param[k] or where k = 0 to 73 (Maximum number of kinematic parameters)<br>kin_step[i].trafo[j].param[k] (multi-step transformations) |
| Data type | REAL64 |
| Data range | ---- |
| Dimension | 0.1 µm or 0.0001 inch |
| Default value | 0 |
| Remarks | Kinematic parameters can also be entered in the tool data list P-TOOL-00009 (they are then relevant only when a tool is selected, independent on the kinematics).<br>If a kinematic parameter is assigned in both lists, the specified values are added in the NC. This is only valid for transformation step 1.<br>No additional kinematic parameters can be entered in the tool data for transformation step 2.<br>Further details on the parameterisation of a kinematic transformation, see [KITRA] and [PROG].<br>(Parameter syntax as of V300 and higher) |

**Configuration examples:**

**Example 1: Exclusively single step transformations**

Configuration of kinematics 33 and 87. Kinematics with ID 87 are to be the default kinematics.

```
kinematik_id 87
#
trafo[0].id 33
trafo[0].param[0] 1457000
trafo[0].param[1] 0
trafo[0].param[2] 225000
#
trafo[1].id 87
trafo[1].param[0] 1558000
trafo[1].param[1] 140000
trafo[1].param[2] 2000

Alternatively:
default_id_of_kin_step[0] 87
#
kin_step[0].trafo[0].id 33
kin_step[0].trafo[0].param[0] 1457000
kin_step[0].trafo[0].param[1] 0
kin_step[0].trafo[0].param[2] 225000
#
kin_step[0].trafo[1].id 87
kin_step[0].trafo[1].param[0] 1558000
kin_step[0].trafo[1].param[1] 140000
kin_step[0].trafo[1].param[2] 2000
```

**Example 2: Multi-step transformations**

Kinematics 33 and 87 are created in the first step and kinematics 5 and 9 in the second step. After start-up the default kinematics are to be kinematics 87 in the first step and kinematics 5 in the second step.

```
default_id_of_kin_step[0] 87
default_id_of_kin_step[1] 5
#
#First step
kin_step[0].trafo[0].id 33
kin_step[0].trafo[0].param[0] 1457000
kin_step[0].trafo[0].param[1] 0
kin_step[0].trafo[0].param[2] 225000
#
kin_step[0].trafo[1].id 87
kin_step[0].trafo[1].param[0] 1558000
kin_step[0].trafo[1].param[1] 140000
kin_step[0].trafo[1].param[2] 2000
#
#Second step
kin_step[1].trafo[0].id 5
kin_step[1].trafo[0].param[0] 150000
kin_step[1].trafo[0].param[1] 0
kin_step[1].trafo[0].param[2]
#
kin_step[1].trafo[1].id 9
kin_step[1].trafo[1].param[0] 1567000
kin_step[1].trafo[1].param[1] 130000
kin_step[1].trafo[1].param[2] 4000
```

### 1.31.2.2.3 Configuration of Universal Kinematics

The **Universal Kinematics** require special parameters. They are described in the section Parameters of Universal Kinematics [▶ 173] .

The parameters of Universal Kinematics are filed under **Kinematic ID 91** . For more information see [FCT-C27] and [PROG].

### 1.31.2.2.4 Correction values for kinematics (P-CHAN-00438)

| P-CHAN-00438 | Correction values for kinematic parameters |
|---|---|
| Description | This field contains the correction values for kinematic offset values. The correction values are not directly entered by the operator or user. They are the results of calibration cycles. The calibration cycles are supplied by the controller manufacturer [CYCLES-KinOpt]. |
| Parameter | trafo[j].corr[k] where k = 0 to 73 (Maximum number of correction values)<br>kin_step[i].trafo[j].corr[k] (multistep transformations) |
| Data type | STRING |
| Data range | ---- |
| Dimension | ---- |
| Default value | * |
| Remarks | These correction values should not be directly entered by the operator or user. Instead, they are the results of calibration cycles . The calibration cycles are supplied by the controller manufacturer, see [CYCLES-KinOpt]<br>This parameter is the encrypted representation of a REAL64 value with the dimension 0.1 µm or 0.0001 inch.<br>The parameter is available as of V3.1.3068.09<br>* Note: The default value of variables is a blank string. |

### Definition of the kinematic name (P-CHAN-00443)

| P-CHAN-00443 | Definition of the kinematic name |
|---|---|
| Description | The name serves as a unique identifier of the kinematic in case a kinematic ID is configured multiple times. |
| Parameter | trafo[j].name<br>kin_step[i].trafo[j].name (multistep transformations) |
| Data type | STRING |
| Data range | Maximum of 16 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | When a coupling kinematic is configured, the kinematic name must be configured for all partial kinematics.<br>It is only possible to access kinematics using the kinematic ID in the NC program. If a kinematic ID is configured multiple times, it cannot be activated with the command #KIN_ID, for example.<br>Invalid names: DEFAULT, NONE, ON, OFF<br>The following characters are permitted: a-z, A-Z, 0-9, _<br>* Note: The default value of variables is a blank string. |

### Basic offsets for stacked kinematics (P-CHAN-00446)

| P-CHAN-00446 | Definition of the Cartesian basic offsets for stacked kinematics. |
|---|---|
| Description | The basic offsets are active for stacked kinematics and they describe the Cartesian offsets between the MCS of a kinematic and the TCP of the kinematic stacked below it.<br>A robot is situated on a slide on a linear axis. Due to the presence of the slide there is a 10 cm offset in z direction between the linear axis and the base of the robot. Enter this offset for the robot here:<br>*trafo[0].id 45*<br>*trafo[0].base[2] -1000000* |

| | |
|---|---|
| Parameter | trafo[j].base[k] where k=0 … 5<br>kin_step[i].trafo[j].base[k] (multistep transformations) |
| Data type | REAL64 |
| Data range | MIN(REAL64) … MAX(REAL64) |
| Dimension | 0.1µm for linear offsets, 0.0001° for angles |
| Default value | 0.0 |
| Remarks | The basic offsets are only considered when the coupling kinematic is active. |

## Name of the default kinematic after start-up (P-CHAN-00454)

| **P-CHAN-00454** | **Name of the default kinematic after start-up** |
|---|---|
| Description | If the kinematic ID is configured multiple times, this parameter defines the unique name of the default kinematic after start-up. |
| Parameter | kinematik_name |
| Data type | STRING |
| Data range | Maximum of 16 characters |
| Dimension | ---- |
| Default value | - |
| Remarks | Parameterisation example:<br>The valid default kinematic after start-up has the ID 2 with the name “test”.<br>*kinematik_id 2*<br>*kinematik_name test* |

## Select default kinematic names for multi-step transformations (P-CHAN-0452)

| **P-CHAN-00452** | **Select default kinematic names for multi-step transformations** |
|---|---|
| Description | The name serves as a unique identifier of the kinematic in case a kinematic ID is configured multiple times.<br>A default kinematic name can be defined for each transformation step. This is valid after the controller is started. The default kinematics must be configured in each of the transformation steps; otherwise an error message is output when the transformations (#TRAFO...) are selected. |
| Parameter | default_name_of_kin_step[i] where i = 0, 1 |
| Data type | STRING |
| Data range | Maximum of 16 characters |
| Dimension | ---- |
| Default value | - |
| Remarks | Parameterisation example:<br>After controller start-up, the transformation with ID2 and the name"test’ is valid for the first transformation step (.step[0]).<br>*default_id_of_kin_step[0] 2*<br>*default_name_of_kin_step[0] test* |

## 1.31.3 Parameters of the Universal Kinematics (kinematik[91].*, trafo[j].*

Universal Kinematics parameters can be configured in the structure *kinematik[91].** in CNC Builds V2.11.2xxx and in the structures of a kinematic ID 91 for simple and multiple transformations *trafo[j].** and *kin_step[i].trafo[j].** in CNC Builds as of V3.00 and higher.

Further details on Universal Kinematics are contained in the function description [FCT-C27].

### 1.31.3.1 Zero orientation of the tool (P-CHAN-00285)

| **P-CHAN-00285** | **Zero orientation of the tool (Universal Kinematics)** |
|---|---|
| Description | This parameter defines the tool orientation in the zero setting (vector X, Y, Z, tool direction). |
| Parameter | trafo[j].zero_orientation[k] where k = 0, 1, 2<br>kin_step[i].trafo[j].zero_orientation[k] (multistep transformations)<br>kinematik[91].zero_orientation[k] (up to Version V2.11.28xx) |
| Data type | REAL64 |
| Data range | ---- |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.31.3.2 Zero position of the tool (P-CHAN-00286)

| **P-CHAN-00286** | **Zero position of the tool (Universal Kinematics)** |
|---|---|
| Description | This parameter defines the tool position in the zero position (position X, Y, Z, home position). |
| Parameter | trafo[j].zero_position[k] where k = 0, 1, 2<br>kin_step[i].trafo[j].zero_position[k] (multistep transformations)<br>kinematik[91].zero_position[k] (up to Version V2.11.28xx) |
| Data type | REAL64 |
| Data range | ---- |
| Dimension | 0.1 µm or 0.0001 inch |
| Default value | 0 |
| Remarks | |

### 1.31.3.3 Angle transformation (P-CHAN-00287)

| **P-CHAN-00287** | **Angle transformation (Universal Kinematics)** |
|---|---|
| Description | When angle programming mode is active (P-CHAN-00288 [▶ 174]), this parameter defines how the programmed angle is treated. |
| Parameter | trafo[j].rtcp<br>kin_step[i].trafo[j].rtcp (multistep transformations)<br>kinematik[91].rtcp (up to Version V2.11.28xx) |
| Data type | REAL64 |
| Data range | 0: Complete transformation: transforming angle within the range (-p, p] and output to the machine (default).<br>1: RTCP transformation: programmed angles are output directly to the machine. |

| | |
|---|---|
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.31.3.4 Programming mode (P-CHAN-00288)

| **P-CHAN-00288** | **Programming mode (Universal Kinematics)** |
|---|---|
| Description | The programming mode defines the tool orientation from the programmed values (point-vector mode or angle modes). Alternatively the mode can also be set by the channel parameter P-CHAN-00112 [▶ 181] .<br>A setting of the kinematics has priority over P-CHAN-00112 [▶ 181]. |
| Parameter | trafo[j].programming_mode<br>kin_step[i].trafo[j].programming_mode (multistep transformations)<br>kinematik[91].programming_mode (up to Version V2.11.28xx) |
| Data type | REAL64 |
| Data range | 10: Point-vector programming. Tool orientation is programmed by the axes U, V, W. The vector [U, V, W] need not be normalised but neither may it be the zero vector.<br>11: Free programming. Not currently supported.<br>12: Direct programming. The configured kinematic chain is used to calculate the tool position and orientation from programmed Cartesian coordinates and angles.<br>13: Conformal programming. Same as direct programming but without any axis offsets, shifts or direction flags. For example, enables the programming of 45° axis positions.<br>14: AB programming.<br>15: BA programming.<br>16: AC programming.<br>17: CA programming.<br>18: BC programming.<br>19: BC programming. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.31.3.5 Number of axes (P-CHAN-00289)

| **P-CHAN-00289** | **Number of axes (Universal Kinematics)** |
|---|---|
| Description | Parameter defines the number of axes of the kinematic chain. |
| Parameter | trafo[j].number_of_axes<br>kin_step[i].trafo[j].number_of_axes (multistep transformations)<br>kinematik[91].number_of_axes (up to Version V2.11.28xx) |
| Data type | REAL64 |
| Data range | 3 ... 6 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | The parameter is also available in multi-step transformations. Access to the parameter is:<br>kin_step[i].trafo[j].number_of_axes |

### 1.31.3.6 Axis sequence (P-CHAN-00290)

| **P-CHAN-00290** | **Axis sequence (Universal Kinematics)** |
|---|---|
| Description | Parameter defines the axis sequence in the kinematic chain. |
| Parameter | trafo[j].chain[k] where k = 0 ... P-CHAN-00289 [▶ 174] – 1<br>kin_step[i].trafo[j].chain[k] (multistep transformations)<br>kinematik[91].chain[k] (up to Version V2.11.28xx) |
| Data type | REAL64 |
| Data range | ---- |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.31.3.7 Axis-specific data (kinematik[91].axis[k].*, trafo[j].axis[k].*)

Additional properties for each of the axes in the kinematic chain are configured in the structure element 'axis[k]'.

| **Structure name** | **Index** |
|---|---|
| axis[k] | k = 0 ... P-CHAN-00289 [▶ 174] - 1 |

### 1.31.3.7.1 Axis type (P-CHAN-00291)

| **P-CHAN-00291** | **Axis type (Universal Kinematics)** |
|---|---|
| Description | Parameter defines the axis type. |
| Parameter | trafo[j].axis[k].type<br>kin_step[i].trafo[j].axis[k].type (multistep transformations)<br>kinematik[91].axis[k].type (up to Version V2.11.28xx) |
| Data type | REAL64 |
| Data range | 1: Translator<br>2: Rotator |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.31.3.7.2 Axis orientation (P-CHAN-00292)

| **P-CHAN-00292** | **Axis orientation (Universal Kinematics)** |
|---|---|
| Description | Parameter defines the direction vector (X, Y, Z, no zero vector) of the axis. |
| Parameter | trafo[j].axis[k].orientation[i] where i = 0, 1, 2<br>kin_step[i].trafo[j].axis[k].orientation[i] (multistep transformations)<br>kinematik[91].axis[k].orientation[i] (up to on V2.11.28xx) |
| Data type | REAL64 |
| Data range | ---- |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.31.3.7.3 Interpolation point on the axis (P-CHAN-00293)

| **P-CHAN-00293** | **Interpolation point on the axis (Universal Kinematics)** |
|---|---|
| Description | Parameter defines an interpolation point on the axis (position X, Y, Z, only relevant for rotary axes). |
| Parameter | trafo[j].axis[k].point[l] where l = 0, 1, 2<br>kin_step[i].trafo[j].axis[k].point[l] (multistep transformations)<br>kinematik[91].axis[k].point[l] (up to Version V2.11.28xx) |
| Data type | REAL64 |
| Data range | ---- |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | |

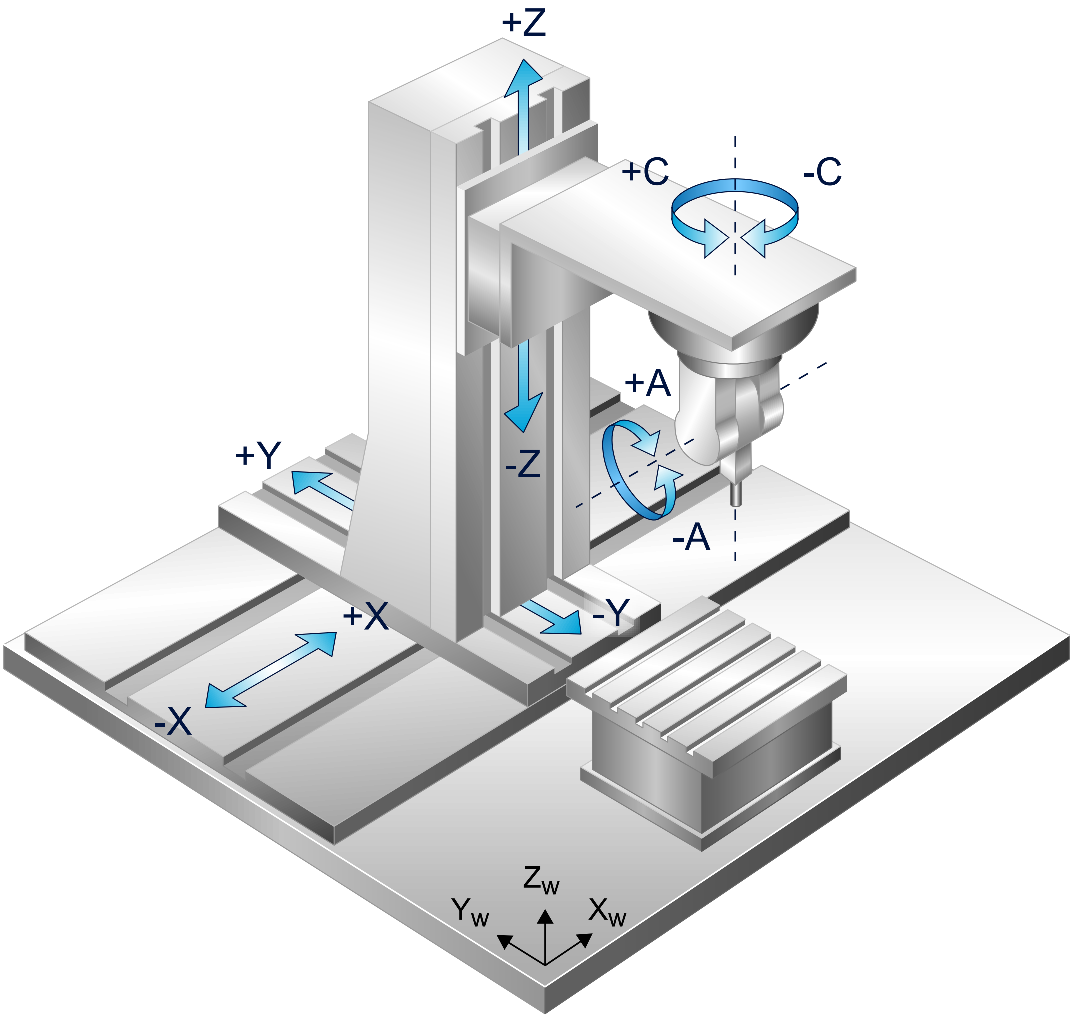


Fig. 34: Configuration example of a mapping of the classic CA kinematics according to Kinematic ID 9 (Y, X, Z, C, A) onto the configuration of Universal Kinematics

**Configuration of universal kinematics in channel parameter list:**

```
# Zero orientation of the tool
# Tool points in Z direction

kinematik[91].zero_orientation[0] 0
kinematik[91].zero_orientation[1] 0
kinematik[91].zero_orientation[2] 1

# Zero position of the tool
# Tool rests at point (12000, -3200, 500)

kinematik[91].zero_position[0] 12000
kinematik[91].zero_position[1] -3200
kinematik[91].zero_position[2] 500

kinematik[91].number_of_axes 5

kinematik[91].programming_mode 17
kinematik[91].rtcp 1

# Define X axis (index 0)
kinematik[91].axis[0].type 1
kinematik[91].axis[0].orientation[0] 1
kinematik[91].axis[0].orientation[1] 0
kinematik[91].axis[0].orientation[2] 0
...

# Define Y axis (index 1)
kinematik[91].axis[1].type 1
kinematik[91].axis[1].orientation[0] 0
kinematik[91].axis[1].orientation[1] 1
kinematik[91].axis[1].orientation[2] 0
...
# Define Z axis (index 2)
kinematik[91].axis[2].type 1
kinematik[91].axis[2].orientation[0] 0
kinematik[91].axis[2].orientation[1] 0
kinematik[91].axis[2].orientation[2] 1
...
# Define C axis (index 3)
# points in Z direction and crosses
# point (800, 1200, 0)
kinematik[91].axis[3].type 2
kinematik[91].axis[3].orientation[0] 0
kinematik[91].axis[3].orientation[1] 0
kinematik[91].axis[3].orientation[2] 1
kinematik[91].axis[3].point[0] 800
kinematik[91].axis[3].point[1] 1200
kinematik[91].axis[3].point[2] 0

# Define A axis (index 4)
kinematik[91].axis[4].type 2
kinematik[91].axis[4].orientation[0] 1
kinematik[91].axis[4].orientation[1] 0
kinematik[91].axis[4].orientation[2] 0
...
# Sequence in kin. chain: YXZCA
kinematik[91].chain[0] 1 # Y axis
kinematik[91].chain[1] 0 # X axis
kinematik[91].chain[2] 2 # Z axis
kinematik[91].chain[3] 3 # C axis
kinematik[91].chain[4] 4 # A axis
```

## 1.31.3.8 Mode for the transformation between axis values and Cart. coordinates (P-CHAN-00294)

| **P-CHAN-00294** | **Mode for the transformation between axis values and Cartesian coordinates (Universal Kinematics)** |
|---|---|
| Description | Sets the mode for the transformation between axis values and Cartesian coordinates. |
| Parameter | trafo[j].linkage_mode<br>kin_step[i].trafo[j].linkage_mode (multistep transformations)<br>kinematik[91].linkage_mode (up to Build V2.11.28xx) |

| | |
|---|---|
| Data type | REAL64 |
| Data range | 0 transformation deactivated<br>1 transformation activated |
| Dimension | --- |
| Default value | 0 |
| Remarks | |

### 1.31.3.9 Transformation between axis values and Cart. coordinates (P-CHAN-00295)

| P-CHAN-00295 | Transformation between axis values and Cartesian coordinates (Universal Kinematics) |
|---|---|
| Description | Definition of a matrix and an offset vector to describe a linear transformation between axis values of the linear axes and Cartesian coordinates.<br>$$\begin{bmatrix} X_{cart} \\ Y_{cart} \\ Z_{cart} \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}$$<br>Rotary axes are not included in the calculation. The letters X, Y, Z stand here for the three linear axes of the Universal Kinematics in the order of their definition in the parameter trafo[].axis[] (P-CHAN-00293 [▸ 176]) is inactive.<br>The matrix is defined in trafo[].linkage[0-2][0-2]. The first index specifies the line number; the second index specifies the column number; both are 0-based.<br>The offset vector is defined in trafo[].linkage[0-2][3].<br>If the matrix is not invertible, the error ID 292010 is output.<br>For configuration example, see [FCT-C27// Transformation between axis values and Cartesian coordinates]. |
| Parameter | trafo[j].linkage[k][l] where k = 0, 1, 2 and l = 0, 1, 2, 3<br>kin_step[i].trafo[j].linkage[k][l] (multistep transformations)<br>kinematik[91].linkage[k][l] (up to Build V2.11.28xx) |
| Data type | REAL64 |
| Data range | |
| Dimension | for the matrix: ----<br>for the offset vector: 0.1 µm |
| Default value | 0 |
| Remarks | |

## 1.31.4 Parameters of the coupling kinematic

**This function is available as of CNC Build V3.1.3080**

### 1.31.4.1 Group name of a coupling kinematic (P-CHAN-00447)

| P-CHAN-00447 | Definition of a group name of a coupling kinematic |
|---|---|
| Description | A coupling kinematic group includes a number of stacked partial kinematics and is uniquely identifiable by its name. |

| | |
|---|---|
| Parameter | trafo[j].group[k].name<br>kin_step[i].trafo[j].group[k].name (multistep transformations) |
| Data type | STRING |
| Data range | Maximum of 16 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | As of Build V300<br>A group name may not be assigned more than once within a coupling kinematic.<br>* |

### 1.31.4.2 Definition of a group containing the tool of a coupling kinematic

| **P-CHAN-00448** | **Definition of a workpiece CS for a group of a couple kinematic.** |
|---|---|
| Description | If a tool CS is specified (group name of another group of the couple kinematic), all TCP coordinates of this group are interpreted in the TCP system of the group specified as the workpiece CS.<br>This group therefore follows the movements of the workpiece CS group. |
| Parameter | trafo[j].group[k].workpiece_cs<br>kin_step[i].trafo[j].group[k].workpiece_cs (multistep transformations) |
| Data type | STRING |
| Data range | Maximum of 16 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | Configuration example:<br>`trafo[0].id                 210`<br>`trafo[0].group[0].name      CA_TABLE`<br>`trafo[0].group[1].name      ROBOT`<br>`trafo[0].group[1].workpiece_cs   CA_TABLE`<br>After the kinematic is selected, the group ROBOT follows the movements of CA_TABLE since it is configured as the tool coordinate system of ROBOT.<br><br>* Note: The default value of variables is a blank string. |

### 1.31.4.3 Kinematic chain of a group (P-CHAN-00449)

| **P-CHAN-00449** | **Definition of a kinematic chain for a group of a coupling kinematic** |
|---|---|
| Description | Description of the sequence in which the partial kinematics within a coupling kinematic are stacked on top of each other. Index m = 0 points to the lowest kinematic. |
| Parameter | trafo[j].group[k].chain[l]<br>kin_step[i].trafo[j].group[k].chain[l] (multistep transformations) |
| Data type | STRING |
| Data range | Maximum of 16 characters |
| Dimension | ---- |
| Default value | * |

| | |
|---|---|
| Remarks | Each kinematic used here must be configured as an autonomous kinematic with a corresponding name.<br>Configuration example: Robot on a linear axis<br>*trafo[0].id 210*<br>*trafo[0].group[0].name LIN_ROB*<br>*trafo[0].group[0].chain[0] LINEAR*<br>*trafo[0].group[0].chain[1] ROBOT*<br>*trafo[1].id 45*<br>*trafo[1].name ROBOT*<br>*trafo[2].id 91*<br>*trafo[2].name LINEAR*<br>* Note: The default value of variables is a blank string. |

### 1.31.4.4 Sequence of TCP movements of partial kinematics (P-CHAN-00450)

| **P-CHAN-00450** | **Priority which the couple kinematic uses to split the TCP movement among the partial kinematics.** |
|---|---|
| Description | The algorithm which distributes the programmed TCP coordinates among the partial kinematics precedes the priority defined here: Beginning with the index m = 0 the largest possible part of the movement is to be traversed by this kinematic. The resulting difference to the programmed TCP is handed over to the next kinematic in the list. This step is repeated until the TCP coordinates are reached, or until the last entry within the sequence defined here is reached. |
| Parameter | trafo[j].group[k].move_prio[m] where m = 0 … Length of the kinematic chain<br>kin_step[i].trafo[j].group[k].move_prio[m] (multistep transformations) |
| Data type | STRING |
| Data range | Maximum of 16 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | Each of the names used here must be present within the kinematic chain.<br>Configuration example: Robot on a linear axis<br>*trafo[0].id 210*<br>*trafo[0].group[0].name LIN_ROB*<br>*trafo[0].group[0].chain[0] LINEAR*<br>*trafo[0].group[0].chain[1] ROBOT*<br>*trafo[0].group[0].move_prio[0] ROBOT*<br>*trafo[0].group[0].move_prio[1] LINEAR*<br>*trafo[1].id 45*<br>*trafo[1].name ROBOT*<br>*trafo[2].id 91*<br>*trafo[2].name LINEAR*<br>In this example, the entire TCP movement is first handed over to the robot. What the robot cannot clear is then adopted by the linear axis.<br>* Note: The default value of variables is a blank string. |

### 1.31.4.5 Lock the Cartesian degree of freedom of the coupling kinematic (P-CHAN-00458)

| **P-CHAN-00458** | **Lock the Cartesian degrees of freedom of the couple kinematic** |
|---|---|

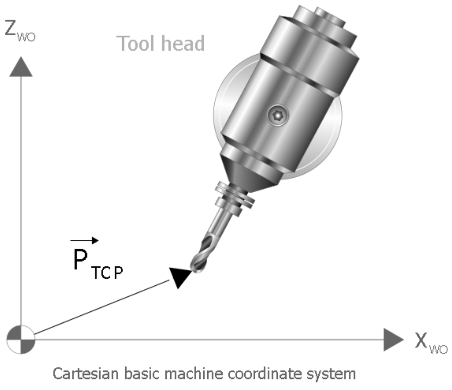
| | |
|---|---|
| Description | This parameter influences the behaviour of the coupling kinematic with a programmed TCP.<br>While the TCP movement is distributed among the individual partial kinematics, the locked axes are not moved.<br>Index k describes the axis index to be locked. A parameter value > 0 marks the axis with axis index k as locked. |
| Parameter | trafo[i].lock_dof[k]<br>kin_step[i].trafo[j].lock_dof[k] (multistep transformations) |
| Data type | BOOLEAN |
| Data range | 0 / 1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | A locked axis may continue to be programmed directly by its axis identifier.<br>Configuration example:<br>*trafo[0].id 210*<br>*trafo[0]. lock_dof[7] 1    (The axis with Index 7 is not moved while*<br>*(TCP movement is distributed* |

## 1.31.5 Orientation angle mode (P-CHAN-00112)

| **P-CHAN-00112** | **Mode of orientation angle programming for kinematic transformations** |
|---|---|
| Description | With a complete transformation, orientation about the coordinate system axes can be programmed either by using an orientation vector with the three components U, V, W or by using three rotation angles A, B, C depending on the transformation type.<br>Due to the additional degree of freedom provided by manual orientation, programming the rotation angles A, B, C is often the property of robot structures.<br>The sequence of the three rotations about the assigned rotary axes X, Y, Z leads to the required target orientation or to the target effector coordinate system. If not otherwise defined, single rotations are executed in a mathematically positive direction about the coordinate system axes which are to be reset.<br>The starting point is an axis sequence with the Cartesian axes X, Y, Z and the rotary axes A, B, C. The default assignment of rotations about the coordinate system axes is A -> X, B -> Y, C -> Z. This may deviate with special angle modes.<br>Some kinematics use special sequences of rotation which are not listed here. In this case, switching using P-CHAN-00112 is not possible. P-CHAN-00112 has no measuring with standard five-axis kinematics.<br>Special values can be entered in P-CHAN-00112 for Universal Kinematics (KIN-ID91). |
| Parameter | ori_rotation_angle |
| Data type | SGN16 |
| Data range | -1: Programmed orientation axes are forwarded to kinematic transformation without any changes. Any Cartesian transformation which may be active with an active rotation has no influence on these orientation axes<br>**Complete kinematics transformations**<br>**3 degrees of freedom for orientation**<br>0: YPR (Yaw Pitch Role) sequence of rotation: 1st rotation about Z (C), 2nd rotation negative about Y´ (B), 3rd rotation about X´´ (A) (default)<br>1: Euler, order of rotation: 1. rotation about Z (C), 2nd rotation about Y´ (B), 3. rotation about Z´´ (C)<br>2: CBA, similar to YPR with positive B rotation and different axis assignment. Rotation about Z (A), 2nd rotation about Y´ (B), 3rd rotation about X´´ (C).<br>-> A15 B-90 C20 (CBA) is identical to A20 B90 C15 (YPR).<br>3: CAB rotation sequence, 1st rotation about Z (C), 2nd rotation about X’ (A), 3rd rotation about Y´’ (B) (as of Build V3.1.3079.35) |

| | |
|---|---|
| | 4: CBA_STD, corresponds to CBA with a different axis assignment Rotation sequence: 1. rotation about Z(C), 2nd rotation about Y‘(B), 3rd rotation about X‘‘(A) -> A15 B90 C20 (CBA_STD) is identical to A20 B90 C15 (CBA) and A15 B-90 C20 (YPR). (as of V3.1.3079.9)<br>5: ABC rotation sequence, 1st rotation about X (A), 2nd rotation about Y´ (B), 3rd rotation about Z´´ (C). (as of Build V3.1.3079.35)<br>**2 degrees of freedom for orientation (cf. KIN-ID 91)**<br>14: AB Rotation sequence: 1. rotation about X(A), 2nd rotation about Y‘ (B) (as of Build V3.1.3079.30)<br>15: BA Rotation sequence: 1. rotation about Y(B), 2nd rotation about X‘(A) (as of Build V3.1.3079.30) 2nd<br>**Universal kinematic transformations (KIN-ID 91):**<br>10: Point-vector programming. Tool orientation is programmed by the axes U, V, W. The vector [U, V, W] need not be normalised but neither may it be the zero vector.<br>11: Free programming. Not currently supported.<br>12: Direct programming. The configured kinematic chain is used to calculate the tool position and orientation from programmed Cartesian coordinates and angles.<br>13: Conformal programming. Same as direct programming but without any axis offsets, shifts or direction flags. For example, allows the programming of 45° axis positions.<br>14: AB programming. 15: BA programming. 16: AC programming.<br>17: CA programming. 18: BC programming. 19: CB programming. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.31.6 Activation of TCP display data (P-CHAN-00145)

| **P-CHAN-00145** | **Activation of TCP display data** |
|---|---|
| Description | This parameter is used to activate W0 display data (TCP position referred to the Cartesian basic coordinate system of the machine - MCS). The TCP position is calculated dependent on the active kinematic ID based on the current command axis coordinates, the selected tool (length) and the kinematic offset parameters. The calculation also takes place when transformation is inactive. All axes in the kinematic structure must exist in the channel.<br> Z<sub>WO</sub><br>Tool head<br>P<sub>TCP</sub><br>X<sub>WO</sub><br>Cartesian basic machine coordinate system |
| Parameter | kin_trafo_display |
| Data type | UNS16 |
| Data range | 0: MCS display function inactive (default)<br>1: MCS display function active |

| | |
|---|---|
| | 2: MCS display function active (only for multistep transformation, see Addendum) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | The axes must be homed to obtain the correct display.<br>Programmed tool offsets (V.G.WZ_AKT.V.*) are only considered if they are followed by the programming of #KIN ID[<kinematic-ID>].<br>**Note:**<br>As of CNC Build V3.1.3105 the data type of the parameter changed from BOOLEAN to UNS16. |

### Supplements for multistep kinematic transformations

A kinematic step can be defined in the parameter lists or in the NC program. A kinematic step is activated by programmed the NC command #TRAFO ON.

For multistep transformations, see Concatenating transformations, multistep transformations.

Data value 0 (default):

Kinematic transformations are only executed to display axis positions when they are activated.

| | **PCS not active** | **PCS active** |
|---|---|---|
| Kin. step 0 = defined,<br>Kin. step 1 = defined | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f($CS_{active}$) |
| Kin. step 0 = active,<br>Kin. step 1 = defined | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0)<br>PCS = f(MCS, $CS_{active}$) |
| Kin. step 0 = defined,<br>Kin. step 1 = active | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f(MCS, $CS_{active}$) |
| Kin. step 0 = active,<br>Kin. step 1 = active | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f(MCS, $CS_{active}$) |

Data value 1:

| | **PCS is not active** | **PCS active** |
|---|---|---|
| Kin. step 0 = defined,<br>Kin. step 1 = defined | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f(MCS, $CS_{active}$) |
| Kin. step 0 = active,<br>Kin. step 1 = defined | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f(MCS, $CS_{active}$) |
| Kin. step 0 = defined,<br>Kin. step 1 = active | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f(MCS, $CS_{active}$) |
| Kin. step 0 = active,<br>Kin. step 1 = active | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f(MCS, $CS_{active}$) |

Data value 2:

Kinematic transformations are always executed to display axis positions as soon as they are activated.

| | **PCS not active** | **PCS active** |
|---|---|---|
| Kin. step 0 = defined,<br>Kin. step 1 = defined | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f(MCS, $CS_{active}$) |
| Kin. kin. step 0 = active,<br>Kin. kin. step 1 = defined | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f(MCS, CSactive) |
| Kin. kin. step 0 = defined,<br>Kin. kin. step 1 = active | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f(MCS, $CS_{active}$) |

| | | |
|---|---|---|
| Kin. kin. step 0 = active,<br>Kin. kin. step 1 = active | MCS = f(kin. step 0, kin. step 1) | MCS = f(kin. step 0, kin. step 1)<br>PCS = f(MCS, $CS_{active}$) |

## 1.31.7 Switchover of TCP display data (P-CHAN-00184)

| P-CHAN-00184 | Switch over TCP display data between commanded and actual positions |
|---|---|
| Description | When the W0 display function is active (P-CHAN-00145 [▸ 182]), this parameter switches over between command and actual positions. |
| Parameter | kin_trafo_display_curr_pos |
| Data type | BOOLEAN |
| Data range | 0: Display of W0 command positions (default)<br>1: Display of W0 actual positions |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.31.8 Parameters for automatic alignment function (tool_ori_cs.*)

Using the CNC command '#TOOL ORI CS' followed by a motion block with main axis movement, the CNC can generate an orientation movement that aligns the tool, e.g. perpendicular to the first two main axes of the workpiece coordinate system, ([PROG]).

### 1.31.8.1 Axis in parallel with the tool (P-CHAN-00188)

| P-CHAN-00188 | Define the main axis of the current coordinate system to which the tool is to be aligned |
|---|---|
| Description | This parameter can define the axis of the current coordinate system to which the tool is to be aligned. |
| Parameter | tool_ori_cs.axis |
| Data type | UNS16 |
| Data range | 1: Alignment takes place with 1st main axis of current coordinate system X<br>2: Alignment takes place with 2nd main axis of current coordinate system Y<br>3: Alignment takes place with 3rd main axis of current coordinate system Z |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.31.8.2 Selecting the orientation resolution (P-CHAN-00189)

| P-CHAN-00189 | Selecting the orientation resolution |
|---|---|
| Description | Typically, several axis positions are possible for an orientation resolution on a coordinate system. With a C-A tool head, for example, there is an alternative position C=180° and A=-45° for the axis position C=0° and A=45°.<br>This parameter defines which axis position is used for the orientation movement (see figure). |
| Parameter | tool_ori_cs.mode |
| Data type | UNS16 |

| | |
|---|---|
| Data range | 0: Align to axis position 1 or 2 depending on internal algorithm (default, in general not shortest way)<br>1: Align depending on shortest way<br>2: Prio1: Align taking SLS limits,<br>Priority 2: Align depending on shortest way<br>3: Align to axis position 1<br>4: Align to axis position 2 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

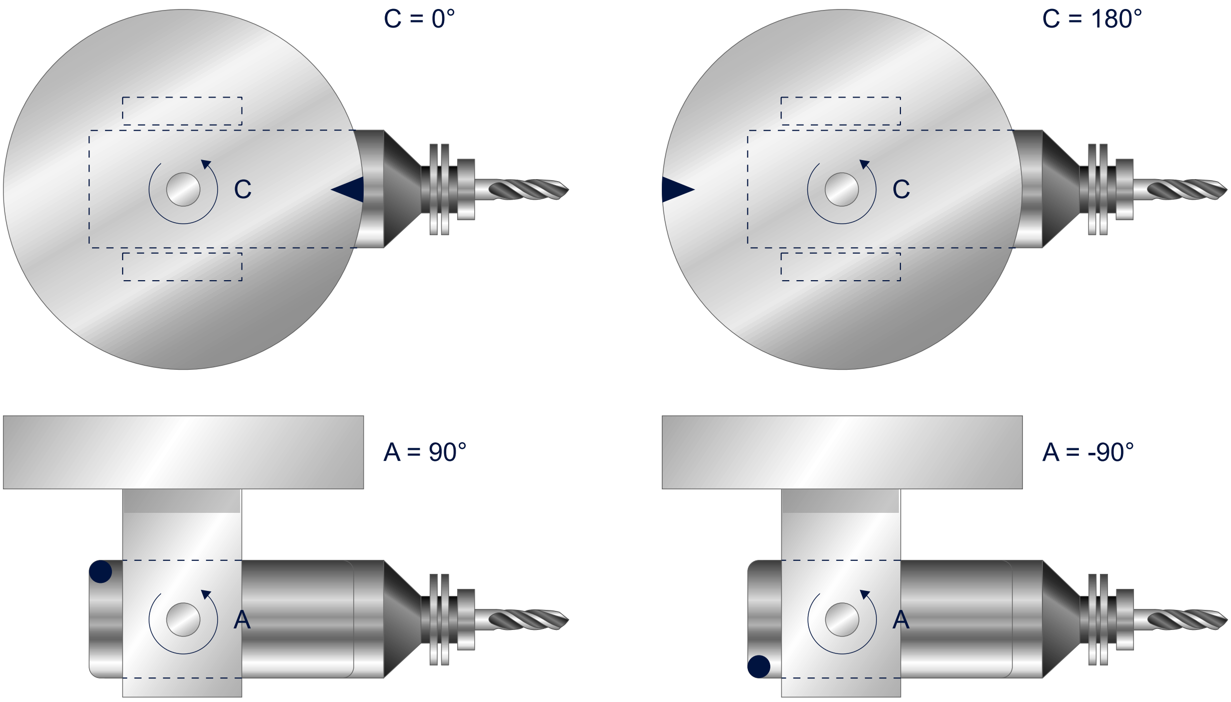


Fig. 35: Selecting the orientation resolution

## 1.31.9 Transformation with gaps in the axis (P-CHAN-00213)

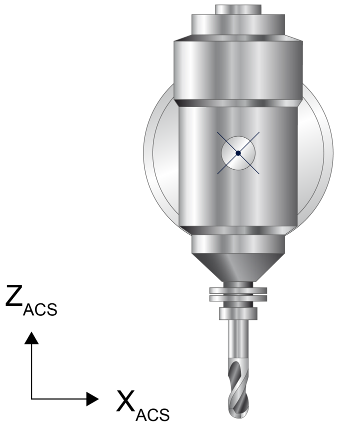
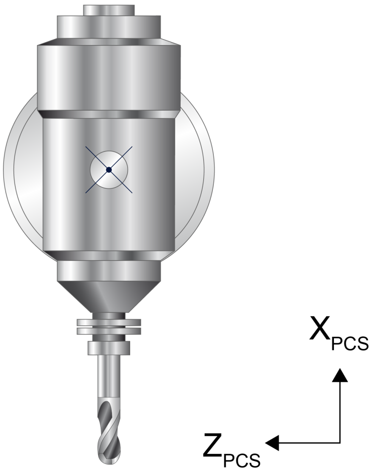
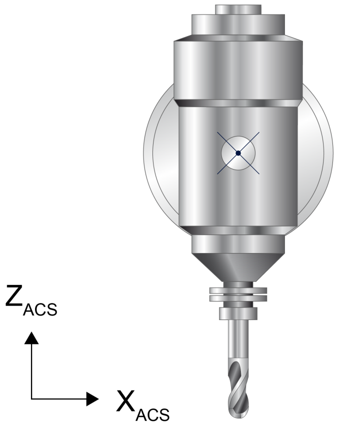
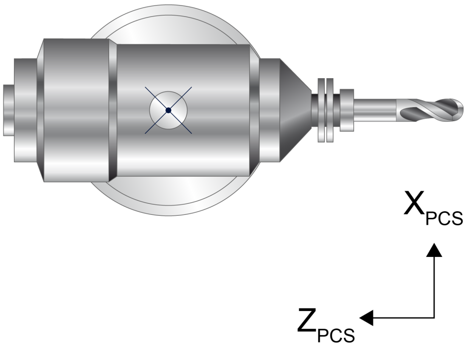
| P-CHAN-00213 | Permit gapped axis configuration when a transformation is selected |
|---|---|
| Description | When Cartesian or kinematic transformations are used, an axis configuration free from gaps is expected. When selected, the CNC checks whether the required number of axes exists without gaps in the current configuration and can complete the axis configuration with an inserted simulation axis.<br>In special cases, it may be necessary to permit axis configurations with gaps, e.g. a tool head without physical Z slides. An axis configuration with gaps is recognisable by the increasing non-linear axis index sequence in the CNC axis exchange command, see example below.<br>Axis configuration with no gaps using NC command:<br>#SET AX[X, 1, 0][Y, 2, 1][Z, 3, 2][C, 4, 3]<br>Axis configuration with gaps using NC command:<br>#SET AX[X, 1, 0][Y, 2, 1][C, 4, 3] |
| Parameter | gap_in_trafo_axis_sequence |

| | |
|---|---|
| Data type | BOOLEAN |
| Data range | 0: Axis configuration with gaps not permitted (default).<br>1: Axis configuration with gaps permitted. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.31.10 Tool orientation within the machining coordinate system (P-CHAN-00247)

| **P-CHAN-00247** | **Mapping of the tool orientation in the active machining coordinate system** |
|---|---|
| Description | In combination with complete kinematic five-axis transformations, this parameter can define whether the programmed tool orientation refers to the Cartesian basic coordinate system of the machine or whether it is mapped to the active machining coordinate system (#CS[], #ACS[]). |
| Parameter | ori_wcs |
| Data type | BOOLEAN |
| Data range | 0: No mapping of tool orientation (default).<br>1: Mapping of tool orientation. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is only effective with Cartesian basic structures. |

**Tool orientation in the active machining coordinate system**

| P-CHAN-00247<br>ori_wcs | …<br>`G00 G90 C0 B0 X0 Z100`<br>… | `#CS ON[0,0,0,90,0,0]`<br>`G00 G90 C0 B0 X0 Z100`<br>… |
|---|---|---|
| 0 |  |  |
| 1 |  |  |

## 1.31.11 Permissible calculation tolerance with forward / backward transformation (P-CHAN-00228)

| P-CHAN-00228 | Permissible calculation tolerance with forward / backward transformation |
|---|---|
| Description | Especially when self-created kinematic transformations are used (see [McCOM-TRAFO]), this parameter can maintain the required position consistency in the NC channel.<br>The result of the calculation sequence (Input values - Forward transformation - Backward transformation - Output values) must match the input values within specific limits.<br>Therefore, the CNC checks whether these deviations are within the permissible tolerance according to the parameter and generates an error message if necessary. The default tolerance value here is 0.3 µm.<br>Due to internal algorithms (e.g. iterative solution method, numeric inaccuracies), this may lead to greater deviations which then require adjustment using this parameter. |
| Parameter | permitted_trafo_deviation_limit |
| Data type | UNS32 |

| | |
|---|---|
| Data range | 0 < permitted_trafo_deviation_limit < MAX(UNS32) |
| Dimension | 0.1µm or 0.0001° |
| Default value | 3 |
| Remarks | Setting large tolerances should be avoided since this may cause relatively large position jumps which can lead to the response of the dynamic monitoring function.<br>The standard tolerance value here should be maximum 0.5 microns.<br>Therefore, when self-created kinematic transformations are used, the internal algorithms must first be checked before setting excessive tolerance values. |

## 1.31.12 Implicit selection of kinematic transformation (P-CHAN-00151)

| **P-CHAN-00151** | **Implicit selection of kinematic transformation at program start** |
|---|---|
| Description | With many machine types, it makes sense to program only path motions within the workpiece coordinate system. Implicit selection/deselection of the configured default transformation P-CHAN-00032 [▶ 165] /P-CHAN-00264 [▶ 167] in the CNC controller allows selection/deselection to be omitted in the NC program.<br>Using this parameter, the configured transformation can be selected automatically at program start-up.<br>A following manual block or a movement using manual mode is then carried out with an active transformation.<br>This is based on the condition that all the axes involved in the transformation are homed. This in turn in only possible with absolute encoders, which means that P-AXIS-00014 must be assigned the value 1 for all the axes involved.<br>If an axis involved in the transformation of the involved axes is not homed, error ID 21415 is output for simulation drives and with real drives, error ID 21425 is output. |
| Parameter | auto_enable_kin_trafo |
| Data type | BOOLEAN |
| Data range | 0: No automatic deselection of configured transformation (default).<br>1: Automatic selection of configured transformation at program start. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.31.13 Implicit deselection of kinematic transformation (P-CHAN-00152)

| **P-CHAN-00152** | **Implicit deselection of kinematic transformation at program end** |
|---|---|
| Description | With many machine types it makes sense to program only path motions within the workpiece coordinate system. Implicit selection/deselection of the configured default transformation P-CHAN-00032 [▶ 165] / P-CHAN-00264 [▶ 167] in the CNC controller allows selection/deselection to be omitted in the NC program.<br>This parameter can deselect the configured transformation automatically at program end. |
| Parameter | auto_disable_kin_trafo |
| Data type | BOOLEAN |
| Data range | 0: No automatic deselection of configured transformation (default).<br>1: Automatic deselection of configured transformation at program end. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.31.14 Mode of kinematic transformation (P-CHAN-00456)

| **P-CHAN-00456** | **Properties of kinematic transformation** |
|---|---|
| Description | This parameter can influence the behaviour or options of the kinematic transformation in the channel.<br>Possible settings:<br>• NONE: Default behaviour<br>• FAST_TARGET_POS_CALC: Deactivate multiple calls of the kinematic robot transformation to calculate target coordinates in the display.<br>• BEVEL_ANGLE_TOOL_LENGTH_CORR: Angle-dependent tool length compensation is activated for kinematic transformations which support this function.<br>• BEVEL_ANGLE_LATERAL_OFFSET: Angle-dependent tool length compensation and offset correction is activated for kinematic transformations which support this function.<br>**Example**:<br>With complete transformations, e.g. an articulated robot, the ACS target point calculation for the display data is based on pre-interpolated interpolation points starting with PCS start through to PCS target In general, this is necessary due to the ambiguity of the backward transformation of the robot. Cyclical multiple calls lead to a higher realtime load in the cyclic CNC task. If this higher load is not required, it can be deactivated by setting the parameter named above to<br>FAST_TARGET_POS_CALC<br>. Due to the ambiguity of the kinematic backward transformation, the pre-calculated ACS target point may possibly deviate from the real interpolated ACS target point in the display data. |
| Parameter | trafo_mode |
| Data type | String |
| Data range | NONE<br>FAST_TARGET_POS_CALC<br>BEVEL_ANGLE_TOOL_LENGTH_CORR<br>BEVEL_ANGLE_LATERAL_OFFSET |
| Dimension | ---- |
| Default value | NONE |
| Remarks | Parameter available as of V3.1.3080. |

## 1.31.15 Velocity limit of the TCP

### 1.31.15.1 Activate velocity limit (see P-CHAN-00464)

| **P-CHAN-00464** | **Activate velocity limit** |
|---|---|
| Description | When this parameter is set, the specified kinematic is used to calculate the TCP. When the limit is switched on via tcp_velocity_limit control unit , the limit is applied. |
| Parameter | limit.kin[i].active |
| Data type | BOOLEAN |
| Data range | 0: No calculation<br>1: The corresponding velocity of the limit is calculated. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter available as of V3.1.3079.26 |

### 1.31.15.2 Name of the kinematic (P-CHAN-00465)

| P-CHAN-00465 | Name of the kinematic |
|---|---|
| Description | Name of the kinematic used to calculate the velocity. The kinematic must be defined by the same name in kin_step[0]. |
| Parameter | limit.kin[i].name |
| Data type | STRING |
| Data range | |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string.<br>Parameter available as of V3.1.3079.26<br>When the parameter is activated, a name must be specified; otherwise the error message ID 22108 is output. |

### 1.31.15.3 Velocity limit (P-CHAN-00466)

| P-CHAN-00466 | Velocity limit |
|---|---|
| Description | The maximum velocity that may be moved for this kinematic system when the function is active. |
| Parameter | limit.kin[i].velocity.max |
| Data type | REAL64 |
| Data range | > 0.001 µm/s |
| Dimension | µm/s |
| Default value | 0 |
| Remarks | Parameter available as of V3.1.3079.26 |

### 1.31.15.4 Mode with tool head offsets (P-CHAN-00469)

| P-CHAN-00469 | Mode for the use of tool head offsets |
|---|---|
| Description | This parameter defines the strategy with which tool parameters are used for the calculating kinematics. |
| Parameter | limit.kin[i].mode |
| Data type | UNS32 |
| Data range | 0: If the T/D command is output after a tool change, the tool length and the tool head offsets of the kinematics are updated.<br>1: In this mode, the kinematic parameters are **not** updated with a T/D word. In addition, the length or tool head offsets of the active tool are not added to the kinematic data of the kinematic at any time. |
| Dimension | --- |
| Default value | 0 |
| Remarks | Mode = 1 should be used if tool parameters have no relevance for the limit function, they are not compatible between the kinematics or the tool is parameterised differently for the limit.<br>Parameter available as of V3.1.3079.26 |

### 1.31.15.5 Velocity percentage of the interpolation with G201 (P-CHAN-00478)

| P-CHAN-00478 | Velocity percentage of the interpolation with G201 |
|---|---|
| Description | The parameter defines the velocity percentage [in %] of the interpolation in the velocity P-CHAN-00466 when G201 is active. |

| | |
|---|---|
| Parameter | limit.kin[i].velocity.ipo_weight_factor |
| Data type | UNS16 |
| Data range | |
| Dimension | 25 <= ipo_weight_factor <= 75 |
| Default value | 70 |
| Remarks | Manual mode contains the usual part of 100%; the default value is 70, i.e. 30.<br>Parameter available as of V3.1.3079.26 |

## 1.32 Default settings at program start (prog_start.*)

The elements of this structure can initialise G, M, F and additional functions at program start. The functions are therefore grouped according to their functionality.

If no values are specified for a group in the channel parameter list, the default settings are used.

### 1.32.1 Initialisation of G functions (prog_start.g_gruppe[i].*)

The G functions were divided into groups according to their functionality to execute change logging of the NC kernel (aep.*). This grouping is also used to initialise the G functions.

Initially, only settings for certain G groups are possible (see Section. G function numbers [▶ 191]).

| **Structure name** | **Index** |
|---|---|
| g_gruppe[i] | i = 0 ... 37 (maximum number of G function groups: 38) |

#### 1.32.1.1 G function numbers (P-CHAN-00063)

| **P-CHAN-00063** | **Default G function numbers at program start** |
|---|---|
| Description | This element defines the default G function for the G function groups. |
| Parameter | prog_start.g_gruppe[i].nr |
| Data type | SGN16 |
| Data range | -1 ≤ nr ≤ MAX(SGN16) |
| Dimension | ---- |
| Default value | -1 |

| Remarks | Group | Permissible default G functions |
|---|---|---|
| | g_gruppe[0].nr | 0, 1, 2, 3, 4 (G00, G01, G02, G03, G04) |
| | g_gruppe[4].nr | 17, 18, 19 (G17, G18, G19) |
| | g_gruppe[6].nr | 25, 26 (G25, G26) |
| | g_gruppe[8].nr | 51, 52 (G51, G52) |
| | g_gruppe[12].nr | 70, 71 (G70, G71) |
| | g_gruppe[14].nr | 90, 91 (G90, G91) |
| | g_gruppe[21].nr | 135, 137 (G135, G137) |
| | g_gruppe[22].nr | 138, 139, 237 - 239 (236) (G138, G139, G237 - G239) (G236) |
| | g_gruppe[23].nr | 161, 162 (G161, G162) |
| | g_gruppe[25].nr | 164, 165 (G164, G165) |
| | g_gruppe[32].nr | 140, 141 (G140, G141) |

Parameterisation example: At program start, the permissible G groups are initialised as follows.

```
prog_start.g_gruppe[0].nr 0 (range: WEG_BED)
prog_start.g_gruppe[4].nr 17 (range: EBENE)
prog_start.g_gruppe[6].nr 25 (range: WRK_UEBER)
prog_start.g_gruppe[8].nr 52 (range: DURCHM)
prog_start.g_gruppe[12].nr 71 (range: MASSEINH)
prog_start.g_gruppe[14].nr 90 (range: MASSANG)
prog_start.g_gruppe[21].nr 135 (range: VORST)
prog_start.g_gruppe[22].nr 139 (range: WRK_E_A)
prog_start.g_gruppe[23].nr 162 (range: MPKT_A_R)
prog_start.g_gruppe[25].nr 165 (range: MPKT_KORR)
prog_start.g_gruppe[32].nr 140 (range: CONT_MASK)
```

## 1.32.2 Initialisation of M functions (prog_start.m_gruppe[i].*)

M functions are classified into groups. However, for the moment, there is only the group of spindle M functions.

For the moment, only settings are possible for the default spindle M function (m_gruppe[0]).

| Structure name | Index |
|---|---|
| m_gruppe[i] | i = 0 ... 1 (maximum number of M function groups: 2) |

### 1.32.2.1 M function number (P-CHAN-00064)

| **P-CHAN-00064** | **Default M function numbers at program start** | |
|---|---|---|
| Description | This element sets the default M function for each M function group. | |
| Parameter | prog_start.m_gruppe[i].nr | |
| Data type | SGN16 | |
| Data range | -1 ≤ nr ≤ MAX(SGN16) | |
| Dimension | ---- | |
| Default value | -1 | |
| Remarks | Group | Permissible default M functions |
| | m_gruppe[0].nr | 3, 4, 5 (M03, M04, M05) |
| | Parameterisation example: At program start, the spindle M function 'spindle stop' (M05) is active.<br>*prog_start.m_gruppe[0].nr 5 (range:: SPDL_M_FCT)* | |

## 1.32.3 Initialisation of slope parameters (prog_start.slope.*)

The elements in this structure initialise the default values of slope parameters. At program start, these parameters are sent to the look ahead function.

### 1.32.3.1 Acceleration profile (P-CHAN-00071)

| **P-CHAN-00071** | **Default acceleration profile at program start** |
|---|---|
| Description | Default value for the selected acceleration profile type with path movements and oscillating axis movements. |
| Parameter | prog_start.slope.profile |
| Data type | SGN16 |
| Data range | 0: Step-shaped acceleration profile (default)<br>1: Trapezoidal acceleration profile<br>2: Sine-square acceleration profile<br>3: Trapezoidal HSC acceleration profile (across blocks) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | The acceleration profile and the associated acceleration and ramp time weighting can be programmed in the NC program with the #SLOPE [TYPE..] command [PROG].<br>The maximum of the weighted ramp times P-AXIS-00195 is always effective with the trapezoidal HSC acceleration profile. P-AXIS-00198.<br>For every independent axis, the acceleration profile can also be programmed specifically in the NC command with the key word SLOPE_TYPE [PROG]. |

### 1.32.3.2 Effect of ramp time weighting (P-CHAN-00073)

| **P-CHAN-00073** | **Default effect of ramp time weighting at program start** |
|---|---|
| Description | Default value for setting ramp time weighting. This value is only relevant for trapezoidal or sine-square acceleration profiles. |
| Parameter | prog_start.slope.ramp_time |
| Data type | SGN16 |
| Data range | 0: Weighting affects all ramp times (default)<br>1: Weighting acts on $T_{R,accel,inc}$<br>2: Weighting acts on $T_{R,accel,red}$<br>3: Weighting acts on $T_{R,decel,inc}$<br>4: Weighting acts on $T_{R,decel,red}$ |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.32.3.3 Effect of acceleration weighting (P-CHAN-00001)

| **P-CHAN-00001** | **Default effect of acceleration weighting at program start** |
|---|---|
| Description | Default value to set acceleration weighting. |
| Parameter | prog_start.slope.acceleration |
| Data type | SGN16 |
| Data range | 0: Weighting acts on all accelerations (default)<br>1: Weighting acts on $a_{accel}$<br>2: Weighting acts on $a_{decel}$ |

| | |
|---|---|
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.32.3.4 Profile calculation mode (P-CHAN-00349)

| **P-CHAN-00349** | **Profile calculation mode** |
|---|---|
| Description | Default value for the profile calculation mode. Acts on active jerk-limited acceleration profiles 1 and 2 (see P-CHAN-00071 [▶ 194]) and acceleration and deceleration processes on the path. Parameter has no effect on rapid traverse blocks (G00). |
| Parameter | prog_start.slope.mode_ramp_time |
| Data type | BOOLEAN |
| Data range | 0: Constant jerk, time-optimised (default)<br>1: Constant ramp time, not time-optimised but lower excitation on the path |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is available as of CNC Build V2.11.2030.08 and higher. |

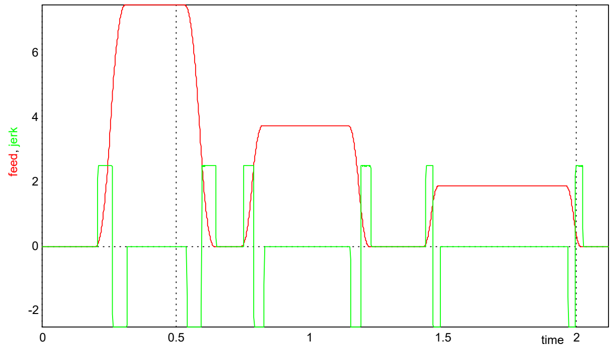


Fig. 36: const_ramp_time = 0

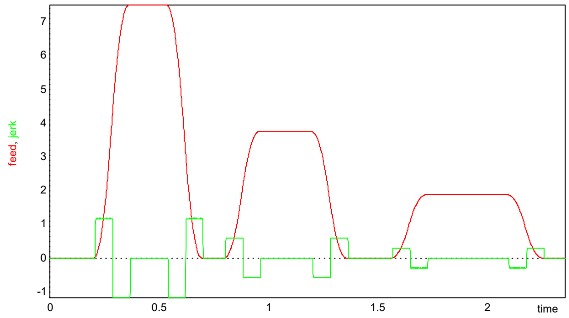


Fig. 37: const_ramp_time = 1

## 1.32.4 Initialisation of default feed rate (P-CHAN-00099)

| **P-CHAN-00099** | **Default feed rate at program start** |
|---|---|
| Description | Default value for setting the F word when programming a feed rate block. |
| Parameter | prog_start.feedrate |
| Data type | REAL64 |
| Data range | ---- |
| Dimension | mm/min |
| Default value | 0.0 |

| | |
|---|---|
| Remarks | Parameterisation example: When a feed block (G01/G02/G03) is programmed, the velocity should be 1000 mm/min if the F word is missing.<br>*prog_start.feedrate 1000* |

### 1.32.5 Initialisation of feed rate unit (P-CHAN-00108)

| **P-CHAN-00108** | **Default feed rate unit at program start** |
|---|---|
| Description | This parameter defines the unit used to interpret the feed rate value (e.g. F1000). |
| Parameter | prog_start.feedrate_factor |
| Data type | REAL64 |
| Data range | 0.1: Unit of the F word is m/min<br>100: Unit of the F word is mm/min<br>≤ 0 : Warning output. Internal default assignment with value 100 (mm/min).<br>Exception:<br>When the value is -1 or if the parameter is not in the list, no warning is generated. Also internal default assignment with value of 100 (mm/min). |
| Dimension | ---- |
| Default value | 100 |
| Remarks | Parameterisation example: The programmed feed rate is interpreted in the unit of mm/min.<br>*prog_start.feedrate_factor 100* |

### 1.32.6 Late synchronisation at program end (P-CHAN-00033)

| **P-CHAN-00033** | **Default setting for 'Late synchronization at program end' at program start** |
|---|---|
| Description | This parameter is used in conjunction with synchronisation types MVS_SLM and MVS_SLP (Late Sync). It defines the reaction to open Late Sync M functions at program end. This may occur if there is no G01 block in the NC program until program end (with MVS_SLM) or if no explicit synchronisation was programmed (with MVS_SLP). |
| Parameter | prog_start.late_sync_ready |
| Data type | BOOLEAN |
| Data range | 0: Open Late Sync M functions at the end of the program are still active the next time the program is started. This means that one or several Late Sync M functions of the first NC program are only triggered by a Late Sync event (G01 block or #EXPL SYN) of the second NC program.<br>1: At program end, the program waits until all open Late Sync M functions are acknowledged by the PLC. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example: At program end, the program waits until all open Late Sync M functions are acknowledged by the PLC.<br>*prog_start.late_sync_ready 1* |

### 1.32.7 Settings for Job Manager mode

The parameters described in this chapter define the default configuration of various operation modes for Job Manager mode. These modes can be redefined on program start by the PLCopen module MC_MovePath or by the #START command in the NC program. The changes are only valid for the ordered program.

### 1.32.7.1 Requesting current positions at program start (P-CHAN-00316)

| P-CHAN-00316 | Job manager mode: Request current positions at program start |
|---|---|
| Description | This parameter controls the processing modes for requesting current positions at program start. |
| Parameter | prog_start.initialize_on_actual_position |
| Data type | BOOLEAN |
| Data range | 0: No request of current positions at program start.<br>1: Request current positions at program start. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.32.7.2 Initialisation of working data at program start (P-CHAN-00317)

| P-CHAN-00317 | Job manager mode: Initialise working data at program start |
|---|---|
| Description | This parameter controls the processing modes for initialising working data at program start. |
| Parameter | prog_start.set_default_config |
| Data type | BOOLEAN |
| Data range | 0: When the program is started, no initialisations or changes of previous NC programs remain active.<br>1: When the program is started, all settings are initialised and assigned with default values. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.32.7.3 Enabling the interface to log scene data (P-CHAN-00318)

| P-CHAN-00318 | Job manager mode: Enable the interface to log scene data |
|---|---|
| Description | This parameter controls the enabling of the interface to log scene data. |
| Parameter | prog_start.report_scene_sample |
| Data type | BOOLEAN |
| Data range | 0: No logging of scene data.<br>1: Enable the interface. Scene data is logged. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.32.7.4 Enabling the interface to log time stamps (P-CHAN-00319)

| P-CHAN-00319 | Job manager mode: Enable the interface to log time stamps |
|---|---|
| Description | This parameter controls the enabling of the interface to log time stamps. |
| Parameter | prog_start.report_run_time_measure |
| Data type | BOOLEAN |
| Data range | 0: No logging of time stamps.<br>1: Enable the interface. Time stamps are logged. |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | |

### 1.32.7.5 Enabling the interface to log axis positions (P-CHAN-00320)

| **P-CHAN-00320** | **Job manager mode: Enable the interface to log axis positions** |
|---|---|
| Description | This parameter controls the enabling of the interface to log axis positions. |
| Parameter | prog_start.report_axes_position_sample |
| Data type | BOOLEAN |
| Data range | 0: No logging of axis positions.<br>1: Enable the interface. Axis positions are logged. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.33 Reduction of tangential transition velocity (P-CHAN-00009)

| **P-CHAN-00009** | **Reduction of tangential transition velocity between circles** |
|---|---|
| Description | Tangential block transitions between circles with different radii and circles and linear blocks and vice versa lead to a jerk depending on circle radius. A jerk produced by activating this function can be reduced in order to reduce the excitation of vibrations on the machine when non-linear velocity profiles are active. At tangential block transitions, the speed is reduced dependent on the permissible jerk. The calculation is based on axis-specific jerk parameters for non-linear speed profiles (see also [AXIS] documentation).<br>On the other hand, the reduction of speed at tangential block transitions is not acceptable with specific machining technologies since the machining process has a very sensitive reaction to a reduction in velocity. |
| Parameter | corr_v_trans_jerk |
| Data type | BOOLEAN |
| Data range | 0: No inclusion of jerk at tangential block transitions.<br>1: Inclusion of jerk at tangential block transitions. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.34 Including jerk in the polynomial (P-CHAN-00110)

| **P-CHAN-00110** | **Including jerk in the polynomial** |
|---|---|
| Description | The curvature of the programmed contour (polynomial) results in an axis jerk. If the value is 1 or 3, this jerk is monitored together with the axis-specific dynamic parameters of the geometrical ramp time (P-AXIS-00199).<br>In addition, an active kinematic compensation movement is included when value 3 is in the polynomial.<br>If the value is 2, the ramp times P-AXIS-00195… P-AXIS-00198 are used for jerk monitoring.<br>If a relatively small maximum jerk is specified in the axis, the velocity on the path is reduced accordingly. If this reduction is not wanted for technological reasons, these parameters suppress jerk monitoring in polynomials. |
| Parameter | check_jerk_on_poly_path |
| Data type | UNS32 |

| | |
|---|---|
| Data range | 0: No jerk limiting in the polynomial.<br>1: Jerk limiting in the polynomial based on P-AXIS-00199 (default).<br>2: Jerk limiting in the polynomial based on the maximum of P-AXIS-00195, P-AXIS-00196, P-AXIS-00197, P-AXIS-00198.<br>3: As for value 1 and in addition polynomials in combination with kinematic compensation movement. |
| Dimension | ---- |
| Default value | 1 |
| Remarks | |

## 1.35 Jerk limiting mode at block transition (P-CHAN-00117)

| **P-CHAN-00117** | **Jerk limiting mode at block transition of any blocks** |
|---|---|
| Description | This parameter allows the selection of several modes for parameterising jerk limiting at any block transition. For example, jerk is adjustable either by weighting the acceleration or directly by specifying maximum jerk values. |
| Parameter | mode_trans_jerk |
| Data type | UNS32 |
| Data range | 0: Weighting parameter P-AXIS-00013 is dependent on P-AXIS-00154, i.e. P-AXIS-00013 may not be set smaller than P-AXIS-00154.<br>1: Weighting parameters P-AXIS-00154 and P-AXIS-00013 can be set dependent on each other.<br>2: Jerk parameters P-AXIS-00339 and P-AXIS-00340 are effective and can be set independent of each other. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.36 Additional consideration of curvature at block transition (P-CHAN-00245)

| **P-CHAN-00245** | **Additional consideration of curvature at block transition between linear blocks** |
|---|---|
| Description | This parameter also considers the approximated curvature at block transition of two linear blocks and from this calculates a maximum transition velocity. |
| Parameter | trans_limit_with_curvature |
| Data type | UNS32 |
| Data range | 0: Not active.<br>1: Additional limit block transition velocity due to a curvature<br>This mode is recommended when Slope Profile 3 (HSC velocity planning) is used. (Default).<br>2: Additional block transition velocity limit due to a curvature P-AXIS-00199 (tr_geom). |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.37 Look ahead speed limit (speed_limit_look_ahead.*)

Depending on the machining technology used, it may be necessary for the controller to inform the PLC about changes in velocity profile. In this case, if velocity falls below the speed limit, the CNC generates a signal 'speed limit detected' (SLD).

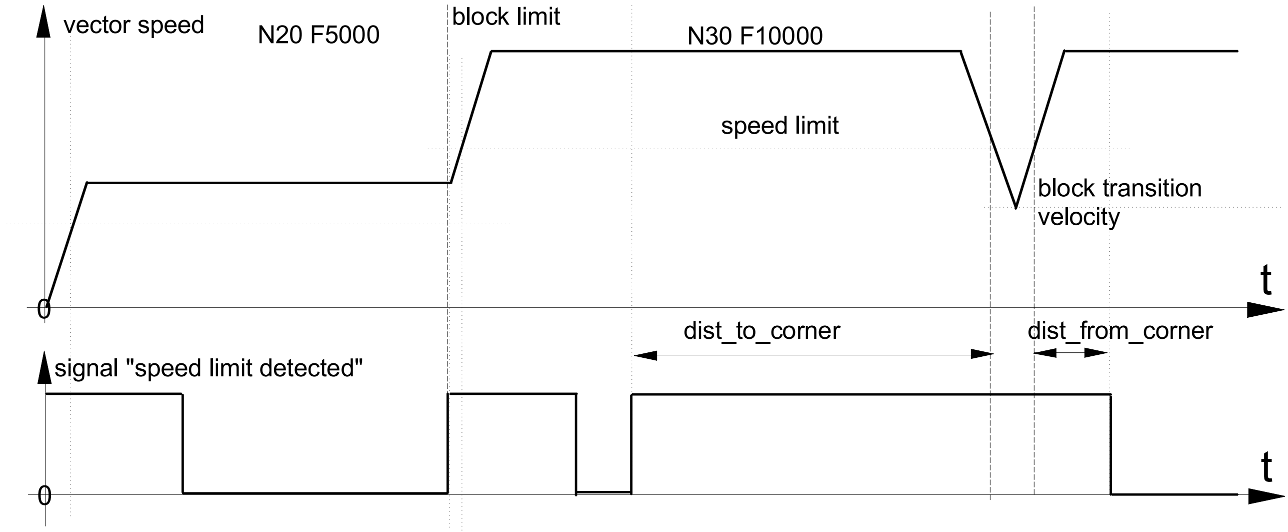


Fig. 38: Timing example of speed limit look ahead

```
speed_limit_look_ahead.enable                      1 0: inactive, 1: active
speed_limit_look_ahead.v_limit 750 [0.1%] of programmed speed
speed_limit_look_ahead.time 1 unit of time or distance
speed_limit_look_ahead.dist_to_corner 5000 [0.1 µm or µs]
speed_limit_look_ahead.dist_from_corner 6000 [0.1 µm or µs]
speed_limit_look_ahead.override_weight_v_limit  0 0: inactive, 1: active
```

### 1.37.1 Enable / disable (P-CHAN-00017)

| **P-CHAN-00017** | **Enable/disable speed limit look ahead** |
|---|---|
| Description | Parameter to enable or disable the speed limit look ahead function. |
| Parameter | speed_limit_look_ahead.enable |
| Data type | BOOLEAN |
| Data range | 0: Speed limit look ahead is disabled.<br>1: Speed limit look ahead is enabled. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | *speed_limit_look_ahead.f_enable (old syntax up to V2.11.2022.13)* |

### 1.37.2 Weighting of speed limit (P-CHAN-00089)

| **P-CHAN-00089** | **Weighting of speed limit for speed limit look ahead** |
|---|---|
| Description | Speed limit value in 0.1 percent of programmed speed. If current speed falls below the limit<br>v = $v_{prog}$*v_limit/1000,<br>the logical signal SLD 0 ->1 is generated. |
| Parameter | speed_limit_look_ahead.v_limit |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | 0.1% |
| Default value | 0 |

| | |
|---|---|
| Remarks | |

## 1.37.3 Unit (P-CHAN-00018)

| P-CHAN-00018 | Unit to interpret the SLD signal for look-ahead speed |
|---|---|
| Description | The logical signal SLD is generated depending on the parameter values for distance or time. |
| Parameter | speed_limit_look_ahead.time |
| Data type | BOOLEAN |
| Data range | 0: The distance parameters P-CHAN-00012 [▶ 201] and P-CHAN-00013 [▶ 201] are interpreted as path.<br>1: The parameters P-CHAN-00012 [▶ 201] and P-CHAN-00013 [▶ 201] are interpreted as time. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | *speed_limit_look_ahead.f_time (old syntax up to V2.11.2022.13)* |

## 1.37.4 Distance to corner (P-CHAN-00013)

| P-CHAN-00013 | Distance to corner for speed limit look ahead |
|---|---|
| Description | The logical signal SLD 0 ->1 is generated in advance depending on the parameters 'distance to corner' or 'time to corner'. Here, corner means the position in the block at which the speed drops below the speed limit. |
| Parameter | speed_limit_look_ahead.dist_to_corner |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | 0.1µm or µs |
| Default value | 0 |
| Remarks | |

## 1.37.5 Distance from corner (P-CHAN-00012)

| P-CHAN-00012 | Distance from corner for speed limit look ahead |
|---|---|
| Description | The logical signal SLD 1 ->0 is generated depending on the parameters 'distance from corner' or 'time from corner'. Here, corner means the position in the block at which the speed rises again above the speed limit. |
| Parameter | speed_limit_look_ahead.dist_from_corner |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | 0.1µm or µs |
| Default value | 0 |
| Remarks | |

## 1.37.6 Weighting the speed limit via override (P-CHAN-00155)

| P-CHAN-00155 | Weighting the speed limit via override for Speed limit Look Ahead function |
|---|---|
| Description | This parameter controls the influence of the speed limit via the real-time feed override. In the default setting, the real-time feed override does not influence the speed limit P-CHAN-00089 [▶ 200] (v_limit). However, if this is desirable e.g. to commission or enter contours, the parameter is set to 1. Then the parametrised speed limit value is weighted |

| | |
|---|---|
| | by the override value. Note that at a non-constant programmed feed, the SLD signal is activated in each acceleration phase because the speed limit at the start of the block is set to the new value. |
| Parameter | speed_limit_look_ahead.override_weight_v_limit |
| Data type | BOOLEAN |
| Data range | 0: No weighting of P-CHAN-00089 [▶ 200] (default).<br>1: Weighting of P-CHAN-00089 [▶ 200] via override. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | *f_override_weight_v_limit (old syntax up to V2.11.2022.13)* |

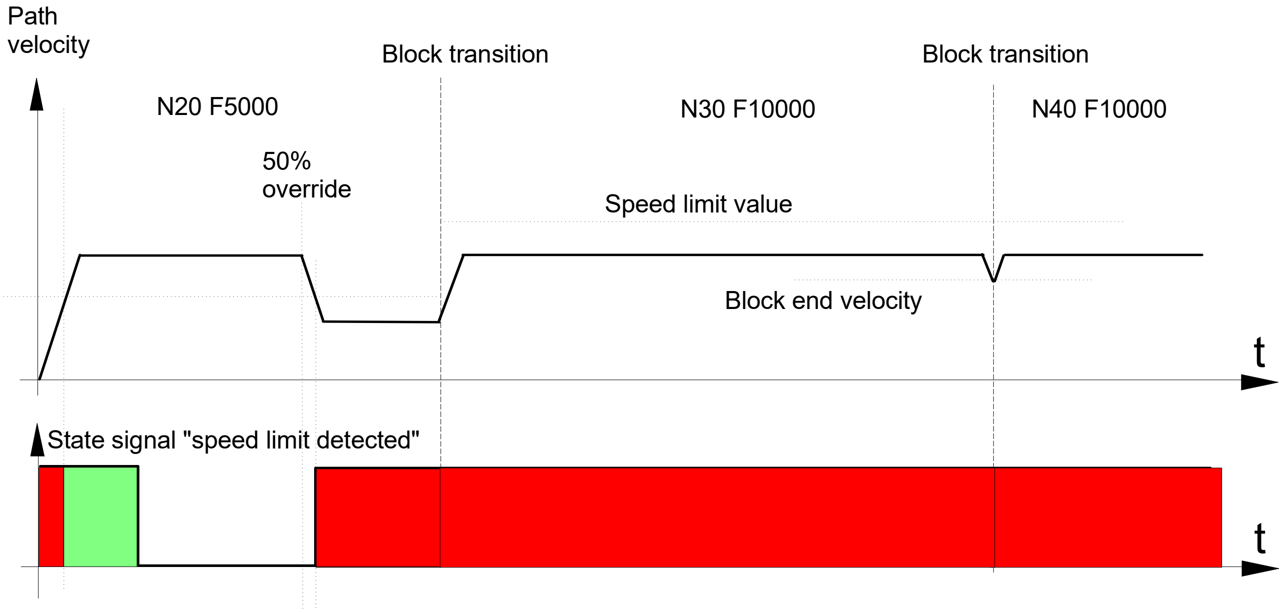


Fig. 39: Timing diagram without override weighting (f_override_weight_v_limit = 0)

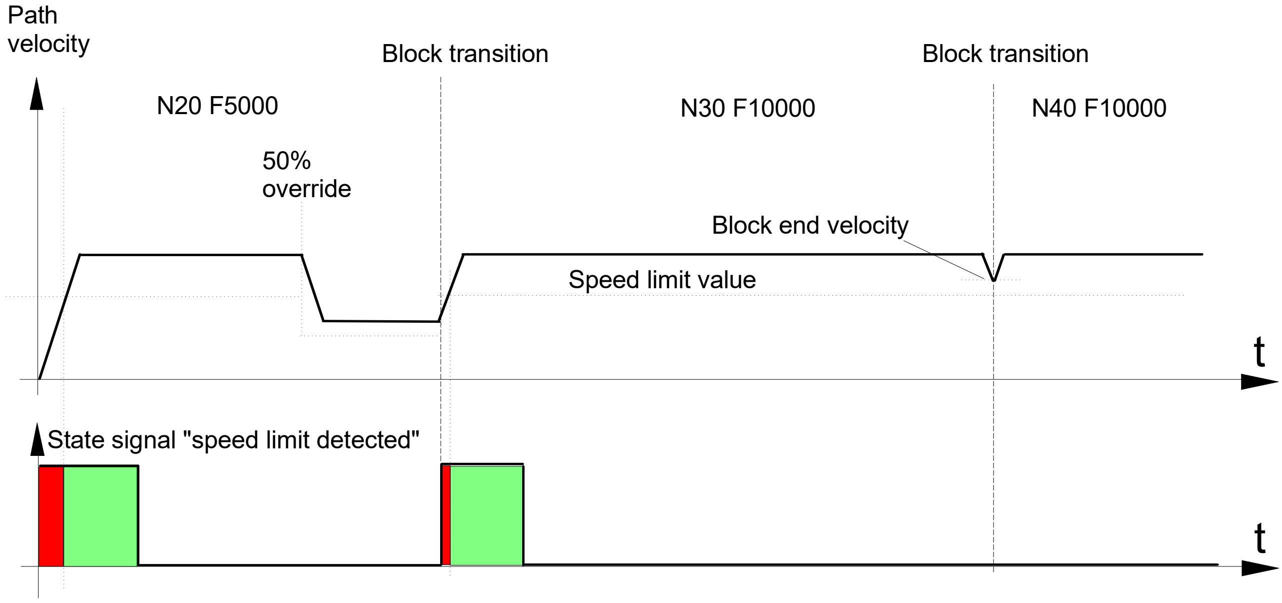


Fig. 40: Timing diagram with override weighting (f_override_weight_v_limit = 1)

# 1.38 Settings for manual mode

## 1.38.1 Rejection of manual mode path (P-CHAN-00113)

| **P-CHAN-00113** | **Reject manual mode path at program end** |
|---|---|
| Description | Active manual mode G201 can be deselected either by G202 or implicitly at program end. The G200 mode can only be deactivated by means a change of operation mode or by 'Continue of processing' in automatic mode. Depending on manual operation mode (incremental jog, continues jog) and the active real-time influences (e.g. feed override), a residual path can still be active at the time of deactivation of manual mode. This residual path can cause an unexpected overtravel of the axis.<br>If this is not required, this parameter rejects the residual path. The movement then stops immediately when manual mode is deactivated. |
| Parameter | path_reject_std_manual_mode |
| Data type | BOOLEAN |
| Data range | 0: Active residual manual mode path is completely executed after deactivation.<br>1: Rejection of active residual manual mode at deactivation. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.38.2 Relative manual mode offset limits with G200 (P-CHAN-00114)

| **P-CHAN-00114** | **Relative manual mode offset limits with G200** |
|---|---|
| Description | With active G200, travel limitation is controlled by predefined software limit switches if homing was done before.<br>If the relative offset limits (#MANUAL LIMITS, P-AXIS-00137, P-AXIS-00138) should also be taken into consideration, this can be controlled by setting this parameter. |
| Parameter | rel_offset_limits_std_manual_mode |
| Data type | BOOLEAN |
| Data range | 0: No consideration of relative offset limits with manual mode G200.<br>1: Consideration of relative offset limits with manual mode G200. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.38.3 Override with manual mode (P-CHAN-00186)

| **P-CHAN-00186** | **Override with manual mode** |
|---|---|
| Description | The axis or path-specific override acts on the feed rate during incremental or continuous jog mode. In handwheel mode no feed rate override weighting is executed. The current feed rate results from the rotation speed of the handwheel, the parameter setting for resolution and increments per CNC cycle. The nominal feed rate is defined by the axis-specific manual mode parametrisation.<br>This parameter sets the effective override also for handwheel mode. |
| Parameter | override_v_handwheel |
| Data type | BOOLEAN |
| Data range | 0: No feed rate weighting by override with manual mode (default)<br>1: Feed rate weighting by override in handwheel mode. |
| Dimension | ---- |

| | |
|---|---|
| Default value | 0 |
| Remarks | |

### 1.38.4 Error message output at manual mode axis overrun (P-CHAN-00437)

| P-CHAN-00437 | Error message output after manual mode axis overrun |
|---|---|
| Description | This parameter helps to diagnose and suppress incorrect operation in manual mode. When the parameter is set, an error message is generated if manual mode axes still move when manual mode is disabled (deselected with ‘G202’ or ‘Continue motion’). |
| Parameter | err_outp_manual_mode_move |
| Data type | BOOLEAN |
| Data range | 0: Overrunning manual mode axis is braked after disable.<br>1: When an overrunning manual mode axis is disabled, it is stopped and an error message is output. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | In normal mode, manual mode axes can still move at the time of disable if they were commanded to do so shortly before by the operator. |

### 1.38.5 Suppression of workspace monitoring in manual operation (P-CHAN-00442)

| P-CHAN-00442 | Suppression of workspace monitoring function in manual operation |
|---|---|
| Description | This parameter influences the workspace monitoring function in manual mode.<br>As from version V3.3070.11 onwards, activating the workspace monitoring function using the NC program also simultaneously activates the workspace monitoring function in manual mode.<br>0: Workspace monitoring is activated as described above<br>1: Workspace monitoring is not activated<br>2: Workspace monitoring is activated; error handling is suppressed when manual mode is selected for axes located outside/within the workspace/protection area. |
| Parameter | suppress_workspace_monitoring_manual_mode |
| Data type | UNS16 |
| Data range | 0, 1, 2 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | The parameter is available as of Build V3.3070.11. |

## 1.39 Exclusivity of measuring system programming (P-CHAN-00116)

| P-CHAN-00116 | Exclusivity of measuring system programming |
|---|---|
| Description | This parameter disables the exclusivity of G90/G91 programming (absolute/relative measuring system) in the same NC block ([PROG//section Measuring systems (G90/G91)]. |
| Parameter | multi_dimension_in_block |
| Data type | BOOLEAN |
| Data range | 0: G90/G91 may only be programmed exclusively in the same NC block (default).<br>1: G90/G91 may be programmed at the same time in the same NC block. The measuring system last programmed applies to all following positions up to the next G90/G91. |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | Programming example:<br>`Nxx G90 X10 Y20 G91 Z30 -> X,Y absolute, Z relative`<br>`Nxx X15 G90 Y25 Z15 -> X relative, Y,Z absolute`<br>`Nxx G91 X10 G90 Z30 G91 Y5 -> X relative, Z absolute, Y relative`<br>`Nxx X20 Z30 G90 Y40 -> X,Z relative, Y absolute` |

## 1.40 Exclusivity of axes programming (P-CHAN-00148)

| **P-CHAN-00148** | **Exclusivity of axis programming** |
|---|---|
| Description | This parameter disables the exclusivity of axis programming. This parameter is only evaluated if multiple programming of the measuring system is permitted (P-CHAN-00116 [▶ 205]).<br>In this case the multiple programming of identical axes in different dimensions (G90/G91) is permitted in the same NC block. |
| Parameter | multi_axes_in_block |
| Data type | BOOLEAN |
| Data range | 0: Axes may only be programmed exclusively in the same NC block (default).<br>1: The same axes may be programmed several times in the same NC block. If an axis in the block was programmed as absolute and relative, the resulting position is output as an absolute value. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Programming example:<br>`Nxx G90 X10 G91 Y10 Z20 Y1 -> Y relative about 11`<br>`Nxx G91 X10 Y10 G90 X2 G91 Z30 -> X absolute to 2`<br>`Nxx G90 X10 Y20 G91 Z30 Y5 -> Y absolute to 25`<br>`Nxx G91 Z15 Y20 G90 X5 G91 Z30 -> Z relative about 45` |

## 1.41 Subroutine call with M6 (P-CHAN-00118)

| **P-CHAN-00118** | **Subroutine call with M6** |
|---|---|
| Description | This parameter defines the name of a global subroutine which is called implicitly when M6 is detected in the NC program.<br>In this case, M6 is not treated as a normal M function but becomes a global subroutine call analogous to 'L ...' This prevents a technological process from being triggered. M6 is always executed as the last action at block end.<br>If the parameter is not assigned or is not present in the channel parameter list, an M6 is processed as a normal M function.<br>In the NC program, the subroutine name can also be set by the command #FILE NAME [M6=“<prog_name>“]. The new subroutine name is then valid until program end M30 or until a further #FILENAME [M6=”<prog_name>”] [PROG].<br>Then after RESET or program start, the entry of the parameter in the channel parameters list is valid again. |
| Parameter | m6_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameterisation example:<br>*m6_prog_file tool_change1*<br>* Note: The default value of variables is a blank string. |

## 1.42 Subroutine call with D word (P-CHAN-00429)

| **P-CHAN-00429** | **Subroutine call with D word** |
|---|---|
| Description | This parameter defines the name of a global subroutine which is called implicitly if a D word is detected in the NC program. A separate subroutine can therefore be specified for each NC channel.<br>The global subroutine call is always executed as the last action at block end. This subroutine executes tool-specific settings.<br>If a D word is implicitly executed after a T selection (P-CHAN-00014 [▸ 94]: einrechnen_mit_t=1), the subroutine is also called by a T word.<br>In the NC program, the subroutine name can also be set by the command #FILE NAME [D="<prog_name>"]. The new subroutine name is then valid until program end M30 or until a further #FILENAME [D="<prog_name>"] [PROG].<br>If the parameter is not assigned or is not present in the channel parameter list, a D word is executed without subroutine call.<br>Then after RESET or program start, the entry of the parameter in the channel parameters list is again valid. |
| Parameter | d_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | This parameter is available as of CNC Build V3.1.3066.03 and higher.<br><br>Parameterisation example:<br>*d_prog_file          tool_data1.nc*<br><br>Programming example:<br>*N..*<br>*N10 D127                              ;Call tool_data1.nc*<br>*N20 #FILE NAME [D="tool_data2.nc]       ;Define new subroutine*<br>*N30 D128                              ; Call tool_data2.nc*<br>*N..*<br>* Note: The default value of variables is a blank string. |

# 1.43 Subroutine call at program start / MDI

The basic assignment and initialisation of settings, e.g for display or other functions, can take place in a global subroutine which is called automatically at the start of NC main program start or MDI. The following two channel parameters are for intended to configure this procedure.

## 1.43.1 Subroutine name (P-CHAN-00119)

| **P-CHAN-00119** | **Name of implicit subroutine call at program start** |
|---|---|
| Description | If the name of a (global) subroutine is entered in this parameter, the subroutine is called implicitly as the first action at the normal start of the NC program start or MDI. In addition the execution at NC program start and / or MDI can be configured by the channel parameter P-CHAN-00260 [▸ 208] . |
| Parameter | start_init_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameterisation example:<br>*start_init_prog_file init.nc*<br>* Note: The default value of variables is a blank string.<br>The parameter P-CHAN-00252 [▸ 210] can call an implicitly called subroutine at program end. |

## 1.43.2 Mode of subroutine call (P-CHAN-00260)

| **P-CHAN-00260** | **Mode of implicit subroutine call at program start** |
|---|---|
| Description | This parameter defines the execution mode of the implicit subroutine call at the start of an NC program and/or MDI.<br>There is no distinction between MDI mode and manual mode. |
| Parameter | start_init_prog_file_mode |
| Data type | STRING |
| Data range | AUTOMATIC: Call a global subroutine by (P-CHAN-00119 [▸ 208]) at start of an NC main program.<br>If no ID is entered, the basic setting is AUTOMATIC.<br>MDI: Call a global subroutine by (P-CHAN-00119 [▸ 208]) at the start of an MDI. Here, the command sequence of a global subroutine call (L <filename>) is prefixed to the received MDI string.<br>Parameterisation example:<br>Received MDI: **G01 X100 Y150 F1000** \n<br>Executed MDI: **L init.nc** \n **G01 X100 Y150 F1000** \n |
| Dimension | ---- |
| Default value | AUTOMATIC |

| | |
|---|---|
| Remarks | Examples of permissible assignments:<br>`start_init_prog_file_mode      AUTOMATIC \| MDI`<br>`start_init_prog_file_mode      AUTOMATIC`<br>`start_init_prog_file_mode      MDI`<br>This parameter is available as of CNC Build V3.1.3021.00<br>Using the variable V.G.MAIN_FILE_NAME in the implicit subroutine, it can be determined whether the NC main program is executed in AUTOMATIC mode or MDI. With MDI, the variable always supplies the character “-” as filename.<br>This permits the programming of mode-specific program sequences. |

# 1.44 Subroutine call at program end / MDI

Resetting and re-initialising settings, e.g for display or other functions, can take place in a global subroutine which is called automatically at the start of an NC main program or MDI. The following two channel parameters are for intended to configure this procedure.

## 1.44.1 Subroutine name (P-CHAN-00252)

| **P-CHAN-00252** | **Name of implicitly called subroutine at program end** |
|---|---|
| Description | If this parameter contains the name of a (global) subroutine, it is executed implicitly at main program end or MDI **before** M02 or M30. The last action is then the execution of M02 or M30. In addition the execution at NC program end and / or MDI end can be configured by the channel parameter P-CHAN-00433 [▶ 211] . |
| Parameter | final_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameterisation example:<br>*final_prog_file final.nc*<br>* Note: The default value of variables is a blank string.<br>The parameter P-CHAN-00119 can call an implicitly called subroutine at program start. |

## 1.44.2 Mode of subroutine call (P-CHAN-00433)

| **P-CHAN-00433** | **Mode of implicit subroutine call at program end** |
|---|---|
| Description | This parameter defines the execution mode of the implicit subroutine call at the end of an NC program and/or MDI.<br>There is no distinction between MDI mode and manual mode. |
| Parameter | final_prog_file_mode |
| Data type | STRING |
| Data range | AUTOMATIC: Call a global subroutine by (P-CHAN-00252 [▸ 210]) at the end (M30) of an NC main program.<br>If no ID is entered, the basic setting is AUTOMATIC.<br>MDI: Call a global subroutine by (P-CHAN-00252 [▸ 210]) at the end of an MDI. Here, a global subroutine call (L <filename>) is appended to the command sequence of the received MDI character string or is inserted before an existing M30.<br>Parameterisation example 1:<br>Received MDI: **G01 X100 Y150 F1000** \n<br>Executed MDI: **G01 X100 Y150 F1000**\n **L final.nc** \n<br><br>Parameterisation example 2:<br>Received MDI: **G01 X100 Y150 F1000** \n **M30** \n<br>Executed MDI: **G01 X100 Y150 F1000** \n **L final.nc** \n **M30** \n |
| Dimension | ---- |
| Default value | AUTOMATIC |
| Remarks | Examples of permissible assignments:<br>`final_prog_file_mode     AUTOMATIC \| MDI`<br>`final_prog_file_mode     AUTOMATIC`<br>`final_prog_file_mode     MDI`<br>This parameter is available as of CNC Build V3.1.3068.01 |

# 1.45 Settings for software limit switch (SLS) monitoring

## 1.45.1 Software limit switch (SLS) monitoring with tolerance (P-CHAN-00120)

| **P-CHAN-00120** | **Software limit switch (SLS) monitoring with tolerance** |
|---|---|
| Description | Software limit switch monitoring (SLS monitoring) on the actual value side is always executed on the basis of P-AXIS-00177 and P-AXIS-00178 and taking the tolerance P-AXIS-00179 into consideration.<br>SLS monitoring on the command value side is executed in path preparation on the basis of P-AXIS-00177 and P-AXIS-00178 . At program start, if some axes are outside the software limit switch positions mentioned above, the CNC logs a software limit switch (SLS) error.<br>This parameter permits the user to activate SLS monitoring with tolerance; this also considers the axis parameter P-AXIS-00179 in path preparation. If the current axis position is outside the software limit switch positions but within the tolerance, no SLS error is generated. The tolerance is only considered here if the axis is not programmed or moved. |
| Parameter | soft_limit_tolerance |
| Data type | BOOLEAN |
| Data range | 0: Software limit switch monitoring without tolerance<br>1: Software limit switch monitoring with tolerance at stationary axes |

| | |
|---|---|
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.45.2 Error reaction for command value based software limit switch monitoring (P-CHAN-00147)

| **P-CHAN-00147** | **Error reaction for software limit switch monitoring on command value side** |
|---|---|
| Description | This parameter defines the error reaction for single channel axes for command value based software limit switch monitoring. This does not affect the error reaction of the actual value based monitoring. The function is a property of the channel axes. |
| Parameter | soft_limit_warning_axes |
| Data type | UNS32 |
| Data range | 0x0..0xFFFFFFFF |
| Dimension | ---- |
| Default value | 0 |
| Remarks | 0: When the command value exceeds the software limit switch, the axis motion stops and the channel remains in error state until CNC RESET (default).<br>!=0: Only a warning is generated when the command value exceeds the software limit switch for the defined axes; machining is continued.<br>Parameterisation example: Warning when the software limit switch is exceeded on the command value side for axes 1 to 3 of the channel.<br>*soft_limit_warning_axes     0x7* |

## 1.45.3 Suppress software limit switch monitoring on command value side (P-CHAN-00459)

| **P-CHAN-00459** | **Suppress the software limit switch monitor on the command value side** |
|---|---|
| Description | This parameter suppresses software limit switch monitoring for individual axes in the channel on the command value side.<br>The (cross-channel) monitoring mechanisms of axes in the position controller, such as the monitoring of software limit switches and axis collisions on the command value and actual value sides, are not affected.<br>If monitoring is suppressed for a single axis in the channel, the index for this axis in the channel must be set bit-coded in P-CHAN-00459.<br>The function is a property of the channel axes. |
| Parameter | suppress_soft_limit_monitoring_axes |
| Data type | UNS32 |
| Data range | 0x0..0xFFFFFFFF |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | 0: SLS monitoring is active for all axes in the channel; active SLS'es limit the offset limits in manual mode (default).<br>!=0: The following applies to the selected axes in the channel:<br>• The SLS monitor is inactive.<br>• Offset limit monitoring in manual mode remains active; active SLS'es do not affect manual mode offset limits.<br>• The monitoring mechanisms of assigned axes in the position controller remain active.<br><br>Parameterisation example: SLS monitoring of axes 1 to 3 of the channel inactive.<br>*suppress_soft_limit_monitoring_axes 0x7* |

**Application example - Deactivate SLS monitoring**

With an axis coupling via PLC, source axes are used and operated in coupling mode 1(HLI_AXIS_COUPLING_ZERO).

If the software limit switch monitor for these axes is suppressed by P-CHAN-00459 [▸ 212] , the monitoring of software limit switches and axis collisions on the actual value sides stored in the axis parameters remain active for each axis in the position controller.

For these axes, the manual mode offset limits to be monitored are specified by CNC objects (manual cmd.abs.limit-, manual cmd.abs.limit+). These manual mode offset limit settings may also be outside the parameterised software limit switches (P-AXIS-00177/ P-AXIS-00178) and consequently limit only the motion path of source axes.

## 1.45.4 Limitation of software limit switches in manual mode with parallel interpolation (P-CHAN-00759)

| P-CHAN-00759 | Software limit switch limits in manual mode with parallel interpolation (G201) |
|---|---|
| Description | This parameter activates the limitation of axis motion path ranges in manual mode with parallel interpolation G201 to the axis software limit switches analogous to the behaviour with exclusive manual mode (G200)<br>The limitation of software limit switches has an additive effect to the relative offset limits of manual mode P-AXIS-00137, P-AXIS-00138 i.e. the end of the motion path range is the minimum resulting from the relative offset limits and the distance to the software limit switches of the axes.<br>The limitation refers exclusively to path motions in manual mode. If the axis is also moved along the path, e.g. across the NC program, error messages for software limit switches in manual mode may continue to occur. |
| Parameter | swe_limits_additive_manual_mode |
| Data type | BOOLEAN |
| Data range | 0 / 1 |
| Dimension | --- |
| Default value | 0 |
| Remarks | This parameter is available as of CNC Build V3.1.3079.39 |

## 1.45.5 Behaviour with programmed software limit switches (P-CHAN-00498)

| P-CHAN-00498 | Behaviour with programmed software limit switches |
|---|---|

| | |
|---|---|
| Description | This parameter defines the limitation behaviour for programming software limits switches in the NC program using G98 and G99.<br>As of CNC Build V3.1.3077.0 it is permitted to increase the range of the software limits switches in the NC program.<br>If this behaviour is not required, the limitation can be activated to the parameterised software limit switches of the corresponding axis parameter list:. (P-AXIS-00177 and P-AXIS-00178)<br>The programmed values for the software limit switches are then limited to the values in the corresponding axis parameter list and the warning ID 21648 or ID 21650 is output.<br>For further information on software limit switches: Extensions to G98 and G99 |
| Parameter | range_check_prog_soft_limits |
| Data type | BOOLEAN |
| Data range | 0: No limitation of the programmed values<br>1: Limitation of the programmed values to the range of the software limit switches |
| Dimension | --- |
| Default value | 0 |
| Remarks | Parameter available as of V3.1.3080.4 |

## 1.46 List interpretation with active NC program (P-CHAN-00146)

| **P-CHAN-00146** | **Adopt configuration lists when NC program is active** |
|---|---|
| Description | Normally, configuration lists may not be adopted when an NC program is active.<br>This parameter can cancel this blocking. The user must then make sure that channel buffering is cancelled at the time when a new list is adopted by suitable programmed (e.g. #FLUSH WAIT in connection with an M function).<br>This allows the adoption of zero offsets, position offsets, tool data, external variables and channel parameters when an NC program is active. |
| Parameter | parameter_change_during_execution |
| Data type | BOOLEAN |
| Data range | 0: List interpretation blocked when an NC program is active (default).<br>1: List interpretation allowed when an NC program is active. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.47 Adoption of block number when subroutine is called (P-CHAN-00150)

| **P-CHAN-00150** | **Adopt block number 'N..' when subroutine is called** |
|---|---|
| Description | When a subroutine is called, the internal value of the last block number read is set to zero in the basic setting.<br>As a result, a read access supplies the value 0 at the start of this subroutine via the variable V.G.BLOCK_NR.<br>When the parameter (1) is set, the value of the last block number read from the superior (main) program is retained internally when the subroutine is called.<br>The variable V.G.BLOCK_NR supplies a value > 0 as a result (when block numbers are used). |
| Parameter | remain_block_number_sub_prog_call |

| | |
|---|---|
| Data type | BOOLEAN |
| Data range | 0: Internal value of the last block number programmed is set to zero at subroutine call.<br>1: Internal value of the last block number programmed is retained at subroutine call. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | **Programming example:**<br>*when remain_block_number_sub_prog_call 0*<br>`:`<br>`N1000 X100 Y200 Z300`<br>`N1010 LL SUB1.NC`<br>`-> V.G.BLOCK_NR access at start of SUB1.NC supplies 0 0`<br>*when remain_block_number_sub_prog_call 1*<br>`:`<br>`N1000 X100 Y200 Z300`<br>`N1010 LL SUB1.NC`<br>`-> V.G.BLOCK_NR access at start of SUB1.NC supplies 0 1010` |

# 1.48 Parameterisation of HSC machining (hsc.*)

HSC machining is parameterised via an controller-internal basic initialisation. With the aid of the parameters described below, this internal basic initialisation can also be modified for HSC machining. A distinction is made between special parameters for HSC machining with BSPLINE and general HSC parameters.

In addition, basic parameterisation after program start can be modified or selected using the appropriate HSC commands (#HSC.. [...]), see [PROG].

## 1.48.1 BSPLINE parameters (hsc.bspline.*)

### 1.48.1.1 Deviation from the path contour (P-CHAN-00122)

| **P-CHAN-00122** | **Maximum deviation of the B spline from the programmed path contour** |
|---|---|
| Description | This parameter defines the maximum deviation of the B spline from the programmed path contour. The spline is deselected automatically if this deviation is exceeded. |
| Parameter | hsc.bspline.path_deviation |
| Data type | REAL64 |
| Data range | 0 ... MAX(REAL64) |
| Dimension | 0.1µm |
| Default value | 2000 |
| Remarks | |

### 1.48.1.2 Deviation of the tracked axis (P-CHAN-00123)

| **P-CHAN-00123** | **Deviation of the tracked axis (B Spline)** |
|---|---|
| Description | This parameter defines the maximum deviation of tracked axes. |
| Parameter | hsc.bspline.track_deviation |
| Data type | REAL64 |
| Data range | 0 ... MAX(REAL64) |
| Dimension | 0.1µm or 0.0001° |
| Default value | 50000 |
| Remarks | |

### 1.48.1.3 Path length of relevant blocks (P-CHAN-00124)

| **P-CHAN-00124** | **Path length of relevant blocks (B Spline)** |
|---|---|
| Description | This parameter defines the maximum path length of relevant blocks. If blocks are longer than the specified length, the B Spline is deselected implicitly. |
| Parameter | hsc.bspline.max_path_length |
| Data type | REAL64 |
| Data range | 0 ... MAX(REAL64) |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | |

### 1.48.1.4 Contour bend angle for linear block transitions (P-CHAN-00125)

| **P-CHAN-00125** | **Contour bend angle for linear block transitions (B Spline)** |
|---|---|

| | |
|---|---|
| Description | This parameter defines the maximum contour bend angle for transitions between two linear blocks up to which a B Spline is inserted. The B spline is cancelled internally if the angle between the two linear blocks is greater. |
| Parameter | hsc.bspline.max_angle |
| Data type | REAL64 |
| Data range | 0 ... MAX(REAL64) |
| Dimension | 0.0001° |
| Default value | 0 |
| Remarks | |

### 1.48.1.5 Cancellation criteria for merging blocks (P-CHAN-00126)

| **P-CHAN-00126** | **Cancellation criteria for merging blocks (B Spline)** |
|---|---|
| Description | The parameter determines the maximum number of attempts to merge blocks into one spline segment in the current merge window P-CHAN-00127 [▶ 217] . |
| Parameter | hsc.bspline.merge_retry |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 5 |
| Remarks | |

### 1.48.1.6 Start criteria for merging blocks (P-CHAN-00127)

| **P-CHAN-00127** | **Start criteria for merging blocks (B Spline)** |
|---|---|
| Description | Merging into spline segments is always started when the number of blocks defined in this parameter is reached. |
| Parameter | hsc.bspline.merge_window |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 5 |
| Remarks | |

### 1.48.1.7 Spline deselection with G00 blocks (P-CHAN-00128)

| **P-CHAN-00128** | **Spline deselection with G00 blocks (B Spline)** |
|---|---|
| Description | This parameter defines the automatic deselection of B Spline interpolation with G00 blocks. |
| Parameter | hsc.bspline.auto_off_g00 |
| Data type | BOOLEAN |
| Data range | 0: No implicit deselection due to a rapid traverse block<br>1: Implicit deselection due to a rapid traverse block |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.48.1.8 Response to maximum path deviation (P-CHAN-00129)

| **P-CHAN-00129** | **Response to maximum path deviation (B Spline)** |
|---|---|

| | |
|---|---|
| Description | When the maximum path deviation is exceeded, this parameter defines what measure is to be applied during B spline generation. |
| Parameter | hsc.bspline.auto_off_path |
| Data type | BOOLEAN |
| Data range | 0: The B spline is changed (corner distance is reduced) in order to maintain the specified maximum corner deviation and the B spline touches the original path before and after the corner. The B spline is not deselected.<br>1: The B spline is automatically deselected at the corner. Therefore, any contour bend at the corner is retained ('sharp corner'). |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.48.1.9 Response to maximum tracked axis deviation (P-CHAN-00130)

| P-CHAN-00130 | Response to maximum tracked axis deviation (B Spline) |
|---|---|
| Description | This parameter defines which measure is used for B spline generation when the maximum tracked axis deviation is exceeded. |
| Parameter | hsc.bspline.auto_off_track |
| Data type | BOOLEAN |
| Data range | 0: The B spline is reduced (corner distance is reduced) in order to maintain the predefined maximum deviation and the B spline touches the original path before and after the corner. The B spline is not deselected.<br>1: The B spline is automatically deselected at the corner. Therefore, any contour bend at the corner is retained. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.48.1.10 Limiting the corner distance (P-CHAN-00131)

| P-CHAN-00131 | Limit the corner distance (B Spline) |
|---|---|
| Description | This parameter defines the minimum corner distance depending on the original block length and this becomes the limit for B spline generation, especially with prismatic parts. |
| Parameter | hsc.bspline.limit_corner_dist |
| Data type | REAL64 |
| Data range | 0 ... 250 (25%) |
| Dimension | 0.1% |
| Default value | 200 |
| Remarks | |

### 1.48.1.11 Automatic insertion of polynomials on internal B spline deselection (P-CHAN-00239)

| P-CHAN-00239 | Automatic insertion of polynomials on internal B spline deselection |
|---|---|

<table>
<tr><td rowspan="7">Description</td><td colspan="2">This parameter automatically selects polynomial smoothing (G261, #CONTOUR MODE [...]) when the B spline is deselected internally. The following B spline parameters #HSC[BSPLINE PATH_DEV=<b>X</b> TRACK_DEV=<b>Y</b>] are adopted for polynomial smoothing:</td></tr>
<tr><td>#CONTOUR MODE[</td><td>DEV</td></tr>
<tr><td></td><td>PATH_DEV=<b>X</b></td></tr>
<tr><td></td><td>TRACK_DEV=<b>Y</b></td></tr>
<tr><td></td><td>RELEVANT_PATH=<b>X</b></td></tr>
<tr><td></td><td>RELEVANT_TRACK=<b>Y</b> ]</td></tr>
<tr><td colspan="2">The exact response of the parameters is described in [PROG//section Polynomial smoothing (G260/G261)] .</td></tr>
<tr><td>Parameter</td><td colspan="2">hsc.bspline.auto_contour_mode</td></tr>
<tr><td>Data type</td><td colspan="2">UNS32</td></tr>
<tr><td>Data range</td><td colspan="2">0: Deactivated (default).<br>1: General insertion of smoothing polynomial if B spline is internally deselected.<br>2: Insert smoothing polynomial on linear-circular, circular-circular and circular-linear transitions (recommended mode).</td></tr>
<tr><td>Dimension</td><td colspan="2">----</td></tr>
<tr><td>Default value</td><td colspan="2">0</td></tr>
<tr><td>Remarks</td><td colspan="2"></td></tr>
</table>

### 1.48.1.12 Definition of feed group for B spline (P-CHAN-00240)

| **P-CHAN-00240** | **Define feed group for B spline** |
|---|---|
| Description | This parameter activates the evaluation of the feed group for B spline. In the basic state the feed is always related on the first three main axes. If a different response is required, set the parameter P-CHAN-00240. Then define the feed group using the channel configuration P-CHAN-00011 [▸ 140] or the programming command #FGROUP. |
| Parameter | hsc.bspline.fgroup |
| Data type | BOOLEAN |
| Data range | 0: Inactive (default).<br>1: If spline interpolation is active, the feed group is evaluated. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.48.1.13 HSC deselection when a deviation is equal to zero (P-CHAN-00241)

| **P-CHAN-00241** | **Deselect HSC if a deviation is equal to zero (B Spline)** |
|---|---|
| Description | If one of the PATH_DEV or TRACK_DEV tolerances is programmed with 0 for HSC machining in BSPLINE mode, no contour with tolerance 0 (i.e. the programmed contour) occurs. Instead there is a distorted contour with any number of large deviations.<br>Therefore, meaningful combinations are only either both deviations != 0 or both deviations = 0. An error message is generated if only one of the two deviations is 0. To achieve this response, P-CHAN-00241 = 1 must be set. |
| Parameter | hsc.bspline.no_hsc_for_deviation_zero |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |

<table>
<tr><td rowspan="6">Remarks</td><td colspan="3">Parameterisation example:<br><i>hsc.bspline.no_hsc_for_deviation_zero 1</i><br>.... configures the following response:</td></tr>
<tr><td><i>PATH_DEV</i></td><td><i>TRACK_DEV</i></td><td></td></tr>
<tr><td>>0</td><td>>0</td><td><i>Monitor deviation of main and tracked axes</i></td></tr>
<tr><td>0</td><td>0</td><td><i>Implicit deselection of HSC machining (#HSC OFF)</i></td></tr>
<tr><td>>0</td><td>0</td><td><i>Error message 21682</i></td></tr>
<tr><td>0</td><td>>0</td><td><i>Error message 21682</i></td></tr>
</table>

### 1.48.1.14 Deselecting spline at exact stop G60 (P-CHAN-00421)

| **P-CHAN-00421** | **Deselecting spline at programmed exact stop G60 (B Spline)** |
|---|---|
| Description | This parameter defines the automatic deselection of B Spline interpolation for programmed exact stop G60 or G360. |
| Parameter | hsc.bspline.auto_off_g60 |
| Data type | BOOLEAN |
| Data range | 0: No implicit deselection due to exact stop<br>1: Implicit deselection with programmed exact stop |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.48.2 General HSC parameters (hsc.gen.*)

### 1.48.2.1 Segmentation of linear blocks (P-CHAN-00132)

| **P-CHAN-00132** | **Linear block segmentation (HSC)** |
|---|---|
| Description | The parameter defines whether linear blocks are to be segmented before splining |
| Parameter | hsc.gen.linear_segmentation |
| Data type | BOOLEAN |
| Data range | 0: Linear block segmentation inactive<br>1: Linear block segmentation active. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.48.2.2 Block selection in the case of linear segmentation (P-CHAN-00133)

| **P-CHAN-00133** | **Number of blocks for linear segmentation (HSC)** |
|---|---|
| Description | This parameter defines the number of blocks which are generated during linear segmentation. Set the parameter P-CHAN-00134 [▶ 221] to 0. Depending on the block motion path available, 4 or 3 block segments are generated. In both cases, the segments at the start and end of the block have the length P-CHAN-00135 [▶ 221]. If the block motion path is not long enough (shorter than 3* P-CHAN-00135 [▶ 221]), the original motion path is split into two blocks. |
| Parameter | hsc.gen.linear_center_point |
| Data type | BOOLEAN |

| | |
|---|---|
| Data range | 0: 3 block segments where the first and last segment P-CHAN-00135 [▶ 221] are long and the middle segment has the length of the distance to go.<br>1: 4 block segments where the first and last segment P-CHAN-00135 [▶ 221] are long and the middle two segments share the distance to go. |
| Dimension | ---- |
| Default value | 1 |
| Remarks | |

### 1.48.2.3 Segmentation specification (P-CHAN-00134)

| **P-CHAN-00134** | **Segmentation specification (HSC)** |
|---|---|
| Description | This parameter determines whether linear blocks are generated continuously, i.e. with equal segment lengths, or whether the special segmentation specification as per P-CHAN-00133 [▶ 220] is applied. |
| Parameter | hsc.gen.linear_continuous_split |
| Data type | BOOLEAN |
| Data range | 0: Special segmentation specification as per P-CHAN-00133 [▶ 220] is active.<br>1: Continuous block segmentation, all segments are of identical length. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.48.2.4 Segment length with linear blocks (P-CHAN-00135)

| **P-CHAN-00135** | **Segment length with linear blocks (HSC)** |
|---|---|
| Description | This parameter defines the segment length with linear blocks. |
| Parameter | hsc.gen.linear_split_length |
| Data type | UNS32 |
| Data range | 1 ... MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 1000 |
| Remarks | |

### 1.48.2.5 Basis of segmentation of circular blocks (P-CHAN-00136)

| **P-CHAN-00136** | **Basis of segmentation of circular blocks (HSC)** |
|---|---|
| Description | This parameter defines whether the block length for segmentation of circular blocks is based on secant error or path length. The secant error or the path length is specified by P-CHAN-00138 [▶ 222]. |
| Parameter | hsc.gen.circular_secant_error |
| Data type | BOOLEAN |
| Data range | 0: Segmentation of circular blocks based on secant error.<br>1: Segmentation of circular blocks based on path length. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.48.2.6 Segmentation of circular blocks (P-CHAN-00137)

| **P-CHAN-00137** | **Segmentation of circular blocks (HSC)** |
|---|---|

| | |
|---|---|
| Description | This parameter defines whether circular blocks are segmented before splining. |
| Parameter | hsc.gen.circular_segmentation |
| Data type | BOOLEAN |
| Data range | 0: Segmentation of circular blocks inactive (default).<br>1: Segmentation of circular blocks active. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.48.2.7 Contour error on segmentation of circular blocks (P-CHAN-00138)

| **P-CHAN-00138** | **Contour error when circular blocks are segmented (HSC)** |
|---|---|
| Description | This parameter defines the permissible contour error when circular blocks are segmented. Depending on P-CHAN-00136 [▶ 221] , either the secant error or the path length of the segment is specified. |
| Parameter | hsc.gen.circular_param |
| Data type | UNS32 |
| Data range | 1 ... MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 50 |
| Remarks | |

### 1.48.2.8 Handling short blocks (P-CHAN-00139)

| **P-CHAN-00139** | **Handling short blocks (HSC)** |
|---|---|
| Description | This parameter defines whether blocks that fall below a minimum length are removed |
| Parameter | hsc.gen.filter_deviation |
| Data type | UNS32 |
| Data range | 0: No removal of short blocks (default).<br>>0: Blocks that are shorter than P-CHAN-00139 are removed. |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | |

### 1.48.2.9 Minimum segment length for block segmentation (P-CHAN-00140)

| **P-CHAN-00140** | **Minimum segment length for block segmentation (HSC)** |
|---|---|
| Description | This parameter defines the minimum segment length for block segmentation due to the dynamics in the contour. Lower values lead to improved use of the dynamics but also to a larger number of blocks. |
| Parameter | hsc.gen.min_segment_length |
| Data type | UNS32 |
| Data range | 1 ... MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 200 |
| Remarks | |

### 1.48.2.10 Jerk limiting algorithm (P-CHAN-00141)

| **P-CHAN-00141** | **Jerk limiting algorithm (HSC)** |
|---|---|
| Description | This parameter defines which algorithm is used for jerk limiting. |
| Parameter | hsc.gen.jerk_monitoring_mode |
| Data type | UNS32 |
| Data range | 0: Jerk limiting without consideration of internal derivations of the polynomials (du/ds).<br>1: Jerk limiting with inclusion of internal derivations of the polynomials (dr/ds).<br>2: Jerk limiting with minimum speed from 0 and 1, MIN[du/ds, dr/ds]. |
| Dimension | ---- |
| Default value | 1 |
| Remarks | |

### 1.48.2.11 Factor for permissible jerk in 5-axis mode (P-CHAN-00142)

| **P-CHAN-00142** | **Factor for permissible jerk in 5-axis mode (HSC)** |
|---|---|
| Description | This parameter defines the weighting factor for the permissible jerk in 5-axis mode. |
| Parameter | hsc.gen.jerk_weighting_5ax |
| Data type | UNS32 |
| Data range | 1 ... MAX(UNS32) |
| Dimension | 0.1% |
| Default value | 2000 |
| Remarks | |

### 1.48.2.12 Factor for permissible jerk in 2.5D mode (P-CHAN-00143)

| **P-CHAN-00143** | **Factor for permissible jerk in 2.5D mode (HSC)** |
|---|---|
| Description | This parameter defines the weighting factor for the permissible jerk in 2.5D mode. |
| Parameter | hsc.gen.jerk_weighting |
| Data type | UNS32 |
| Data range | 1 ... MAX(UNS32) |
| Dimension | 0.1% |
| Default value | 2000 |
| Remarks | |

### 1.48.2.13 Delayed HSC selection (BSPLINE) (P-CHAN-00217)

| **P-CHAN-00217** | **Delayed HSC selection for contour spline processing** |
|---|---|
| Description | With certain machining technologies in connection with NC programs generated from a post processor, it can be advantageous to keep the first motion block of an HSC contour (start-up block) unchanged after #HSC ON [BSPLINE]. This delays contour splining by one motion block. |
| Parameter | hsc.gen.on_delay_output_one_block |
| Data type | BOOLEAN |
| Data range | 0: First motion block after HSC selection is used for splining (default).<br>1: First motion block after HSC selection is not used for splining (default). HSC selection is active after the first subsequent linear or circular motion block. |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | |

### 1.48.2.14 HSC deselection behaviour for tracking axes (P-CHAN-00405)

| P-CHAN-00405 | HSC deselection response for tracking axes |
|---|---|
| Description | This parameter defines the ratio between the motion paths of tracking and main axes before automatic deselection of the splining process. This is especially helpful if great changes of orientation take place with small main axis motion paths but automatic deselection of the HSC spine is not desired. Larger values mean fewer automatic deselection operations in the splining process. |
| Parameter | hsc.gen.max_track_ratio |
| Data type | REAL64 |
| Data range | 0 < max_track_ratio < MAX(REAL64) |
| Dimension | ---- |
| Default value | 100.0 |
| Remarks | |

### 1.48.2.15 Maximum number of segmented blocks for feed profile planning (P-CHAN-00375)

| P-CHAN-00375 | Maximum number of segmented blocks for feed profile planning (HSC) |
|---|---|
| Description | To avoid unnecessarily small feed values, motion blocks are segmented. This parameter defines the maximum number of additional blocks generated by segmentation. |
| Parameter | hsc.gen.slope_segmentation_nbr_blocks |
| Data type | UNS32 |
| Data range | 0 < slope_segmentation_nbr_blocks < MAX(UNS32) |
| Dimension | ---- |
| Default value | 100 |
| Remarks | The default value may only be changed after approval by the controller manufacturer. |

### 1.48.2.16 Analytical dynamic calculation for spline curves (P-CHAN-00468)

| P-CHAN-00468 | Analytical dynamic calculation for spline curves |
|---|---|
| Description | This parameter can activate a more precise dynamic calculation to avoid dynamic overloads in spine curves. |
| Parameter | hsc.gen.use_analytic_dyn_calc |
| Data type | UNS16 |
| Data range | 0: Only an approximative dynamic calculation is executed in spline curves (default)<br>1: Analytical dynamic calculation in spline curves |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter available as of CNC Build V3.1.3079.09 and higher |

## 1.49 Channel synchronisation on axis exchange (P-CHAN-00154)

| P-CHAN-00154 | Axis exchange with implicit channel synchronisation (FLUSH CONTINUE) |
|---|---|

| | |
|---|---|
| Description | This parameter offers the possibility of triggering an implicit FLUSH WAIT for axis exchange commands #CALL AX, #AX REQUEST, #SET AX and #AX DEF. An implicit FLUSH CONTINUE is triggered with axis exchange commands #PUT AX and #AX RELEASE. This can avoid jamming between NC channels with axis exchange operations. |
| Parameter | ax_exchange_with_implicit_flush |
| Data type | BOOLEAN |
| Data range | 0: Default execution of all axis exchange commands according to the programming manual (default).<br>1: Output of an implicit FLUSH WAIT with axis exchange commands #CALL AX, #AX REQUEST, #SET AX and #AX DEF. An implicit FLUSH CONTINUE is triggered with axis exchange commands #PUT AX and #AX RELEASE.<br>FLUSH WAIT and FLUSH CONTINUE are generated before the actual axis exchange sequence. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.50 Channel initialisation with actual values (P-CHAN-00455)

| **P-CHAN-00455** | **Channel initialisation with actual values** |
|---|---|
| Description | This parameter influences the behaviour of the command #CHANNEL INIT [ACTPOS].<br>When the parameter is set, the actual positions are requested and synchronised in the channel when the command and an explicit setting exclusively for these axes are received.<br>If P-CHAN-00455 is not set, the command positions of all unspecified axes are also requested. This may results in a waiting condition if an unspecified axis is moved by an independent command (e.g. MC_MoveAbsolute or INDP_ASYN).<br>In the programming example below, the program waits after the NC command #CHANNEL INIT until the axis reaches its target and only then is the NC channel synchronised with the current positions.<br>`N10 X[INDP_ASYN G90 G00 POS=20]`<br>`; when P-CHAN-00455 = 0, the program waits in N20 until X moves to 20`<br>`N20 #CHANNEL INIT [ACTPOS AX=Y]`<br>`N30 Y=20`<br>`N40 X=0 Y=0` |
| Parameter | channel_init_actpos_exclusive |
| Data type | BOOLEAN |
| Data range | |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.51 Program name for automatic streaming (P-CHAN-00158)

| **P-CHAN-00158** | **Program name for automatic streaming** |
|---|---|

| | |
|---|---|
| Description | When this program is opened as a main program or a subroutine, the ASCII data is not read in from the file system, but is requested from the data streaming interface. The data input is therefore diverted transparently to the streaming interface.<br>If the program name is not entered, the streaming function can not be activated |
| Parameter | streaming_prog_file |
| Data type | STRING |
| Data range | <empty_string>: Streaming function deactivated (default).<br><prog_name>: Name of the file (main program/global subroutine) that automatically triggers a changeover to streaming mode when it is started. |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameterisation example:<br>*streaming_prog_file streaming.nc (name of streaming program)*<br>* Note: The default value of variables is a blank string. |

## 1.52 Modal effect of I/J/K for circle programming (P-CHAN-00159)

| P-CHAN-00159 | Modal effect of I/J/K for circle programming |
|---|---|
| Description | This parameter sets the local/modal effectiveness of the centre point coordinates I, J, K for circles in the NC program [PROG//section Circle interpolation (G02/G03)]]. |
| Parameter | modal_i_j_k_for_circle |
| Data type | BOOLEAN |
| Data range | 0: The circle centre point coordinates I, J, K are non-modal (local). This means they must be reprogrammed for each new G02/G03 NC block.<br>1: The circle centre point coordinates I, J, K are modal. This means they remain valid in all the following G02/G03 NC blocks. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.53 Subroutine calls using G functions

### 1.53.1 Implicit call with G80 (P-CHAN-00160 - P-CHAN-00169)

| P-CHAN-00160 | Implicit call with G80 |
|---|---|
| Description | The parameter ‘g80_prog_file’ defines the name of the global subroutine which is called implicitly when a G80 is detected in the NC program. G80 is always executed as the last action at block end.<br>If the related parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 ‘Unknown G function’ is generated if G80 is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G80=“<prog_name>“]. The new subroutine name is then valid until program end M30 or until a further #FILE NAME [G8x=“<prog_name>“] is valid.<br>Then after RESET or program start, the entry 'g80_prog_file' from the channel parameter list applies again. |
| Parameter | g80_prog_file |
| Data type | STRING |

| | |
|---|---|
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | **Parameterisation example:**<br>**g80_prog_file G80_up_test.nc (subroutine name with G80)**<br>*g81_prog_file G81_up_test.nc (subroutine name with G81)*<br>*g82_prog_file G82_up_test.nc (subroutine name with G82)*<br>*g83_prog_file G83_up_test.nc (subroutine name with G83)*<br>*g84_prog_file G84_up_test.nc (subroutine name with G84)*<br>*g85_prog_file G85_up_test.nc (subroutine name with G85)*<br>*g86_prog_file G86_up_test.nc (subroutine name with G86)*<br>*g87_prog_file G87_up_test.nc (subroutine name with G87)*<br>*g88_prog_file G88_up_test.nc (subroutine name with G88)*<br>*g89_prog_file G89_up_test.nc (subroutine name with G89)*<br><br>* Note: The default value of variables is a blank string. |

## 1.53.2 Implicit call with G81 (P-CHAN-00161)

| **P-CHAN-00161** | **Implicit call with G81** |
|---|---|
| Description | The parameter 'g81_prog_file' defines the name of a global subroutine which is called implicitly when G81 is detected in the NC program. G81 is always executed as the last action at block end.<br>If the parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 'Unknown G function' is generated if G81 is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G81="<prog_name>"]. The new subroutine name is then valid until program end M30 or up to the next #FILE NAME [G81="<prog_name>"].<br>Then after RESET or program start, the entry 'g81_prog_file' from the channel parameter list applies again. |
| Parameter | g81_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | **Parameterisation example:**<br>*g80_prog_file G80_up_test.nc (subroutine name with G80)*<br>**g81_prog_file G81_up_test.nc (subroutine name with G81)**<br>*g82_prog_file G82_up_test.nc (subroutine name with G82)*<br>*g83_prog_file G83_up_test.nc (subroutine name with G83)*<br>*g84_prog_file G84_up_test.nc (subroutine name with G84)*<br>*g85_prog_file G85_up_test.nc (subroutine name with G85)*<br>*g86_prog_file G86_up_test.nc (subroutine name with G86)*<br>*g87_prog_file G87_up_test.nc (subroutine name with G87)*<br>*g88_prog_file G88_up_test.nc (subroutine name with G88)*<br>*g89_prog_file G89_up_test.nc (subroutine name with G89)*<br><br>* Note: The default value of variables is a blank string. |

## 1.53.3 Implicit call with G82 (P-CHAN-00162)

| **P-CHAN-00162** | **Implicit call with G82** |
|---|---|

| | |
|---|---|
| Description | The parameter 'g82_prog_file’ defines the name of a global subroutine which is called implicitly when G82 is detected in the NC program. G82 is always executed as the last action at block end.<br>If the parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 ‘Unknown G function’ is generated if G82 is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G82=“<prog_name>“]. The new subroutine name is then valid until program end M30 or up to the next #FILE NAME [G82=“<prog_name>“].<br>Then after RESET or program start, the entry 'g82_prog_file' from the channel parameter list applies again. |
| Parameter | g82_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | **Parameterisation example:**<br>`g80_prog_file G80_up_test.nc (subroutine name with G80)`<br>`g81_prog_file G81_up_test.nc (subroutine name with G81)`<br>**`g82_prog_file G82_up_test.nc (subroutine name with G82)`**<br>`g83_prog_file G83_up_test.nc (subroutine name with G83)`<br>`g84_prog_file G84_up_test.nc (subroutine name with G84)`<br>`g85_prog_file G85_up_test.nc (subroutine name with G85)`<br>`g86_prog_file G86_up_test.nc (subroutine name with G86)`<br>`g87_prog_file G87_up_test.nc (subroutine name with G87)`<br>`g88_prog_file G88_up_test.nc (subroutine name with G88)`<br>`g89_prog_file G89_up_test.nc (subroutine name with G89)`<br><br>`* Note: The default value of variables is a blank string.` |

## 1.53.4 Implicit call with G83 (P-CHAN-00163)

| **P-CHAN-00163** | **Implicit call with G83** |
|---|---|
| Description | The parameter 'g83_prog_file’ defines the name of a global subroutine which is called implicitly when G83 is detected in the NC program. G83 is always executed as the last action at block end.<br>If the parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 'Unknown G function’ is generated if G83 is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G83=“<prog_name>“]. The new subroutine name is then valid until program end M30 or up to the next #FILE NAME [G83=“<prog_name>“].<br>Then after RESET or program start, the entry 'g83_prog_file' from the channel parameter list applies again. |
| Parameter | g83_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |

| | |
|---|---|
| Remarks | **Parameterisation example:**<br>`g80_prog_file G80_up_test.nc (subroutine name with G80)`<br>`g81_prog_file G81_up_test.nc (subroutine name with G81)`<br>`g82_prog_file G82_up_test.nc (subroutine name with G82)`<br>**`g83_prog_file G83_up_test.nc (subroutine name with G83)`**<br>`g84_prog_file G84_up_test.nc (subroutine name with G84)`<br>`g85_prog_file G85_up_test.nc (subroutine name with G85)`<br>`g86_prog_file G86_up_test.nc (subroutine name with G86)`<br>`g87_prog_file G87_up_test.nc (subroutine name with G87)`<br>`g88_prog_file G88_up_test.nc (subroutine name with G88)`<br>`g89_prog_file G89_up_test.nc (subroutine name with G89)`<br><br>`* Note: The default value of variables is a blank string.` |

## 1.53.5 Implicit call with G84 (P-CHAN-00164)

| P-CHAN-00164 | Implicit call with G84 |
|---|---|
| Description | The parameter 'g84_prog_file’ defines the name of a global subroutine which is called implicitly when G84 is detected in the NC program. G84 is always executed as the last action at block end.<br>If the parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 'Unknown G function’ is generated if G84 is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G84=“<prog_name>“]. The new subroutine name is then valid until program end M30 or up to the next #FILE NAME [G84=“<prog_name>“].<br>Then after RESET or program start, the entry 'g84_prog_file' from the channel parameter list applies again. |
| Parameter | g84_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | **Parameterisation example:**<br>`g80_prog_file G80_up_test.nc (subroutine name with G80)`<br>`g81_prog_file G81_up_test.nc (subroutine name with G81)`<br>`g82_prog_file G82_up_test.nc (subroutine name with G82)`<br>`g83_prog_file G83_up_test.nc (subroutine name with G83)`<br>**`g84_prog_file G84_up_test.nc (subroutine name with G84)`**<br>`g85_prog_file G85_up_test.nc (subroutine name with G85)`<br>`g86_prog_file G86_up_test.nc (subroutine name with G86)`<br>`g87_prog_file G87_up_test.nc (subroutine name with G87)`<br>`g88_prog_file G88_up_test.nc (subroutine name with G88)`<br>`g89_prog_file G89_up_test.nc (subroutine name with G89)`<br><br>`* Note: The default value of variables is a blank string.` |

## 1.53.6 Implicit call with G85 (P-CHAN-00165)

| P-CHAN-00165 | Implicit call with G85 |
|---|---|

| | |
|---|---|
| Description | The parameter 'g85_prog_file’ defines the name of a global subroutine which is called implicitly when G85 is detected in the NC program. G85 is always executed as the last action at block end.<br>If the parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 'Unknown G function’ is generated if G85 is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G85=“<prog_name>“]. The new subroutine name is then valid until program end M30 or up to the next #FILE NAME [G85=“<prog_name>“].<br>Then after RESET or program start, the entry 'g85_prog_file' from the channel parameter list applies again. |
| Parameter | g85_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | **Parameterisation example:**<br>`g80_prog_file G80_up_test.nc (subroutine name with G80)`<br>`g81_prog_file G81_up_test.nc (subroutine name with G81)`<br>`g82_prog_file G82_up_test.nc (subroutine name with G82)`<br>`g83_prog_file G83_up_test.nc (subroutine name with G83)`<br>`g84_prog_file G84_up_test.nc (subroutine name with G84)`<br>**`g85_prog_file G85_up_test.nc (subroutine name with G85)`**<br>`g86_prog_file G86_up_test.nc (subroutine name with G86)`<br>`g87_prog_file G87_up_test.nc (subroutine name with G87)`<br>`g88_prog_file G88_up_test.nc (subroutine name with G88)`<br>`g89_prog_file G89_up_test.nc (subroutine name with G89)`<br><br>`* Note: The default value of variables is a blank string.` |

## 1.53.7 Implicit call with G86 (P-CHAN-00166)

| **P-CHAN-00166** | **Implicit call with G86** |
|---|---|
| Description | The parameter 'g86_prog_file’ defines the name of a global subroutine which is called implicitly when G86 is detected in the NC program. G86 is always executed as the last action at block end.<br>If the parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 ‘Unknown G function’ is generated if G86 is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G86=“<prog_name>“]. The new subroutine name is then valid until program end M30 or up to the next #FILE NAME [G86=“<prog_name>“].<br>Then after RESET or program start, the entry 'g86_prog_file' from the channel parameter list applies again. |
| Parameter | g86_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |

| | |
|---|---|
| Remarks | **Parameterisation example:**<br>g80_prog_file G80_up_test.nc (subroutine name with G80)<br>g81_prog_file G81_up_test.nc (subroutine name with G81)<br>g82_prog_file G82_up_test.nc (subroutine name with G82)<br>g83_prog_file G83_up_test.nc (subroutine name with G83)<br>g84_prog_file G84_up_test.nc (subroutine name with G84)<br>g85_prog_file G85_up_test.nc (subroutine name with G85)<br>**g86_prog_file G86_up_test.nc (subroutine name with G86)**<br>g87_prog_file G87_up_test.nc (subroutine name with G87)<br>g88_prog_file G88_up_test.nc (subroutine name with G88)<br>g89_prog_file G89_up_test.nc (subroutine name with G89)<br><br>* Note: The default value of variables is a blank string. |

## 1.53.8 Implicit call with G87 (P-CHAN-00167)

| P-CHAN-00167 | Implicit call with G87 |
|---|---|
| Description | The parameter 'g87_prog_file' defines the name of a global subroutine which is called implicitly when G87 is detected in the NC program. G87 is always executed as the last action at block end.<br>If the parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 'Unknown G function' is generated if G87 is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G87=“<prog_name>“]. The new subroutine name is then valid until program end M30 or up to the next #FILE NAME [G87=“<prog_name>“].<br>Then after RESET or program start, the entry 'g87_prog_file' from the channel parameter list applies again. |
| Parameter | g87_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | **Parameterisation example:**<br>g80_prog_file G80_up_test.nc (subroutine name with G80)<br>g81_prog_file G81_up_test.nc (subroutine name with G81)<br>g82_prog_file G82_up_test.nc (subroutine name with G82)<br>g83_prog_file G83_up_test.nc (subroutine name with G83)<br>g84_prog_file G84_up_test.nc (subroutine name with G84)<br>g85_prog_file G85_up_test.nc (subroutine name with G85)<br>g86_prog_file G86_up_test.nc (subroutine name with G86)<br>**g87_prog_file G87_up_test.nc (subroutine name with G87)**<br>g88_prog_file G88_up_test.nc (subroutine name with G88)<br>g89_prog_file G89_up_test.nc (subroutine name with G89)<br><br>* Note: The default value of variables is a blank string. |

## 1.53.9 Implicit call with G88 (P-CHAN-00168)

| P-CHAN-00168 | Implicit call with G88 |
|---|---|

| | |
|---|---|
| Description | The parameter 'g88_prog_file' defines the name of a global subroutine which is called implicitly when G88 is detected in the NC program. G88 is always executed as the last action at block end.<br>If the parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 'Unknown G function' is generated if G88 is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G88="<prog_name>"]. The new subroutine name is then valid until program end M30 or up to the next #FILE NAME [G88="<prog_name>"].<br>Then after RESET or program start, the entry 'g88_prog_file' from the channel parameter list applies again. |
| Parameter | g88_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | **Parameterisation example:**<br>*g80_prog_file G80_up_test.nc (subroutine name with G80)*<br>*g81_prog_file G81_up_test.nc (subroutine name with G81)*<br>*g82_prog_file G82_up_test.nc (subroutine name with G82)*<br>*g83_prog_file G83_up_test.nc (subroutine name with G83)*<br>*g84_prog_file G84_up_test.nc (subroutine name with G84)*<br>*g85_prog_file G85_up_test.nc (subroutine name with G85)*<br>*g86_prog_file G86_up_test.nc (subroutine name with G86)*<br>*g87_prog_file G87_up_test.nc (subroutine name with G87)*<br>**g88_prog_file G88_up_test.nc (subroutine name with G88)**<br>*g89_prog_file G89_up_test.nc (subroutine name with G89)*<br><br>* Note: The default value of variables is a blank string. |

## 1.53.10 Implicit call with G89 (P-CHAN-00169)

| **P-CHAN-00169** | **Implicit call with G89** |
|---|---|
| Description | The parameter 'g89_prog_file' defines the name of a global subroutine which is called implicitly when G89 is detected in the NC program. G89 is always executed as the last action at block end.<br>If the parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 'Unknown G function' is generated if G89 is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G89="<prog_name>"]. The new subroutine name is then valid until program end M30 or up to the next #FILE NAME [G89="<prog_name>"].<br>Then after RESET or program start, the entry 'g89_prog_file' from the channel parameter list applies again. |
| Parameter | g89_prog_file |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |

| | |
|---|---|
| Remarks | **Parameterisation example:**<br>*g80_prog_file G80_up_test.nc (subroutine name with G80)*<br>*g81_prog_file G81_up_test.nc (subroutine name with G81)*<br>*g82_prog_file G82_up_test.nc (subroutine name with G82)*<br>*g83_prog_file G83_up_test.nc (subroutine name with G83)*<br>*g84_prog_file G84_up_test.nc (subroutine name with G84)*<br>*g85_prog_file G85_up_test.nc (subroutine name with G85)*<br>*g86_prog_file G86_up_test.nc (subroutine name with G86)*<br>*g87_prog_file G87_up_test.nc (subroutine name with G87)*<br>*g88_prog_file G88_up_test.nc (subroutine name with G88)*<br>**g89_prog_file G89_up_test.nc (subroutine name with G89)**<br><br>* Note: The default value of variables is a blank string. |

## 1.53.11 Additional implicit calls with G8xx (P-CHAN-00187)

| P-CHAN-00187 | Additional implicit calls with G8xx |
|---|---|
| Description | As an extension to the implicit call of subroutines with G80-G89, the indexed parameter ‘g800_prog_file[i]' offers the possibility of defining a further 20 or 40** names of global subroutines which are called implicitly if G800-G819** is detected in the NC program. A G800-G819** is always executed as the last action at block end.<br>If the related parameter is not assigned or not present in the channel parameter list, the error message P-ERR-20131 'Unknown G function’ is generated if G800 - G819** is programmed.<br>The subroutine name defined by the channel parameter can also be overwritten or temporarily assigned by the command #FILE NAME [G8xx=“<prog_name>“]. The new subroutine name is then valid until program end M30 or up to the next #FILE NAME [G8xx=“<prog_name>“].<br>After RESET or program start, the entries of 'g800_prog_file[i]' in the channel parameters list apply again. |
| Parameter | g800_prog_file[i] where i = 0…19 or 0 ... 39** |
| Data type | STRING |
| Data range | Maximum of 83 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | **Parameterisation example:**<br>*g800_prog_file[***0***] G800_up_test.nc (subroutine name with G80***0***)*<br>*g800_prog_file[1] G801_up_test.nc (subroutine name with G801)*<br>…<br>*g800_prog_file[***19***] G819_up_test.nc (subroutine name with G8***19***)*<br>…<br>*g800_prog_file[***39***] G839_up_test.nc (subroutine name with G8***39***)*<br>* Note: The default value of variables is a blank string.<br>** As of Version V3.1.3079.23 40 implicit calls (G800 – G839) are available. |

## 1.54 Active testing whether axis group is 'in position' (P-CHAN-00173)

| P-CHAN-00173 | Active testing whether axis group is 'in position' |
|---|---|
| Description | This parameter suppresses the actual value checking of the exact stop window for single-step mode, programmed stop or optional stop (i.e. typical functions where a handshake is required with the operator).<br>With other functions such as programmed exact stop G60, the check whether the axis group is within the actual value exact stop window always takes place and is independent of this parameter. |
| Parameter | suppress_ax_group_in_pos_check |
| Data type | BOOLEAN |

| | |
|---|---|
| Data range | 0: Actual value check is executed before changing to the next motion block, even for single-step mode, programmed stop or optional stop (default).<br>1: Actual value check is not executed before changing to the next motion block, even for single-step mode, programmed stop or optional stop. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*suppress_ax_group_in_pos_check 1* |

## 1.55 Ignore internal stop conditions with rapid contour visualisation (P-CHAN-00183)

| **P-CHAN-00183** | **Ignore internal stop conditions with rapid contour visualisation** |
|---|---|
| Description | This parameter prevents the NC program from stopping because of internal stop conditions (e. g. M00) during rapid contour visualisation. |
| Parameter | simu_ignore_internal_stop_cond |
| Data type | BOOLEAN |
| Data range | 0: Internal stop conditions are effective (default).<br>1: Internal stop conditions are ignored. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.56 Behaviour with axis errors (P-CHAN-00218)

| **P-CHAN-00218** | **Mode of deceleration response in case of an axis error** |
|---|---|
| Description | This parameter defines whether the axis stops on the programmed path after an axis error or if all channel axes stop at their individual emergency stop deceleration rates P-AXIS-00003 . |
| Parameter | independent_axis_error_stop |
| Data type | BOOLEAN |
| Data range | 0: In case of error, all axes with an error present stop at their individual emergency stop ramps. However, all other axes in the NC channel stop in the path compound.<br>1: In case of error, all axes in the NC channel stop at their individual emergency stop ramps, regardless of whether an error occurred on the axis or not. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.57 Path-dependent dynamic weighting

With certain technologies (e.g. punching), dynamic limit values (velocity, acceleration, jerk) must be weighted during rapid feed motion for **linear movements** dependent on the path profile.

A programmed weighting with G129 in the NC program overwrites the values of the path-dependent dynamic weighting.

### 1.57.1 Activation of dynamic weighting (P-CHAN-00190)

| **P-CHAN-00190** | **Activate path-dependent dynamic weighting** |
|---|---|
| Description | This parameter generally activates path-dependent dynamic weighting. |
| Parameter | dynamic_weighting_active |
| Data type | UNS32 |
| Data range | 0: Path-dependent dynamic weighting not effective.<br>1: Path-dependent dynamic weighting effective for G00 motions.<br>2: Path-dependent dynamic weighting effective for G01 motions.<br>3: Path-dependent dynamic weighting effective for G00 and G01 motions. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Path-dependent dynamic weighting can also be switched in the NC program by the command #DYNAMIC WEIGHT ON/OFF [PATH] [PROG]). |

### 1.57.2 Table of dynamic weightings (dynamic_weighting[i].*)

The channel parameters of the CNC contain a table which defines weighting factors for the dynamic characteristics (v, a, tr) depending on the contour path. The weighting factor of a table index entry is used when the contour path is smaller than the associated motion path limit value in the table. If the contour path is equal to the motion path limit value, the dynamic factors of the next index entry are used.

| **Structure name** | **Index** |
|---|---|
| dynamic_weighting[i] | i = 0 .. 9 (maximum number of table entries: 10, application-specific) |

The following further conditions must be considered when defining table entries:

- The motion path limit in the table increases as index values become larger.
- The axes which form the contour path for dynamic weighting must be marked accordingly (P-AXIS-00015). Weighting is only executed for these axes.
- The weighting factor is limited to a minimum value of 1%. For ramp time, the minimum ramp time tr_min (P-AXIS-00201) must also be considered as the lower limit value.
- Weighting is possible upwards to the maximum value of vb_max (P-AXIS-00212) and a_max (P-AXIS-00008). With ramp time tr_grenz (P-AXIS-00200) the weighting has no upper limit.

### 1.57.3 Motion path limit (P-CHAN-00191)

| **P-CHAN-00191** | **Motion path limit (path-dependent dynamic weighting)** |
|---|---|
| Description | This parameter defines the upper motion path limit. The weighting factors of the current entries under the same index i are effective up to this limit. |
| Parameter | dynamic_weighting[i].path_limit |
| Data type | UNS32 |
| Data range | 0 < path_limit < MAX(UNS32) |

| | |
|---|---|
| Dimension | 0.1 µm or 0.0001° |
| Default value | 0 |
| Remarks | The maximum index i in the table has a special importance:<br>Here the upper limit of the motion path corresponds to the machine-specific maximum contour path. Therefore, this area is valid for all motion blocks which are not covered by the path distance limits of the lower areas of the table. |

## 1.57.4 Weighting factor for speed (P-CHAN-00192)

| **P-CHAN-00192** | **Weighting factor for speed (path-dependent dynamic weighting)** |
|---|---|
| Description | This parameter defines the weighting factor for speed under index i. |
| Parameter | dynamic_weighting[i].velocity_fact |
| Data type | UNS32 |
| Data range | 10 < velocity_fact < MAX(UNS32) |
| Dimension | 0.1% |
| Default value | 0 |
| Remarks | Either the rapid traverse velocity and/or the maximum velocity is weighted depending on P-CHAN-00190. |

## 1.57.5 Weighting factor for acceleration (P-CHAN-00193)

| **P-CHAN-00193** | **Weighting factor for acceleration (path-dependent dynamic weighting)** |
|---|---|
| Description | This parameter defines the weighting factor for acceleration under index i. |
| Parameter | dynamic_weighting[i].acceleration_fact |
| Data type | UNS32 |
| Data range | 10 < acceleration_fact < MAX(UNS32) |
| Dimension | 0.1% |
| Default value | 0 |
| Remarks | Either rapid traverse acceleration and/or G01 acceleration/deceleration is weighted depending on P-CHAN-00190. |

## 1.57.6 Weighting factor for ramp time (P-CHAN-00194)

| **P-CHAN-00194** | **Weighting factor for ramp time (path-dependent dynamic weighting)** |
|---|---|
| Description | This parameter defines the weighting factor for ramp time under index i. |
| Parameter | dynamic_weighting[i].ramp_time_fact |
| Data type | UNS32 |
| Data range | 10 < ramp_time_fact < MAX(UNS32) |
| Dimension | 0.1% |
| Default value | 0 |
| Remarks | Either the rapid traverse ramp time and/or G01 ramp time is weighted depending on P-CHAN-00190. |

### 1.57.7 Example of a table

Basis is the following table of dynamic weighting values with 6 rows:

| Index | Path limit value path_limit [0.1µm] (range limits) | Weighting factor velocity_fact [0.1%] | Weighting factors acceleration_fact [0.1%] | Weighting factors ramp_time_fact [0.1%] |
|---|---|---|---|---|
| 0 | 10000<br>(0-10000) | 100 | 1000 | 1000 |
| 1 | 20000<br>(10000-20000) | 200 | 900 | 1500 |
| 2 | 40000<br>(20000-40000) | 300 | 800 | 2000 |
| 3 | 80000<br>(40000-80000) | 400 | 700 | 3000 |
| 4 | 160000<br>(80000-160000) | 500 | 600 | 4000 |
| 5 | 200000000<br>(160000-200000000) | 1000 | 500 | 5000 |

**Mapping of the weighting table on the structure of the channel parameters:**

```
# Path-dependent dynamic weighting
# ================================
dynamic_weighting_active 1 0: not active 1: active
#
dynamic_weighting[0].path_limit 10000 [0.1µm]
dynamic_weighting[0].velocity_fact 100 [0.1%]
dynamic_weighting[0].acceleration_fact 1000 [0.1%]
dynamic_weighting[0].ramp_time_fact 1000 [0.1%]

dynamic_weighting[1].path_limit 20000 [0.1µm]
dynamic_weighting[1].velocity_fact 200 [0.1%]
dynamic_weighting[1].acceleration_fact 900 [0.1%]
dynamic_weighting[1].ramp_time_fact 1500 [0.1%]

dynamic_weighting[2].path_limit 40000 [0.1µm]
dynamic_weighting[2].velocity_fact 300 [0.1%]
dynamic_weighting[2].acceleration_fact 800 [0.1%]
dynamic_weighting[2].ramp_time_fact 2000 [0.1%]

dynamic_weighting[3].path_limit 80000 [0.1µm]
dynamic_weighting[3].velocity_fact 400 [0.1%]
dynamic_weighting[3].acceleration_fact 700 [0.1%]
dynamic_weighting[3].ramp_time_fact 3000 [0.1%]

dynamic_weighting[4].path_limit 160000 [0.1µm]
dynamic_weighting[4].velocity_fact 500 [0.1%]
dynamic_weighting[4].acceleration_fact 600 [0.1%]
dynamic_weighting[4].ramp_time_fact 4000 [0.1%]

dynamic_weighting[5].path_limit 200000000 [0.1µm]
dynamic_weighting[5].velocity_fact 1000 [0.1%]
dynamic_weighting[5].acceleration_fact 500 [0.1%]
dynamic_weighting[5].ramp_time_fact 5000 [0.1%]
```

## 1.58 Radius-dependent dynamic weighting

With certain technologies (e.g. laser machining), dynamic limit values (velocity, acceleration, jerk) must be limited during rapid feed motion for **circular movements** depending on the circle radius.

A weighting with G130 to G133 programmed in the NC program has an cumulative effect on radius-dependent dynamic weighting.

## 1.58.1 Activation of dynamic weighting (P-CHAN-00230)

| **P-CHAN-00230** | **Activate radius-dependent dynamic weighting** |
|---|---|
| Description | This parameter generally enables radius-dependent dynamic weighting. |
| Parameter | curve_dynamic_weighting_active |
| Data type | BOOLEAN |
| Data range | 0: Table of dynamic weightings inactive. Radius-dependent weighting factors are not effective.<br>1: Table of dynamic weighting active. Radius-dependent weighting factors are not effective. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Radius-dependent dynamic weighting can also be switched in NC program by the command #DYNAMIC WEIGHT ON/OFF [CURVE] ([PPROG]). |

## 1.58.2 Table of dynamic weighting (curve_dynamic_weighting[i].*)

The channel parameters contain a table which defines weighting factors for the dynamic characteristics (v, a, tr) dependent on the circle radius. The weighting factor of a table index entry is used when the radius is smaller than the corresponding radius limit value in the table. If the current radius is equal to the radius limit value, the dynamic factors of the next index entry are used.

| **Structure name** | **Index** |
|---|---|
| curve_dynamic_weighting[i] | 0 <= i <= 2 |

The following further conditions must be considered when defining table entries:

- The radius limit value in the table increases as the index values become larger.
- The weighting factor is limited to a minimum value of 1%. For ramp time, the minimum ramp time tr_min (P-AXIS-00201) must also be considered as the lower limit value.
- Weighting is restricted to an upward limit of 100% of the set axis value for velocity and acceleration (P-AXIS-00212, P-AXIS-00001, P-AXIS-00002 or P-AXIS-00011, P-AXIS-00012 . With ramp time tr_grenz (P-AXIS-00200) the weighting has no upper limit.

### 1.58.2.1 Radius limit (P-CHAN-00231)

<table>
<tr><th>P-CHAN-00231</th><th colspan="4">Radius limit (radius-dependent dynamic weighting)</th></tr>
<tr><td rowspan="4">Description</td><td colspan="4">This parameter defines the upper radius limit up to which the weighting factors of the current entries are effective under index i.<br>The following relationship applies:</td></tr>
<tr><td colspan="2"></td><td><b>i =0</b></td><td>i ≥1</td></tr>
<tr><td rowspan="2">Radius area limit</td><td>Lower</td><td>radius_limit[i-1]</td><td>radius_limit[i-1]</td></tr>
<tr><td>Upper</td><td>radius_limit[i]</td><td>radius_limit[i]</td></tr>
<tr><td>Parameter</td><td colspan="4">curve_dynamic_weighting[i].radius_limit</td></tr>
<tr><td>Data type</td><td colspan="4">UNS32</td></tr>
<tr><td>Data range</td><td colspan="4">0 < radius_limit < MAX(UNS32)</td></tr>
<tr><td>Dimension</td><td colspan="4">0.1µm</td></tr>
<tr><td>Default value</td><td colspan="4">0</td></tr>
</table>

| | |
|---|---|
| Remarks | |

### 1.58.2.2 Weighting factor for speed (P-CHAN-00232)

| **P-CHAN-00232** | **Weighting factor for velocity (radius-dependent dynamic weighting)** |
|---|---|
| Description | This parameter defines the weighting factor for maximum speed in index i. |
| Parameter | curve_dynamic_weighting[i].velocity_fact |
| Data type | UNS32 |
| Data range | 10 < velocity_fact < MAX(UNS32) |
| Dimension | 0.1% |
| Default value | 0 |
| Remarks | |

### 1.58.2.3 Weighting factor for acceleration (P-CHAN-00233)

| **P-CHAN-00233** | **Weighting factor for acceleration (radius-dependent dynamic weighting)** |
|---|---|
| Description | This parameter defines the weighting factor for acceleration in index i. |
| Parameter | curve_dynamic_weighting[i].acceleration_fact |
| Data type | UNS32 |
| Data range | 10 < acceleration_fact < MAX(UNS32) |
| Dimension | 0.1% |
| Default value | 0 |
| Remarks | |

### 1.58.2.4 Weighting factor for ramp time (P-CHAN-00234)

| **P-CHAN-00234** | **Weighting factor for ramp time (radius-dependent dynamic weighting)** |
|---|---|
| Description | This parameter defines the weighting factor for ramp time in index i. |
| Parameter | curve_dynamic_weighting[i].ramp_time_fact |
| Data type | UNS32 |
| Data range | 10 < ramp_time_fact < MAX(UNS32) |
| Dimension | 0.1% |
| Default value | 0 |
| Remarks | |

## 1.58.3 Example of a table

**Basis is the following table of dynamic weighting values with 3 rows:**

| **Index** | **Radius limit value**<br>**radius_limit [0.1µm]**<br>**(range limits)** | **Weighting factor**<br>**velocity_fact [0.1%]** | **Weighting factor**<br>**acceleration_fact [0.1%]** | **Weighting factor**<br>**ramp_time_fact [0.1%]** |
|---|---|---|---|---|
| 0 | 10000<br>(0-10000) | 100 | 10 | 3000 |
| 1 | 100000<br>(10000-100000) | 500 | 100 | 2000 |
| 2 | 1000000<br>(100000-1000000) | 1000 | 1000 | 1000 |

### Mapping of the weighting table on the structure of the channel parameters:

```
curve_dynamic_weighting_active 1 0: not active 1: active
#
curve_dynamic_weighting[0].radius_limit 10000 [0.1um]
curve_dynamic_weighting[0].velocity_fact 100 [0.1%]
curve_dynamic_weighting[0].acceleration_fact 10 [0.1%]
curve_dynamic_weighting[0].ramp_time_fact 3000 [0.1%]

curve_dynamic_weighting[1].radius_limit 100000 [0.1um]
curve_dynamic_weighting[1].velocity_fact 500 [0.1%]
curve_dynamic_weighting[1].acceleration_fact 100 [0.1%]
curve_dynamic_weighting[1].ramp_time_fact 2000 [0.1%]

curve_dynamic_weighting[2].radius_limit 1000000 [0.1um]
curve_dynamic_weighting[2].velocity_fact 1000 [0.1%]
curve_dynamic_weighting[2].acceleration_fact 1000 [0.1%]
curve_dynamic_weighting[2].ramp_time_fact 1000 [0.1%]
```

# 1.59 Special manual mode data (man_mode.*)

This structure does not contain axis-specific parameters for manual mode.

## 1.59.1 Settings for Cartesian manual mode (man_mode.vector_limit.*)

These parameters are used to set the dynamics for Cartesian manual mode on a non-Cartesian machine structure if the movements executed are to use active kinematic transformation. For example, the speed in manual mode must be higher than a movement executed on joint angle level on a robot with Cartesian movement in the X direction.

### 1.59.1.1 Dynamic data for linear axes

#### 1.59.1.1.1 Command speed of a linear axis in manual mode (P-CHAN-00195)

| **P-CHAN-00195** | **Command speed of a linear axis in manual operation mode** |
|---|---|
| Description | This parameter defines the Cartesian dynamics of a manual mode axis for an X, Y, Z movement. |
| Parameter | man_mode.vector_limit.v_max_pos |
| Data type | UNS32 |
| Data range | 1 < v_max_pos < MAX(UNS32) |
| Dimension | μm/s or 0.001°/s |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*man_mode.vector_limit.v_max_pos 100000 [μm/s]* |

#### 1.59.1.1.2 Command acceleration of a linear axis in manual mode (P-CHAN-00196)

| **P-CHAN-00196** | **Command acceleration of a linear axis in manual operation mode** |
|---|---|
| Description | This parameter defines the Cartesian dynamics of a manual mode axis for an X, Y, Z movement. |
| Parameter | man_mode.vector_limit.a_max_pos |
| Data type | UNS32 |
| Data range | 1 < a_max_pos < MAX(UNS32) |
| Dimension | mm/s² or °/s² |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*man_mode.vector_limit.a_max_pos 1000 [mm/s²]* |

#### 1.59.1.1.3 Ramp time of a linear axis in manual mode (P-CHAN-00197)

| **P-CHAN-00197** | **Ramp time of a linear axis in manual operation mode** |
|---|---|
| Description | This parameter defines the Cartesian dynamics of a manual mode axis for an X, Y, Z movement. |
| Parameter | man_mode.vector_limit.tr_pos |
| Data type | UNS32 |
| Data range | 1 < tr_pos < MAX(UNS32) |
| Dimension | μs |

| | |
|---|---|
| Default value | 0 |
| Remarks | Parameterisation example:<br>*man_mode.vector_limit.tr_pos 100000 [µs]* |

### 1.59.1.2 Dynamic data for orientation axes

#### 1.59.1.2.1 Command speed of an orientation axis in manual mode (P-CHAN-00198)

| **P-CHAN-00198** | **Command speed of an orientation axis in manual mode** |
|---|---|
| Description | This parameter defines the Cartesian dynamics of a manual mode axis with an A, B, C movement. |
| Parameter | man_mode.vector_limit.v_max_ori |
| Data type | UNS32 |
| Data range | 1 < v_max_ori < MAX(UNS32) |
| Dimension | µm/s or 0.001°/s |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*man_mode.vector_limit.v_max_ori 50000 [µm/s]* |

#### 1.59.1.2.2 Command acceleration of an orientation axis in manual mode (P-CHAN-00199)

| **P-CHAN-00199** | **Command acceleration of an orientation axis in manual mode** |
|---|---|
| Description | This parameter defines the Cartesian dynamics of a manual mode axis with an A, B, C movement. |
| Parameter | man_mode.vector_limit.a_max_ori |
| Data type | UNS32 |
| Data range | 1 < a_max_ori < MAX(UNS32) |
| Dimension | mm/s² or °/s² |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*man_mode.vector_limit.a_max_ori 500 [mm/s²]* |

#### 1.59.1.2.3 Ramp time of an orientation axis in manual mode (P-CHAN-00200)

| **P-CHAN-00200** | **Ramp time of an orientation axis in manual mode** |
|---|---|
| Description | This parameter defines the Cartesian dynamics of a manual mode axis with an A, B, C movement. |
| Parameter | man_mode.vector_limit.tr_ori |
| Data type | UNS32 |
| Data range | 1 < tr_ori < MAX(UNS32) |
| Dimension | µs |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*man_mode.vector_limit.tr_ori 100000 [µs]* |

#### 1.59.1.2.4 Command jerk of an orientation axis in manual mode (P-CHAN-00343)

| **P-CHAN-00343** | **Command jerk of an orientation axis in manual mode** |
|---|---|

| | |
|---|---|
| Description | This parameter defines the Cartesian dynamics of a manual mode axis with an A, B, C movement. |
| Parameter | man_mode.vector_limit.j_max_ori |
| Data type | UNS32 |
| Data range | 1 < j_max_ori < MAX(UNS32) |
| Dimension | mm/s³ or °/s³ |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*man_mode.vector_limit.j_max_ori     5000 [mm/s³]* |

# 1.60 Cycle programming settings

## 1.60.1 Modal effect of changes in the cycle (P-CHAN-00210)

| **P-CHAN-00210** | **Modal effect of changes in the cycle** |
|---|---|
| Description | The modal G functions, circle geometry data and the current feed rate which are active before a cycle call are restored when the cycle ends, i.e. changes in the cycle with respect to this data are not retained.<br>If the parameter 'cycle_changes_modal' is assigned the value 1, changes in the cycle with regard to this data are retained after the cycle ends (modal). The cycle then responds like a normal subroutine with regard to effectiveness. |
| Parameter | cycle_changes_modal |
| Data type | BOOLEAN |
| Data range | 0: Changes of modal G functions, circle data and the feed rate in the cycle are effective only up to the end of the cycle (default). *<br>1: Changes of modal G functions, circle data and the feed rate in the cycle are also effective after the end of the cycle (modal). |
| Dimension | ---- |
| Default value | 0 |
| Remarks | * At the end of the cycle, only the types of modal G functions are restored that can be programmed on their own (e.g. G00, G01, G90, G91 etc.). G functions programmed in the cycle retain their validity in combination with value specifications (e.g. G129) or axis names (e.g. G92, G98, G99, G100, G112, G130 etc.). |

## 1.60.2 Suppress NC blocks from cycles and M6 in trace data and display (P-CHAN-00211)

| **P-CHAN-00211** | **Suppress NC blocks from cycles and M6 in trace data and display** |
|---|---|
| Description | This parameter defines whether the NC blocks are to be suppressed or visible in the display in the running NC program or in single-block mode during execution of a cycle or of the M6 subroutine and whether they are to be logged in the trace data. When active display is suppressed, only the cycle call or subroutine call of M6 is displayed during this time. |
| Parameter | suppress_cycle_logging |
| Data type | BOOLEAN |
| Data range | 0: NC blocks from cycles or from the M6 subroutine are displayed and logged in the trace (default). Encrypted NC programs are logged in the trace by a separate key in an NC kernel for diagnosis.<br>1: No display or logging of NC blocks in the trace from cycles or from the M6 subroutine. |
| Dimension | ---- |

| | |
|---|---|
| Default value | 0 |
| Remarks | Suppressing NC blocks from the M6 subroutine is available as of Build V3.1.3066.02. |

In TwinCAT systems, all cycle rows are visible in the display in the basic setting.

## 1.60.3 Memory size for @P parameters (P-CHAN-00481)

| P-CHAN-00481 | Memory size for @P parameters |
|---|---|
| Description | This parameter extends the memory area for cycle transfer parameters (@P-Parameter).<br>The value is assigned as bytes. |
| Parameter | cycle_stack_memory |
| Data type | UNS32 |
| Data range | 0 <= P-CHAN-00481 < MAX_UNS32 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter available as of V3.1.3079.20 |

## 1.60.4 Automatic creation of @P parameters (P-CHAN-00463)

| P-CHAN-00463 | Automatic creation of @P parameters |
|---|---|
| Description | This parameter defines whether a @ parameter is automatically created in a cycle on receipt of a read access if it was not programmed as a transfer parameter in the cycle call. |
| Parameter | create_cycle_param_on_read |
| Data type | BOOLEAN |
| Data range | 0: No automatic creation of a @P parameter on receipt of a read access. Output of error with ID 20394<br>1: Automatic creation of a @P parameter on receipt of a read access (default): Any read accesses for check purposes increases the memory consumption and the cycle becomes more complex. |
| Dimension | ---- |
| Default value | 1 |
| Remarks | Parameter available as of V3.1.3079.20 |

## 1.60.5 HSC settings for cycles

### 1.60.5.1 NC filename to deselect HSC settings (P-CHAN-00470)

| P-CHAN-00470 | NC filename to deselect HSC settings |
|---|---|
| Description | The NC filename specified in this parameter is called in the SysHscSettings cycle when the HSC settings are deselected. |
| Parameter | hscs.deselect.prog |
| Data type | STRING |
| Data range | |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string. |

### 1.60.5.2 Default machining tolerance to deselect HSC settings (P-CHAN-00471)

| P-CHAN-00471 | Default machining tolerance to deselect HSC settings |
|---|---|
| Description | The value specified in this parameter for the machining tolerance is set in the SysHscSettings cycle when the HSC settings are deselected. |
| Parameter | hscs.deselect.tolerance |

| | |
|---|---|
| Data type | SGN32 |
| Data range | MIN(SGN32) <= P-CHAN-00471 <= MAX(SGN32) |
| Dimension | [0.1 µm] |
| Default value | 100 |
| Remarks | |

### 1.60.5.3 NC filename for HSC settings - roughing (P-CHAN-00472)

| **P-CHAN-00472** | **NC filename for HSC settings - roughing** |
|---|---|
| Description | The NC filename specified in this parameter is called in the SysHscSettings cycle when the HSC settings are selected for roughing. |
| Parameter | hscs.prefinish.prog |
| Data type | STRING |
| Data range | |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string. |

### 1.60.5.4 Default machining tolerance for HSC settings - roughing (P-CHAN-00473)

| **P-CHAN-00473** | **Default machining tolerance for HSC settings - roughing** |
|---|---|
| Description | The value specified in this parameter for the machining tolerance is set in the SysHscSettings cycle when the HSC settings are selected for roughing. |
| Parameter | hscs.rough.tolerance |
| Data type | SGN32 |
| Data range | MIN(SGN32) <= P-CHAN-00473 <= MAX(SGN32) |
| Dimension | [0.1 µm] |
| Default value | 1000 |
| Remarks | |

### 1.60.5.5 NC filename for HSC settings- Pre-finishing (P-CHAN-00474)

| **P-CHAN-00474** | **NC filename for HSC settings - pre-finishing** |
|---|---|
| Description | The NC filename specified in this parameter is called in the SysHscSettings cycle when the HSC settings are selected for pre-finishing. |
| Parameter | hscs.prefinish.prog |
| Data type | STRING |
| Data range | |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string. |

### 1.60.5.6 Default machining tolerance for HSC settings - Pre-finishing (P-CHAN-00475)

| **P-CHAN-00475** | **Default machining tolerance for HSC settings - Pre-finishing** |
|---|---|
| Description | The value specified in this parameter for the machining tolerance is set in the SysHscSettings cycle when the HSC settings are selected for pre-finishing. |
| Parameter | hscs.prefinish.tolerance |
| Data type | SGN32 |

| | |
|---|---|
| Data range | MIN(SGN32) <= P-CHAN-00475 <= MAX(SGN32) |
| Dimension | [0.1 µm] |
| Default value | 500 |
| Remarks | |

### 1.60.5.7 NC filename for HSC settings - Finishing (P-CHAN-00476)

| **P-CHAN-00476** | **NC filename for HSC settings - Finishing** |
|---|---|
| Description | The NC filename specified in this parameter is called in the SysHscSettings cycle when the HSC settings are selected for finishing. |
| Parameter | hscs.finish.prog |
| Data type | STRING |
| Data range | |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string. |

### 1.60.5.8 Default machining tolerance for HSC settings - Finishing (P-CHAN-00477)

| **P-CHAN-00477** | **Default machining tolerance for HSC settings - Finishing** |
|---|---|
| Description | The value specified in this parameter for the machining tolerance is set in the SysHscSettings cycle when the HSC settings are selected for finishing. |
| Parameter | hscs.finish.tolerance |
| Data type | SGN32 |
| Data range | MIN(SGN32) <= P-CHAN-00477 <= MAX(SGN32) |
| Dimension | [0.1 µm] |
| Default value | 100 |
| Remarks | |

## 1.61 Settings for edge machining (edge_machining.*)

This functionality is available as of Release **V2.11.2009.12**. It is enabled in the start-up list ([STUP]) for each CNC channel as follows (as an example):

```
configuration.channel[<i>].path_preparation.function FCT_DEFAULT |
FCT_EMF
```

Depending on the machining technology it may be necessary to control the machining process especially on sharp contours (edges). In the case of a sharp edge (defined by the angle difference between two contour elements), the path velocity at the edge is modified depending on pre-defined parameters. In addition, three signals are generated, derived from path velocity. These signals are supplied to the PLC interface (see HLI documentation, Status information of a channel -> Edge functions -> Signal_1, Signal_2, Signal_3). Linear or circular blocks can be programmed as contour elements. No check is made here whether they are inner or outer contours. If the functions ‘Insert chamfers and roundings (G301/G302)’ or tool radius compensation (G41/G42) are active, additional contour elements can be generated. They then result in a different bend angle than exists between the two original contour elements.

**The following parameters are required to set the edge machining function:**

```
# Parameterise edge machining function
# =====================================
edge_machining.enable 1
edge_machining.angle_limit 150000 [0.0001°(degrees)]
edge_machining.pre_dist 100000 [0.1µm]
edge_machining.pre_feed 16666 [µm /sec]
edge_machining.wait_time 10000 [[1 µs]]
edge_machining.post_dist 200000 [0.1µm]
edge_machining.post_feed 333333 [µm/sec]
edge_machining.disable_feed_adaption 0
edge_machining.mode 0
#
```

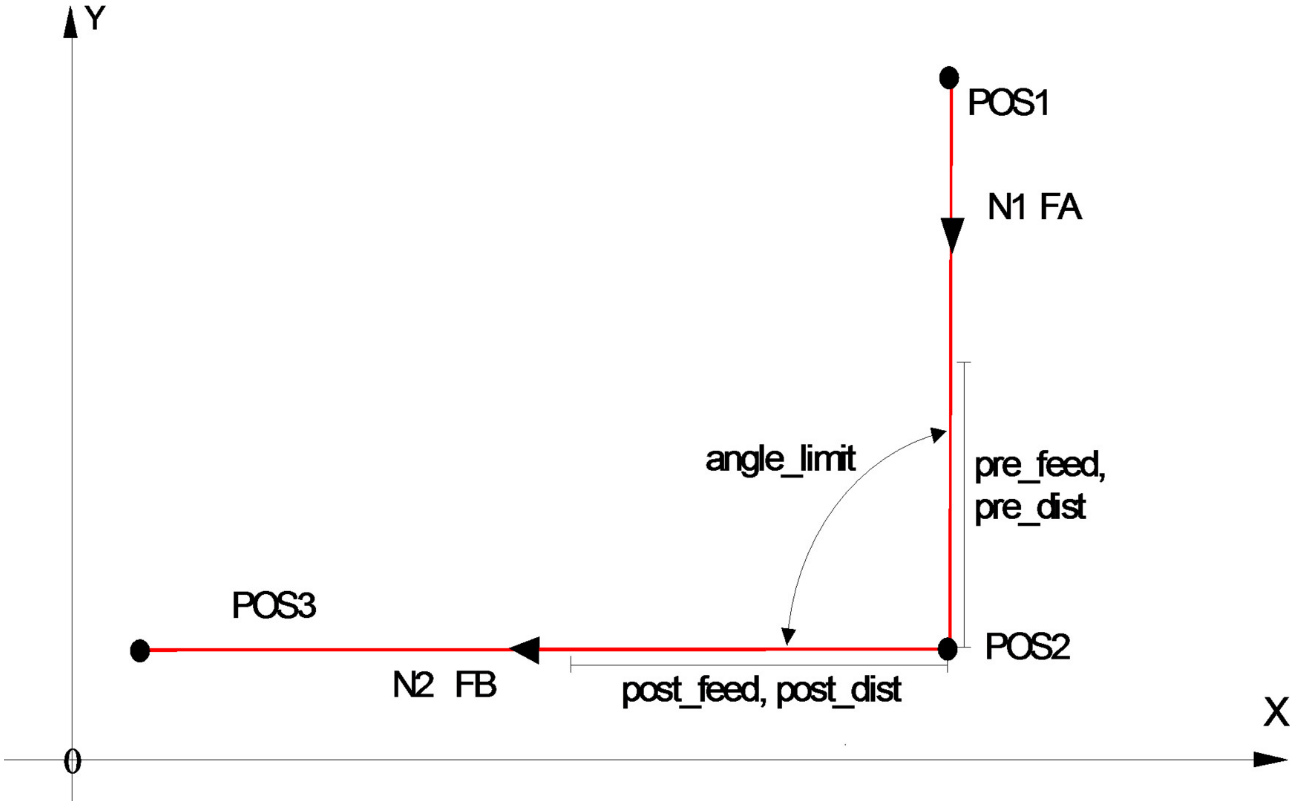


Fig. 41: Edge between two contour elements

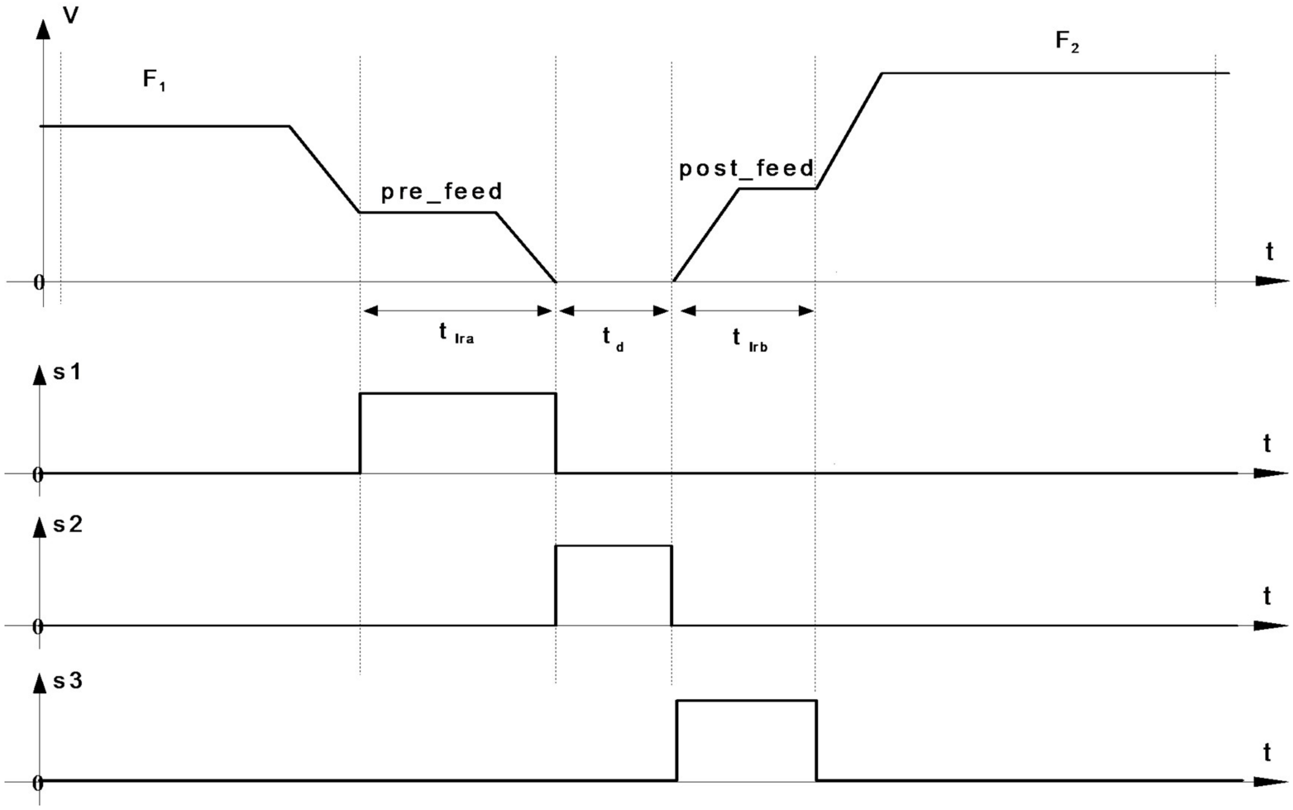


Fig. 42: Timing diagram of s1, s3 signals

### 1.61.1 Enable / disable (P-CHAN-00220)

| P-CHAN-00220 | Enable / disable the edge machining function |
|---|---|
| Description | Parameter for activating/deactivating the edge machining function. |
| Parameter | edge_machining.enable |
| Data type | BOOLEAN |
| Data range | 0: Edge machining function is inactive (default).<br>1: Edge machining function is active. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | *edge_machining.f_enable (old syntax up to V2.11.2022.13)* |

### 1.61.2 Critical bend angle (P-CHAN-00221)

| P-CHAN-00221 | Critical bend angle (edge machining function) |
|---|---|
| Description | If the bend angle between two contour elements falls below the critical angle, the special edge function is executed. Before the distance *pre_dist* is reached, the motion decelerates in the first contour element to the feed rate *pre_feed*. A stop is executed between the two blocks and a waiting time *wait_time* is inserted. The motion is then executed at the feed rate *post_feed* and after it exceeds *post_dist*, the motion continues at the originally programmed feed rate. |
| Parameter | edge_machining.angle_limit |
| Data type | UNS32 |
| Data range | 0 ... .1800000 |
| Dimension | 0.0001° |
| Default value | 0 |
| Remarks | |

### 1.61.3 Distance before the edge (P-CHAN-00222)

| P-CHAN-00222 | Distance before the edge (edge machining function) |
|---|---|
| Description | The logical signal S1 0 ->1 is generated if the distance before a sharp edge is undershot. The value may be limited internally to half the block length of the programmed contour element. |
| Parameter | edge_machining.pre_dist |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | |

### 1.61.4 Feed before edge (P-CHAN-00223)

| P-CHAN-00223 | Feed before edge (edge machining function) |
|---|---|
| Description | After undershooting the distance *pre_dist* at the feed rate *pre_feed*, the CNC moves to the stop in the edge. |
| Parameter | edge_machining.pre_feed |
| Data type | UNS32 |
| Data range | 1 ... MAX(UNS32) |
| Dimension | µm/s |
| Default value | 0 |

| | |
|---|---|
| Remarks | An externally commanded feed rate from the PLC ([HLI//section Control commands of a channel]) has a higher priority. The feed rate defined by P-CHAN-00223 then has no effect. |

## 1.61.5 Waiting time in edge (P-CHAN-00224)

| **P-CHAN-00224** | **Waiting time in edge (edge machining function)** |
|---|---|
| Description | The logical signal S2 is generated for the duration of *wait_time*. |
| Parameter | edge_machining.wait_time |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | µs |
| Default value | 0 |
| Remarks | |

## 1.61.6 Distance after the edge (P-CHAN-00225)

| **P-CHAN-00225** | **Distance after the edge (edge machining function)** |
|---|---|
| Description | The logical signal S3 0 ->1 is generated after a sharp edge until the distance *post_dist* is reached. The value may be limited internally to half the block length of the programmed contour element. |
| Parameter | edge_machining.post_dist |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | |

## 1.61.7 Feed rate after the edge (P-CHAN-00226)

| **P-CHAN-00226** | **Feed rate after the edge (edge machining function)** |
|---|---|
| Description | The CNC moves at this feed rate until the distance *post_dist* is reached. The motion then continues at the originally programmed feed rate. |
| Parameter | edge_machining.post_feed |
| Data type | UNS32 |
| Data range | 1 ... MAX(UNS32) |
| Dimension | µm/s |
| Default value | 0 |
| Remarks | An externally commanded feed rate from the PLC ([HLI//section Control commands of a channel]) has a higher priority. The feed rate defined by P-CHAN-00226 then has no effect. |

## 1.61.8 Switching feed rate adjustment (P-CHAN-00300)

| **P-CHAN-00300** | **Switching feed rate adjustment (edge machining function)** |
|---|---|
| Description | This parameter controls the response of feed adaption in and after an edge. Depending on the value setting, deceleration takes place only on approach of the edge and then re-acceleration takes place; the feed parameters P-CHAN-00223 [▶ 250] (*pre_feed)* and P-CHAN-00226 [▶ 251] (*post_feed*) have no effect. In addition, the range can be deactivated by signal S2 (dwell time). |
| Parameter | edge_machining.disable_feed_adaption |

| | |
|---|---|
| Data type | UNS16 |
| Data range | 0: Feed adaption is active; dwell time is active with signal S2.<br>1: Feed adaption is inactive; the range is inactive with signal S2.<br>2: Feed adaption is inactive; the range is active with signal S2. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

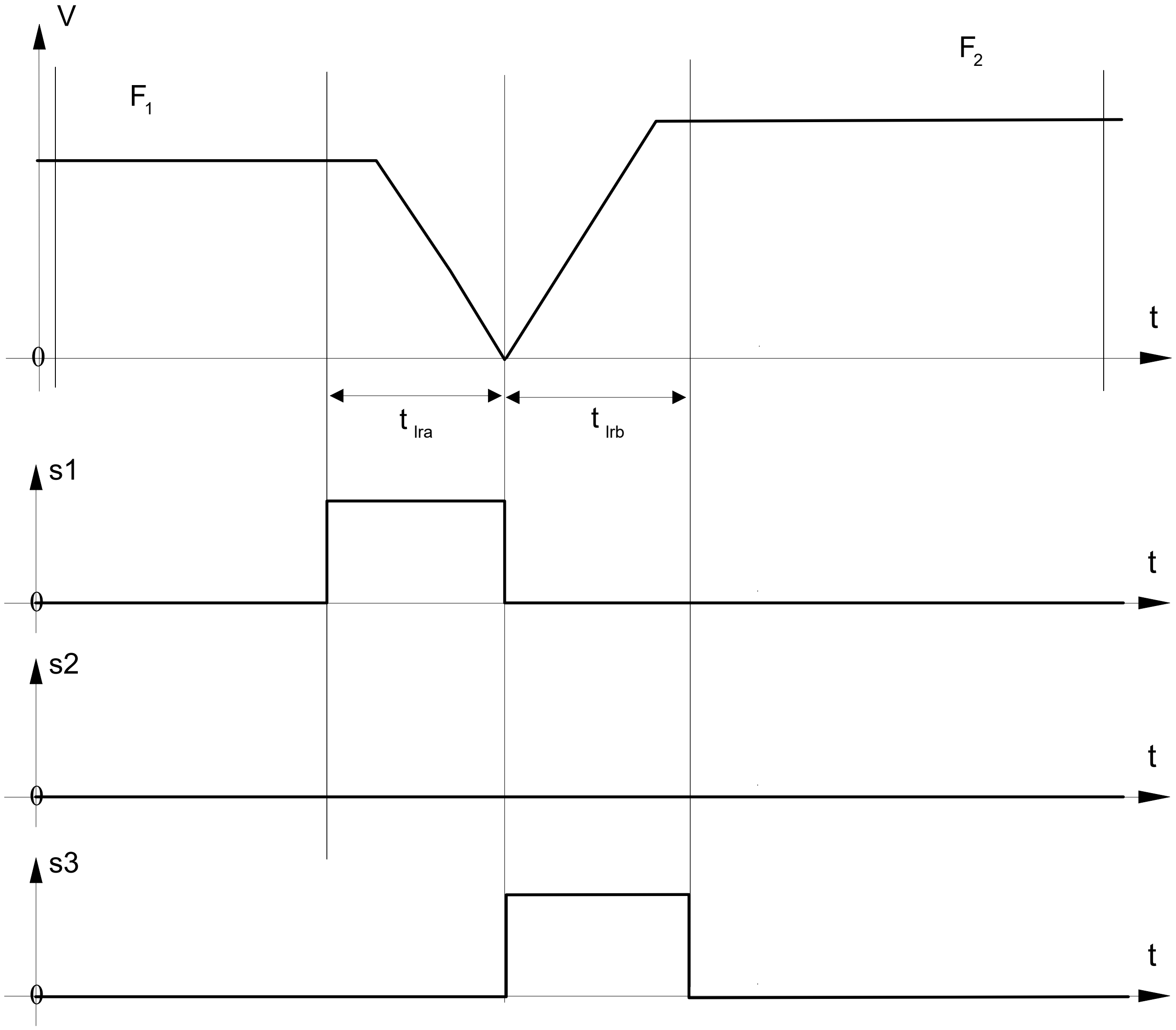


Fig. 43: Timing diagram of s1..s3 signals when P-CHAN-00300 = 1

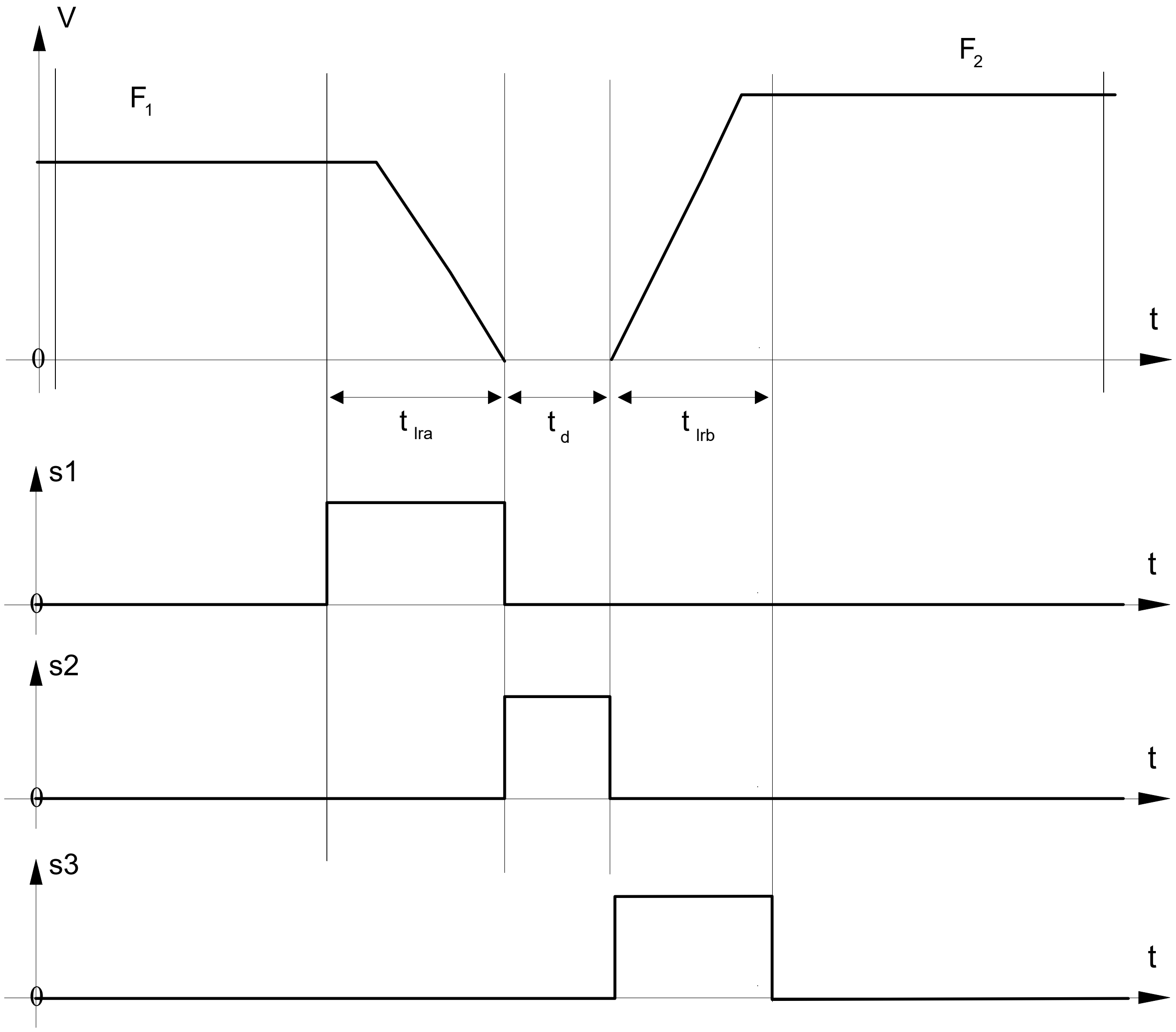


Fig. 44: Timing diagram of s1..s3 signals when P-CHAN-00300 = 2

### 1.61.9 Edge detection mode (P-CHAN-00301)

| P-CHAN-00301 | Edge detection mode |
|---|---|
| Description | Parameter for setting 3D (default) or 2D edge detection. When 2D edge detection is active, only motion in the XY place is considered. |
| Parameter | edge_machining.mode 0 |
| Data type | UNS16 |
| Data range | 0: 3D edge detection is active (default).<br>1: 2D edge detection is active. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.62 Decoder block ahead limiting

The following parameters can define how far forward the decoder may be ahead of the interpolator to process NC blocks. For more information please refer to [FCT-C24].

Only one type of decoder block ahead limiting function may be active. This is active immediately after program start and it remains selected until main program end M02/M30.

### 1.62.1 Block ahead limiting via number of blocks (P-CHAN-00216)

| P-CHAN-00216 | Select decoder block ahead limiting via number of blocks |
|---|---|
| Description | Parameter for activating and configuring the decoder block ahead limiting function based on NC blocks by setting a value unequal to zero. This value defines the maximum number of (relevant) NC blocks (rows) by which the decoder block ahead function may be ahead of the interpolator. |
| Parameter | max_nc_blocks_ahead |
| Data type | UNS32 |
| Data range | 0: No decoder block ahead limiting (default).<br>>0: After program start, decoder block ahead limiting function is immediately active with the number of NC blocks and this function remains selected until main program end M02/M30. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Only one type of decoder block ahead limiting function may be active. This is active immediately after program start and it remains selected until main program end M02/M30. |

### 1.62.2 Block ahead limiting via number of NC motion blocks (P-CHAN-00246)

| P-CHAN-00246 | Selecting decoder ahead limiting via number of NC motion blocks |
|---|---|
| Description | Parameter for activating and configuring the decoder block ahead limiting function based on NC blocks by setting a value unequal to zero. The value defines the maximum number of pure NC motion blocks by which the decoder may be ahead of the interpolator. |
| Parameter | max_motion_blocks_ahead |
| Data type | UNS32 |
| Data range | 0: No decoder block ahead limiting (default). |

| | |
|---|---|
| | >0: After program start the decoder block ahead limiting function is immediately active with the defined number of NC motion blocks and remains selected until main program end M02/M30. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Only one type of decoder block ahead limiting function may be active. This is active immediately after program start and it remains selected until main program end M02/M30. |

## 1.62.3 Time-based block ahead limiting (P-CHAN-00269)

| **P-CHAN-00269** | **Selecting time-based decoder block ahead limiting** |
|---|---|
| Description | This parameter activates and configures the time-based limiting of the decoder ahead function by setting a value unequal to zero. The value defines the maximum time in microseconds by which the decoder may be ahead of the interpolator. |
| Parameter | max_time_ahead |
| Data type | REAL64 |
| Data range | 0: No decoder block ahead limiting (default).<br>>0: The time-based decoder block ahead limiting function is active after program start at once and remains selected until main program end M30. |
| Dimension | µs |
| Default value | 0 |
| Remarks | Only one type of decoder block ahead limiting function may be active. This is active immediately after program start and it remains selected until main program end M02/M30. |

## 1.62.4 Block ahead limiting in protected mode (P-CHAN-00270)

| **P-CHAN-00270** | **Block ahead limiting in protected mode** |
|---|---|
| Description | To ensure a stable process, the block ahead limiting function can be executed in a special mode. In this mode the function is only effective if the block supply to the interpolator and path dynamics are not jeopardised. |
| Parameter | dec_max_ahead_protected |
| Data type | STRING |
| Data range | NONE: No monitoring of block ahead limiting (default).<br>ACTIVE: Block ahead limiting is monitored.<br>This function is optional for block-based ahead limiting.<br>This function is always ACTIVE when the time-based ahead limiting function is selected. |
| Dimension | ---- |
| Default value | NONE |
| Remarks | |

## 1.62.5 Selection/deselection of chamfers and roundings function (P-CHAN-00273)

| **P-CHAN-00273** | **Select/deselect the chamfers and roundings function** |
|---|---|
| Description | This parameter can disable the chamfers and roundings function (G301/G302). It is used to optimise the block ahead limiting function since it reduces the buffering effect of the NC channel. If the parameter is set to 1, the functions G301 and G302 are no longer usable. |
| Parameter | disable_chamfers_roundings |
| Data type | BOOLEAN |

| | |
|---|---|
| Data range | 0: The chamfers and roundings function is enabled and available (default).<br>1: The chamfers and roundings function is disabled. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.62.6 Time ahead calculation based on mean feed velocity (P-CHAN-00428)

| P-CHAN-00428 | Deactivate time ahead calculation based on mean feed velocity |
|---|---|
| Description | In order to estimate the motion time for block ahead limiting realistically, the interpolation supplies a mean feed velocity (average velocity) of the next motion blocks to be executed.<br>In default mode the mean feed velocity is always included in the calculation of time ahead. This function can be deactivated by this channel parameter to run in programs and for diagnostic purposes. |
| Parameter | calc_average_feed_ahead |
| Data type | BOOLEAN |
| Data range | 0: Time ahead calculation without mean feed velocity.<br>1: Time ahead calculation with mean feed velocity (default). |
| Dimension | ---- |
| Default value | TRUE |
| Remarks | This parameter is available as of CNC Build V2.11.2808.05 and higher.<br>Only use for tests or diagnostic purposes! |

## 1.63 Exact stop with rapid traverse (P-CHAN-00235)

| P-CHAN-00235 | Exact stop after every rapid traverse |
|---|---|
| Description | If this parameter is set to 1, an exact stop (G60) is implicitly executed after every rapid traverse block (G00). Active smoothing (G61, G261) is then suppressed for this NC block |
| Parameter | exact_stop_after_g00 |
| Data type | BOOLEAN |
| Data range | 0: Rapid traversing movement without exact stop (G60) (default).<br>1: Rapid traverse motion followed by automatic exact stop (G60). |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.64 Check exact stop with additive movements (P-CHAN-00486)

| P-CHAN-00486 | Check exact stop with additive movements |
|---|---|
| Description | With exact stop G60, a check is made that the axis is located within a specified position window (P-AXIS-00236) i.e.<br>the value of P-AXIS-00236 is smaller than or equal to the value of the position lag.<br>Monitoring is performed in the position controller if the axis is not interpolated.<br>If the axis is moved by additive commands, such as axis couplings or the additional position interface in the position controller, the axis cannot reach the position window and program processing stops at the exact stop check G60.<br>This parameter extracts the additive movements from the exact stop check. If the axis is moved by one of the suppressed motion options, the exact stop condition is fulfilled as soon as the axis is no longer interpolated by the path or spindle interpolator.<br>The movement specifications can also be combined, for example:<br>`in_position_ignored_movement AXIS_COUPLING \| ADD_CMD_VALUES` |
| Parameter | in_position_ignored_movement |

| | |
|---|---|
| Data type | STRING |
| Data range | NONE: The exact stop check considers all axis movements (default)<br>AXIS_COUPLING: Movements (see [FCT-A9]) by axis couplings are not considered<br>ADD_CMD_VALUES: Movements by the position controller additional interface are not considered (see [HLI//External commanding of an axis])<br>ALL: All additive movements are not considered. |
| Dimension | ---- |
| Default value | NONE |
| Remarks | This parameter is available as of CNC Build V3.1.3079.29, |

# 1.65 Settings for lifting

## 1.65.1 Minimum path length for lift movements (P-CHAN-00244)

| **P-CHAN-00244** | **Minimum path length for lift movements** |
|---|---|
| Description | This parameter defines a minimum path distance for lift movement. If the main axis motion is shorter than the parameter value, no lift movement is executed. |
| Parameter | lift_min_dist |
| Data type | UNS32 |
| Data range | 0: Inactive (default).<br>1: Lift movements are suppressed if the main path motion is below the limit value. |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | |

## 1.65.2 Lifting an axis with time-based calculation (P-CHAN-00345)

| **P-CHAN-00345** | **Switch-over to time-based calculation when an axis is lifted** |
|---|---|
| Description | When an axis is lifted (see [FCT-A11]), it can be lifted or lowered automatically independent of the path motion. The CNC limits the maximum lift height so that the axis can reach the target point of the lowering movement and not influence the path motion. Normally this takes place during path preparation with a path-based coupling of the axis to the main motion path. Instead the 'enable_time_base_lift' parameter can enable a time-based consideration in the real-time GEO task of the controller. As a result, greater lifting height can be reached afterwards. However, time-based coupling requires considerably more computing power in the real-time task of the controller. The HSC slope profile and the time-based approach cannot be used at the same time. |
| Parameter | enable_time_based_lift |
| Data type | BOOLEAN |
| Data range | 0: Path-based approach (default).<br>1: Time-based approach. |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | The time-based approach must also be included in the configuration data of the path preparation and interpolation function in the controller. Here, set the key word FCT_LIFT_UP_TIME in the parameters P-CHAN-00600 [▶ 54] and P-CHAN-00650 [▶ 61] (alternatively: P-STUP-00060 and P-STUP-00070).<br>Parameterisation example with P-CHAN-00600 [▶ 54] / P-CHAN-00650 [▶ 61]<br>`configuration.path_preparation.function FCT_DEFAULT \| FCT_LIFT_UP_TIME`<br>`.interpolator.function`<br>`FCT_DEFAULT \| FCT_LIFT_UP_TIME`<br>Alternatively, the function can be parameterised in the start-up list (P-STUP-00060 / P-STUP-00070).<br>Example of the 1st CNC channel:<br>`configuration.channel[0].path_preparation.function.`<br>`FCT_DEFAULT \| FCT_LIFT_UP_TIME`<br>`configuration.channel[0].interpolator.function.`<br>`FCT_DEFAULT \| FCT_LIFT_UP_TIME` |

# 1.66 Settings for robotics

## 1.66.1 Orientation programming (ori.*)

With 5-axis machining with active complete kinematic transformation, the Cartesian coordinates are programmed either in point/Euler angle representation or in a point/direction vector representation.

The orientation is specified by means of address letters A/B/C or also I/J/K.

In basic setting, the point/Euler angle representation is active in the NC channel (orientation specified via rotation angles A, B, C) where typically 2 rotation angles are sufficient for 5-axis machining. All 3 rotation angles are typically used only for machines with six axes (e.g. robots) to define the manual coordinate system.

In CAD/CAM systems, the coordinates of a workpiece are usually stored in vector representation and the axis coordinates are represented in vector notation in automatically generated NC programs.

To permit direct processing of these NC programs, the following channel parameters can be used to set whether the values programmed with A,B,C or I,J,K are interpreted as angle values or as vector components.

Alternatively, the default setting in the channel parameters in the NC program can be modified at any time by means of a corresponding NC command [PROG//Command #ORI].

### 1.66.1.1 Type of orientation representation (P-CHAN-00177)

| **P-CHAN-00177** | **Type of orientation representation** | |
|---|---|---|
| Description | The operator uses this parameter to define whether the values programmed in the NC program with A, B, C or I, J, K are read as coordinate or angle values when complete kinematic transformation is active, or whether they are interpreted as corresponding vector components.<br>Alternatively, the type of orientation display is defined in the NC program by #ORI MODE[..]. | |
| Parameter | ori.mode | |
| Data type | STRING | |
| Data range | ANGLE: | A, B, C or I, J, K represent coordinate or angle values (default). |
| | VECTOR_ABC: | A, B, C represent vector components. Accordingly, 5- or 6-axis machining then takes place depending on the activated transformation. |
| | VECTOR_IJK: | I, J, K are vector components (only permitted with 6-axis kinematics, e.g. robots). |
| Dimension | ---- | |
| Default value | ANGLE | |
| Remarks | Note that the I, J, K coordinate values are used for circle programming in compliance with DIN 66025. This means that circle programming in combination with vector component programming via I, J, K is not permissible. | |

### 1.66.1.2 Index of fixed rotation axis (P-CHAN-00178)

| **P-CHAN-00178** | **Index of fixed rotary axis to program Euler angles** |
|---|---|

| | |
|---|---|
| Description | This parameter is only used for point-vector programming in combination with 6-axis kinematics (robots).<br>The tool's direction vector is defined via the three vector components (e.g. "VECTOR_ABC"). These vector components result in two Euler rotation angles.<br>If the orientation can be defined by more than two angles (e.g. robots), the free third Euler angle must be specified by this parameter for use of vector representation.<br>This index results from considering the sequence of axes which define the position and manual orientation of the machine (e.g. robot). |
| Parameter | ori.fixed_axis_index |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter may only be used exclusively with P-CHAN-00436 [▸ 261] . |

Example: Kinematic structure with 3 main axes X, Y, Z and 3 rotary axes A, B, C (robot):

The rotation convention is: A about Z -> B about Y -> C about X

| **Axis name** | **index** | **Axis type** |
|---|---|---|
| : | : | : |
| A | 3 | 1. rotary axis |
| B | 4 | 2. rotary axis |
| C | 5 | 3rd rotary axis |

According to the rotation convention above, the last axis (C) rotates about the X axis. If this axis is simultaneously the tool axis, a rotation about this axis only produces a change in the position of the manual coordinate system.

For example, if the rotation axis (C) is to be the fixed axis, the value 5 must be assigned to 'fixed_axis_index'.

### 1.66.1.3 Plane of parallel tool axis (P-CHAN-00436)

| **P-CHAN-00436** | **Plane parallel to tool axis plane** |
|---|---|
| Description | This parameter is only used for point-vector programming in combination with 6-axis kinematics (robots).<br>This parameter defines an identifier for the basic plane (YZ, ZX) that contains either the Z or the Y tool axis. The third angle is then determined by an internal controller calculation so that the selected tool axis at the target point is oriented parallel to the defined basic plane. |
| Parameter | ori.tool_ax_in_plane |
| Data type | UNS32 |
| Data range | 0: Not assigned<br>1: Tool axis Z lies parallel to the ZX plane of the active basic CS or MCS.<br>2: Tool axis Y lies parallel to the YZ plane of the active basic CS or MCS. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter may only be used exclusively with P-CHAN-00178 [▸ 260] . |

## 1.66.2 Alternative programming of axis-specific movements in robotics (P-CHAN-00253)

| **P-CHAN-00253** | **Alternative programming of axis-specific movements in robotics** |
|---|---|
| Description | Regardless of the type of programmed movement, axes in the NC channel are always programmed with the same configured name. In robotics and handling, however, it is also common to alternatively program the (rotary) axis movements with A1, A2, etc. and the Cartesian movements e.g. with X, Y, Z, A, B, C.<br>This option is enabled by setting the 'use_alias_name_in_ax_list' parameter. |
| Parameter | use_alias_name_in_ax_list |
| Data type | BOOLEAN |
| Data range | 0: Axis-specific and Cartesian motions must always be programmed with the same axis names as configured in the channel parameter list (P-CHAN-00006 [▸ 139], e.g. X, Y, Z, A, B, C) (default).<br>1: Alternatively, axis-specific motions can be programmed with A1 - A32 when transformation (#TRAFO OFF, #MCS ON) is disabled.<br>Condition:<br>• A1 - A32 must be entered as default names in the corresponding axis lists (P-AXIS-00297).<br>• Only A1, A2 to A32 are explicitly permissible as names.<br>• The equals sign to assign the position or the angle must follow directly after the axis name without any blanks.<br>• Use of A1 - A32 is only permitted if transformation is disabled.<br><br>For compatibility reasons, the axes can continue to be programmed with the same axis names (P-CHAN-00006 [▸ 139], e.g. X, Y, Z, A, B, C) as in the channel parameter list.<br>The combined use (mixed mode) of axis names within the NC row is also possible but is not advisable for reasons of clarity.<br><br>EXAMPLE:<br>Nxx #TRAFO ON<br>Nxx X.. Y.. Z.. A.. B.. C..<br>..<br>Nxx **#MCS ON**<br>Nxx A1=.. A2=.. A3=.. A4=.. A5=.. A6=.. ; alias names when transformation is inactive<br>...<br>Nxx #MCS OFF<br>Nxx X.. Y.. Z.. A.. B.. C..<br>... |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.66.3 Suppress an active kinematic transformation at G0 (P-CHAN-00423)

| **P-CHAN-00423** | **Suppress an active kinematic transformation at G0** |
|---|---|

| | |
|---|---|
| Description | This parameter sets a response to permit rapid movements at PTP (G0). This is typically required for robot applications. |
| Parameter | suppress_trafo_in_g0_blocks |
| Data type | BOOLEAN |
| Data range | 0: When kinematic transformation is active, G0 blocks are normally moved as rapid traverse blocks with linear interpolation in the PCS (default).<br>1: An active transformation at G0 is temporarily suppressed and interpolated in the ACS (PTP). |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.67 Limiting the path length of motion block (P-CHAN-00457)

| **P-CHAN-00457** | **Limiting the path length of motion blocks** |
|---|---|
| Description | This parameter limits the path length of motion blocks.<br>Motion blocks with a path length greater than the value entered for the parameter are segmented into new motion blocks. The motions blocks then have the maximum specified path length of P-CHAN-00457.<br>Segmentation only takes place after path-changing functions, e.g.I TRC or Polynomial contouring.<br>The parameter is limited:<br>• if parameter is greater than 0 and less than 10000 (=1mm), P-CHAN-00457 is set to 10000<br>• if parameter is greater than 2000000 (=200mm), P-CHAN-00457 is set to 2000000<br>• P-CHAN-00457= 0, no effect |
| Parameter | post_segmentation_length |
| Data type | UNS32 |
| Data range | 0 - 2000000 |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | The main application is in robotics to achieve higher velocities and accelerations. |

## 1.68 Handling of polynomial blocks of zero length (P-CHAN-00254)

| **P-CHAN-00254** | **Handling polynomial blocks of zero length** |
|---|---|
| Description | When motion blocks have zero length when used with polynomial smoothing, they are rejected by default. If these blocks should be output anyway, for example for display purposes, this parameter must be set to 1. |
| Parameter | output_block_length_zero |
| Data type | BOOLEAN |
| Data range | 0: Motion blocks of zero length are rejected (default).<br>1: Motion blocks of zero length are output. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

# 1.69 Settings for profile tube processing (tube_profile.*)

In section machining, there may be deviations in the material properties of section roundings (e.g. wall thickness). The transition to and from section roundings can be signalled by M/H functions. As a result, the process can be influenced by the PLC. The parameters described below can be used for configure process control on profile roundings.

For further information see the documentation [FCT-M5].

## 1.69.1 Type definition of technological functions (P-CHAN-00251)

| **P-CHAN-00251** | **Type definition of technological functions for tube profile machining** |
|---|---|
| Description | This parameter defines the type of the technological control functions used (M or H function) |
| Parameter | tube_profile.techno_type |
| Data type | SGN16 |
| Data range | 0: M numbers<br>1: H numbers |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.69.2 Technological functions to display rounding transitions when exiting (P-CHAN-00250)

| **P-CHAN-00250** | **Technological functions to display rounding transitions when exiting for tube profile machining** |
|---|---|
| Description | This parameter defines the M/H number on exiting the section rounding. |
| Parameter | tube_profile.techno_nr_rnd_off |
| Data type | SGN16 |
| Data range | -1 ... 999 (application-specific) |
| Dimension | ---- |
| Default value | -1 (not used) |
| Remarks | The M/H functions are always of the MOS type.<br>To activate the function, the M/H numbers of the two parameters must be = 0.<br>M/H numbers may not be already assigned by use in P-CHAN-00041 [▸ 65] (m_synch[...]) or P-CHAN-00027 [▸ 83] (h_synch[...]).<br>Parameterisation example:<br>*tube_profile.techno_type* 1 *Use of H numbers*<br>*tube_profile.techno_nr_rnd_on* 300 *H number, entering into rounding*<br>*tube_profile.techno_nr_rnd_off* 400 *H number, exiting from rounding* |

### 1.69.3 Technological functions to signal rounding transitions on engaging (P-CHAN-00249)

| **P-CHAN-00249** | **Technological functions to signal rounding transitions on engaging with tube profile machining** |
|---|---|
| Description | This parameter defines the M/H number on entering the section rounding. |
| Parameter | tube_profile.techno_nr_rnd_on |
| Data type | SGN16 |
| Data range | -1 ... 999 (application-specific) |
| Dimension | ---- |
| Default value | -1 (not used) |
| Remarks | The M/H functions are always of the MOS type.<br>To activate the function, the M/H numbers of the two parameters must be >= 0.<br>M/H numbers may not be already assigned by use in P-CHAN-00041 [▶ 65] (m_synch[...]) or P-CHAN-00027 [▶ 83] (h_synch[...]).<br>Parameterisation example:<br>*tube_profile.techno_type* *1* *Use of H numbers*<br>*tube_profile.techno_nr_rnd_on* *300* *H number, engaging in rounding*<br>*tube_profile.techno_nr_rnd_off* *400* *H number, exiting from rounding* |

## 1.70 Optimised polynomial smoothing (P-CHAN-00259)

| **P-CHAN-00259** | **Optimised polynomial smoothing for transitions in circular blocks.** |
|---|---|
| Description | If polynomial contouring of type 4 (#CONTOUR MODE [DEV...] ) is used, transitions with circle blocks can be traversed faster by additionally setting this parameter.<br>If jerk monitoring in polynomial blocks is selected )P-CHAN-00110 [▶ 198], check_jerk_on_poly_path is 1 or 2), the optimisation smooths the curvature and increases the feed rate. Particular advantages are obtained with tangential transitions in circular blocks. |
| Parameter | opt_contour_mode |
| Data type | UNS32 |
| Data range | 0, 1 where:<br>0: Inactive (default).<br>1: Optimised polynomial smoothing active. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

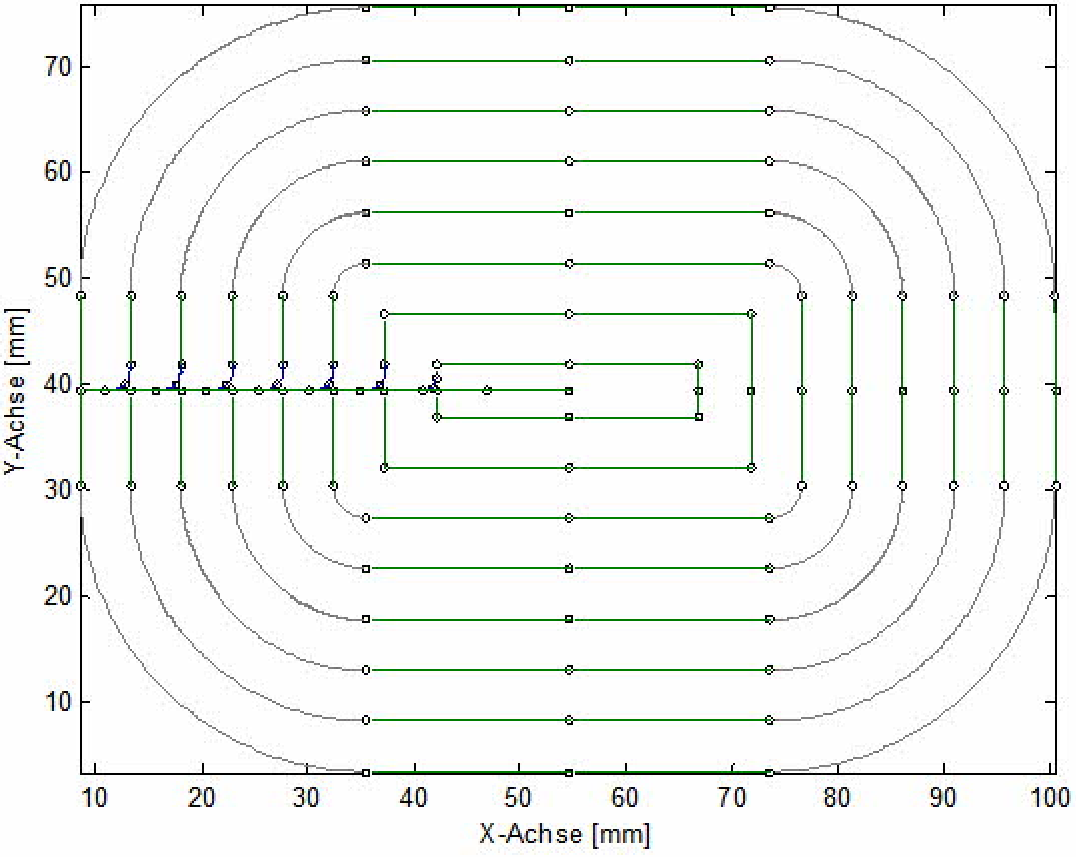


Fig. 45: Example: Pocket milling cycle with polynomial contouring

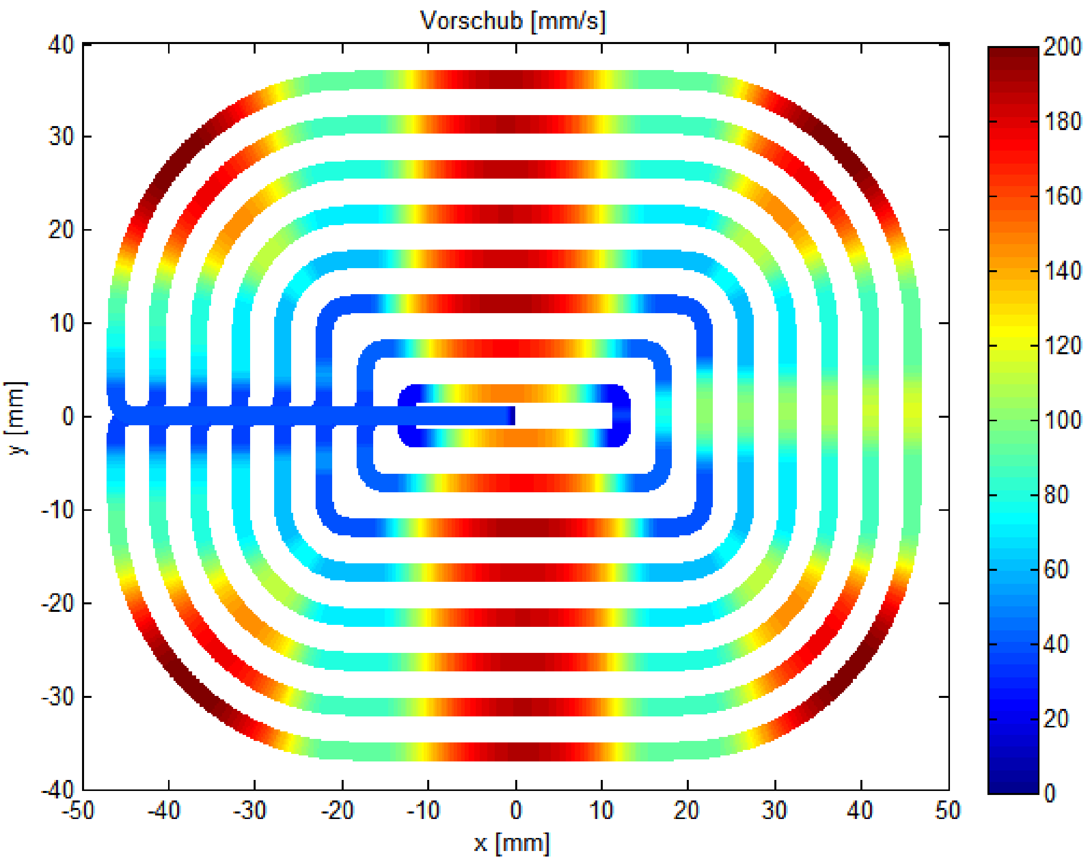


Fig. 46: Feed along the contour with opt_contour_mode=0:

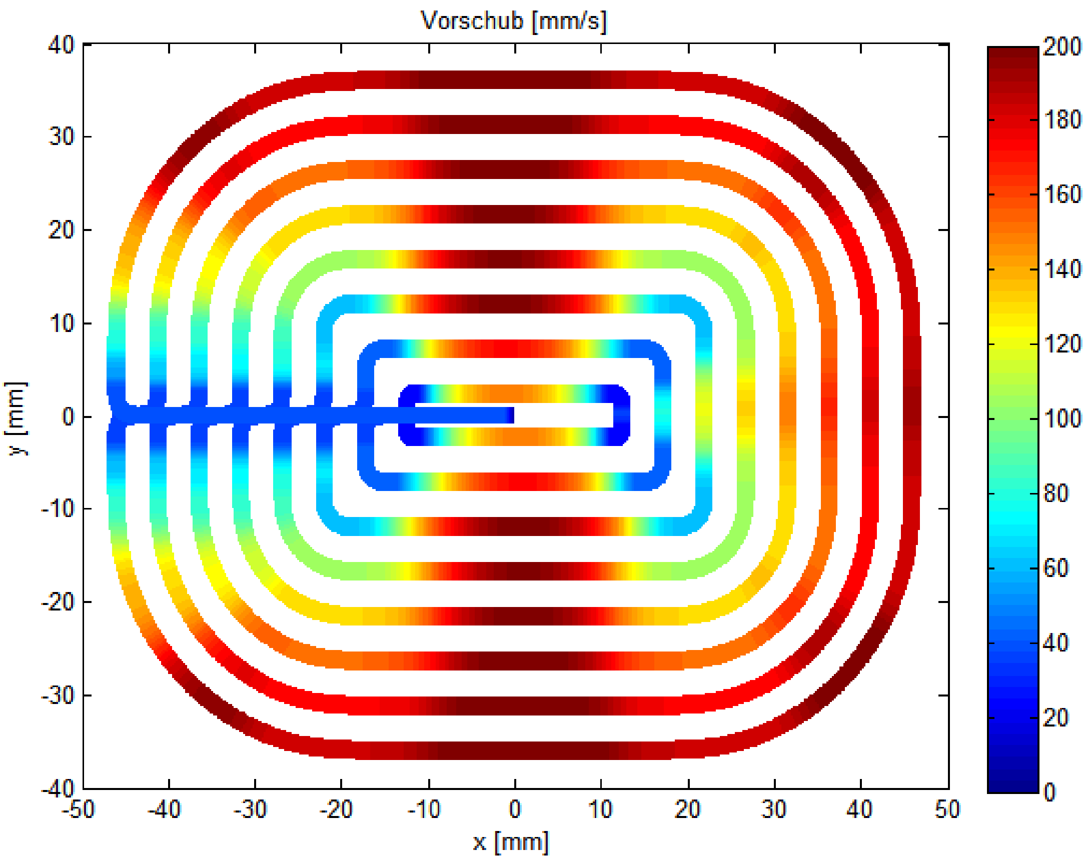


Fig. 47: Feed along the contour with opt_contour_mode=1

→ In this example the machining time is reduced by 18%.

## 1.71 Configuration of interpolator input disable (P-CHAN-00267)

| **P-CHAN-00267** | **Configuring interpolator input disable** |
|---|---|
| Description | Reading further NC blocks can be prevented in the interpolator by using the “Read disable interpolator“ [HLI// Control command of a channel]). By default this control unit acts independently from the currently read block type. This means that when this control unit is activated, no further blocks are read and path motion stops.<br>The 'hli_input_disable_condition' parameter sets a condition for the action of the control unit. For example, it suppresses the reading of the next non-rapid traverse block so that the movement is stopped at the start of the next G01 movement. |
| Parameter | hli_input_disable_condition |
| Data type | STRING |
| Data range | DEFAULT: Reading is stopped immediately when the read disable is set (default).<br>NEXT_NON_RAPID_BLOCK: Reading the first non-rapid traverse block is prevented after the control unit is activated.<br>EXT_RAPID_BLOCK: Reading the first rapid traverse block is prevented after the control unit is inhibited. |
| Dimension | ---- |
| Default value | DEFAULT |

| | |
|---|---|
| Remarks | The process stops immediately before the motion is executed. Technology functions as well as spindle M functions which may be programmed in the same NC row are output to the PLC.<br>If the control unit is deactivated while the interpolator is decelerating because of an active read disable, the motion is continued without stopping.<br>Parameterisation example:<br>Reading the next rapid traverse block is prevented when the read disable is activated. This rapid traverse block is only executed when the read disable is deactivated again.<br>*hli_input_disable_condition NEXT_RAPID_BLOCK* |

## 1.72 Settings for forward/backward movement on the path (forward_backward.*)

With ‘forward/backward movement on the path’ (see [FCT-C7]) an NC program can be moved forwards and backwards. Influences by program decoding are not evaluated again. Only the original motions specified in the NC program and originally decoded are moved forwards and backwards.

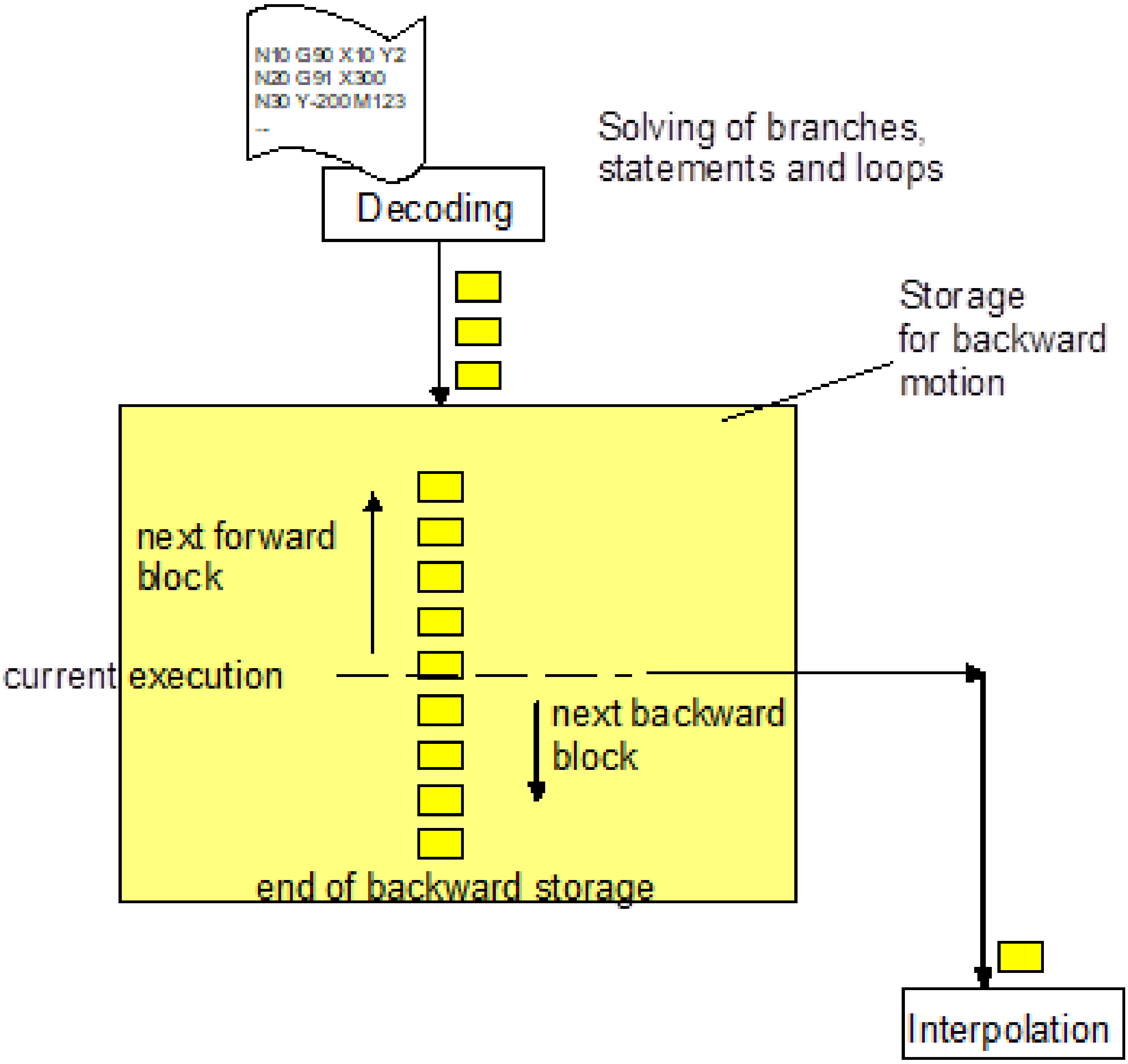
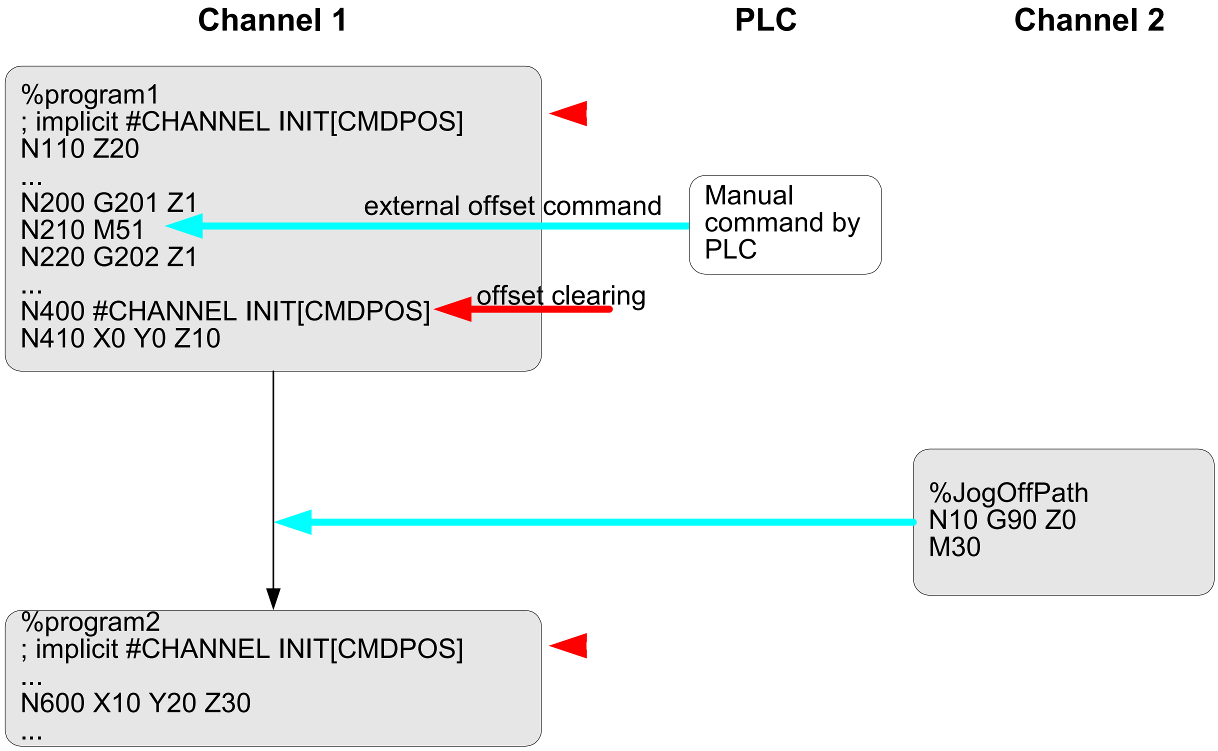


Fig. 48: Scheme of forward/backward control

### 1.72.1 Backward movement with external position offsets (P-CHAN-00275)



```
%program1
; implicit #CHANNEL INIT[CMDPOS]
N110 Z20
...
N200 G201 Z1
N210 M51
N220 G202 Z1
...
N400 #CHANNEL INIT[CMDPOS]
N410 X0 Y0 Z10
```

```
%JogOffPath
N10 G90 Z0
M30
```

```
%program2
; implicit #CHANNEL INIT[CMDPOS]
...
N600 X10 Y20 Z30
...
```

Fig. 49: Program execution with offset overlapping

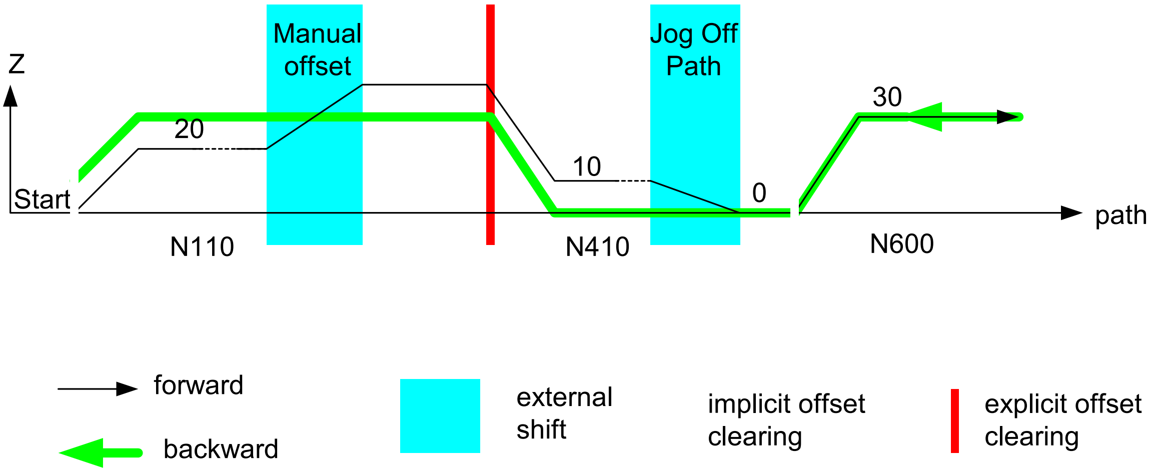


Fig. 50: Movement with forward/backward control with offsets

**NOTICE**

An active kinematic transformation which is especially dynamically different depending on the position, may cause a dynamic overload (or an incomplete loading) of the axes.

Background: If a backward movement has an offset, a deviation is made from the original path. However, dynamic planning was executed with the original path motion in forward direction without offset.

| P-CHAN-00275 | Backward movement with external position offsets |
|---|---|
| Description | The programmed path contour can be shifted by external online influences. Subsequent synchronisation can signal this position offset to the entire NC channel (see #CHANNEL INIT[CMDPOS]). This deletes the offset, i.e. an absolute programmed position then does not include an offset.<br>During forward/backward movement, external offsets are not moved backward like other motions defined in NC program. If there is backward movement on a program position with an external offset, there are two options:<br>1. A further backward movement is not allowed since the positions specified in the NC program must also be reached in the backward direction without offset.<br>2. The offset is retained and backward movement is allowed to continue. The current absolute positions defined in NC program are no longer valid since they are shifted by the currently valid offset.<br>These offsets may be caused by:<br>• Manual mode actions<br>• Shifting the contour with 'Jog of Path' (see [FCT-C15])<br>• Compensations executed online (see [FCT-C20] |
| Parameter | forward_backward.with_offset |
| Data type | BOOLEAN |
| Data range | 0: No backward movement is possible over the program position of an external offset. The backward movement storage is cleared automatically (as for a #BACKWARD STORAGE CLEAR).<br>1: A backward movement is possible beyond the program position of an external offset is possible. The active offset at the time of reversal is retained. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | The backward movement is stopped when the following online influences occur:<br>- Measurement with G100<br>- Homing with G74<br>- Deselection of Online Tool Compensation with #OTC OFF |

## 1.72.2 M00 / M01 synchronisation

During backward and later forward movement, it may be necessary to suppress a stop programmed by M00 or M01. For example, the movement may be stopped only during normal forward movement. The following behaviour of M00 / M01 synchronisation can be parameterised in conjunction with forward/backward movements:

- Suppress stopping during backward movements
- Suppress stopping during next forward movements

For further information see [FCT-C7].

### 1.72.2.1 Programmed M00 stop during backward movements (P-CHAN-00276)

| **P-CHAN-00276** | **Programmed M00 stop during backward movements** |
|---|---|
| Description | Suppress stop during backward movements with M00. |
| Parameter | forward_backward.disable_m00_backward |
| Data type | BOOLEAN |
| Data range | 0: The process also stops in the backward direction with M00.<br>1: Stopping with M00 is omitted in the backward direction. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.72.2.2 Programmed M00 stop during next forward movements (P-CHAN-00277)

| **P-CHAN-00277** | **Programmed M00 stop during next forward movements** |
|---|---|
| Description | Suppress stop in next forward movements with M00. |
| Parameter | forward_backward.disable_m00_2nd_forward |
| Data type | BOOLEAN |
| Data range | 0: Motion is also stopped with M00 in the next forward motion.<br>1: Stopping with M00 is omitted in the next forward direction. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.72.2.3 Programmed M01 stop during backward movement (P-CHAN-00278)

| **P-CHAN-00278** | **Programmed M01 stop during backward movement** |
|---|---|
| Description | Suppress stop during backward movement with M01. |
| Parameter | forward_backward.disable_m01_backward |
| Data type | BOOLEAN |
| Data range | 0: The process also stops in backward direction with M01.<br>1: Stopping with M01 is omitted in the backward direction. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.72.2.4 Programmed M01 stop during next forward movement (P-CHAN-00279)

| **P-CHAN-00279** | **Programmed M01 stop during next forward movement** |
|---|---|

| | |
|---|---|
| Description | Suppression of stop on next forward movement with M01. |
| Parameter | forward_backward.disable_m01_2nd_forward |
| Data type | BOOLEAN |
| Data range | 0: The process is also stopped in next forward direction with M01.<br>1: Stopping with M01 omitted in next forward direction. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.72.3 Automatic reversal after STOP

In the STOP program, the #STOP REVERSIBLE [..] parameter defines the STOP marks at which the machining direction can be reversed.

Similar to M00 / M01, a possible suppression of STOP marks can be configured in the channel parameter list. For example, this suppresses STOP marks in forward or backward movement:

- Suppress stopping during backward movements
- Suppress stopping during single/multiple forward movements

The effectiveness of stops can be pre-assigned globally in the channel parameter list and then individually overwritten in the NC command #STOP REVERSIBLE [..].

For further information see [FCT-C7].

#### 1.72.3.1 Behaviour at STOP marks during backward movement (P-CHAN-00308)

| **P-CHAN-00308** | **Response at STOP marks during backward movement** |
|---|---|
| Description | Suppress stop at STOP during backward movement. |
| Parameter | forward_backward.disable_stop_backward |
| Data type | BOOLEAN |
| Data range | 0: The movement stops at the STOP mark during backward movement.<br>1: The STOP mark is ignored during backward movement |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is available as of CNC Build V3.1.3039.01. |

#### 1.72.3.2 Response at STOP marks during forward movement (P-CHAN-00309)

| **P-CHAN-00309** | **Response at STOP marks during forward movement** |
|---|---|
| Description | Suppress stop at STOP during forward movement. |
| Parameter | forward_backward.disable_stop_1st_forward |
| Data type | BOOLEAN |
| Data range | 0: The motion stops at the STOP mark during forward movement.<br>1: The STOP mark is ignored during forward movement. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is available as of CNC Build V3.1.3039.01. |

#### 1.72.3.3 Response at STOP marks during repeated forward movement (P-CHAN-00310)

| **P-CHAN-00310** | **Response at STOP marks during repeated forward movement** |
|---|---|
| Description | Suppress stop at STOP during forward movement after previous backward movement. |
| Parameter | forward_backward.disable_stop_2nd_forward |
| Data type | BOOLEAN |
| Data range | 0: The motion always stops at the STOP mark during repeated forward movement.<br>1: The STOP mark is ignored during forward movement after previous backward movement. |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | This parameter is available as of CNC Build V3.1.3039.01. |

# 1.73 User-defined data (customer.*)

## 1.73.1 Free values (P-CHAN-00280)

| **P-CHAN-00280** | **Free user-defined values** |
|---|---|
| Description | The user can enter any values in this array. The values are only displayed on the HLI in the element gpCh*[channel_idx]*^.head.**customer_val_r[ ]** (see [HLI:]) and are not used in the controller. This allows the user to transfer configuration data to the PLC or HMI. |
| Parameter | customer.val[i] where i = 0 (application-specific) |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

# 1.74 Encryption of NC programs (P-CHAN-00283)

| **P-CHAN-00283** | **Define file extensions to encrypt NC programs** |
|---|---|
| Description | The NC channel can process encrypted NC programs. Encryption is recognised by the file extension. A maximum of 3 self-defined file extensions are available in the channel parameter 'encryption_extension[...]’ to configure file extensions.<br>A file extension can consist of one to maximum 3 characters. No distinction is made between uppercase and lowercase letters in the file extension. A check is made whether the extension is entered in one of the 3 groups before opening an NC program. If the check is positive, the NC kernel decrypts the NC program with the key belonging to the related group. Both main programs and global subroutines can be encrypted.<br>For more information about encryption see [FCT-C12]. |
| Parameter | encryption_extension[i] where i = 0 2 |
| Data type | STRING |
| Data range | Maximum of 3 characters |
| Dimension | ---- |
| Default value | *encryption_extension[0]* ----<br>*encryption_extension[1]* ----<br>*encryption_extension[2]* ----<br>*encryption_extension[3]* ecy * |
| Remarks | * File extensions can be set for the groups 1 to 3 (Index 0, 1, 2). A further group also exists. This group especially is pre-defined by the controller or machine manufacturer and is used for the encryption of self-created NC programs (e.g. cycles). The extension is 'ecy'. It is recommended not to re-use this extension for new user-defined definitions<br>Parameterisation example:<br>`encryption_extension[0] enc (1st group)`<br>`encryption_extension[1] od (2nd group)`<br>`encryption_extension[2] d (3rd group)` |

# 1.75 Settings for independent axes

## 1.75.1 Position synchronisation of independent axes (P-CHAN-00297)

| P-CHAN-00297 | Position synchronisation of independent axes |
|---|---|
| Description | The synchronisation of independent axis movements with path motions can be controlled by the command WAIT INDP[...]. The synchronisation of independent axes can also be forced by motions within a path compound.<br>The following parameter controls the exact stop characteristic of position synchronisation. |
| Parameter | mode_exact_stop_indp_axis |
| Data type | UNS16 |
| Data range | 0: No implicit exact stop synchronisation with programmed #WAIT INDP[...].<br>1: Exact stop synchronisation with programmed exact stop [G60] on the path.<br>2: Implicit exact stop synchronisation with programmed #WAIT INDP[...]. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

**Examples with effect:**

```
mode_exact_stop_indp_axis 0

%indep1
N10 G00 G90 X0 Y0 Z0
N20 Z[INDP_ASYN POS=500 G0 G90]
N30 #WAIT INDP [Z] ;Wait until command value of Z axis is on target position
N40 X10 Y10
M30


mode_exact_stop_indp_axis 1

%indep2
N10 G00 G90 X0 Y0 Z0
N20 Z[INDP_ASYN POS=500 G0 G90]
N30 G01 G90 X10 Y10 Z500 G60 F3000 ;Wait at motion end until Z
;is in exact stop window (actual value!).
M30



mode_exact_stop_indp_axis 2

%indep3
N10 G00 G90 X0 Y0 Z0
N20 Z[INDP_ASYN POS=500 G0 G90]
N30 #WAIT INDP [Z] ;Wait until Z axis is in exact stop window (actual value !)
N40 X10 Y10
M30
```

## 1.75.2 Implicit synchronisation using independent asynchronous axes (P-CHAN-00451)

| P-CHAN-00451 | Operating the implicit synchronisation using independent asynchronous axes |
|---|---|
| Description | This parameter can be used to influence the operation for the “catching” or the implicit synchronisation of a moving asynchronous axis upon reprogramming. |
| Parameter | mode_implicite_sync_indp_asyn_axis |
| Data type | UNS16 |

| | | |
|---|---|---|
| Data range | 0 , 1<br>The implicit synchronisation will be performed when the affected axis: | |
| | Mode 0 | has a motion path within PCS or when the target points of the motion block with the independent axis movement and the subsequent motion block differ from each other (assuming that the tool offset, the origin offsets and so forth for this axis are 0). |
| | Mode 1 | is programmed within the succeeding motion block |
| Dimension | ---- | |
| Default value | 0 | |
| *Remarks* | The parameter is available as of Build V3.3070.09.<br><br>The default setting is useful if, for instance, the oscillation function is used in connection with NC programs from a post processor which takes along the positions of all axes in the program even though the programmed position does not change. In the default setting, an oscillation which was initiated is not cancelled.<br>Examples<br>N10 G0 X0 Y0 Z0<br>N20 G1 G90 Y10 F5000<br>N30 X[INDP_ASYN G90 **POS50** G01 FEED 3000]<br>N40 Y20<br>N50 X50 (**P-CHAN-00451 = 1**, Synchronisation, waiting for the end of the movement in N30!)<br>N60 Y30<br>M30<br><br>N10 G0 X0 Y0 Z10<br>N20 Z[OSC ON 1ST_POS=-10 **2ND_POS=10** FEED=1000]<br>N30 G1 G90 X100 F1000<br>N40 X100 Y10 Z10<br>(**P-CHAN-00451 = 0**, No synchronisation, oscillation continues!)<br>N50 X100 Y20 Z10<br>N60 X100 Y30 Z10<br>N70 X100 Y40 Z10<br>N80 X100 Y50 Z10<br>M30 | |

## 1.76 Jog of path option after block search (P-CHAN-00305)

| **P-CHAN-00305** | **Jog of path option after block search for coupled axes** |
|---|---|
| Description | This parameter can influences the starting movement of coupled axes (soft gantry) when automatic jog of path is active. If the parameter is 0 (default), master and slave axes are moved decoupled to the forward position.<br>When the parameter is set, the process is reached when soft gantry coupling is active. In this case it is assumed that master and slave axes are already in their correct positions (requested offset between master and slave correctly set or still available. |
| Parameter | block_search_restart_mode |
| Data type | UNS32 |

| | |
|---|---|
| Data range | 0: This setting is useful for machines where the master and slave axes move different workpieces or where a master-slave offset change is required in the jog of path motion since master and slave may be in any position at the time of start.<br>1: This setting is useful for machines, where the master and slave axes move a common work piece (mechanic link) and hence the required distance due to the workpiece fixing must be constant. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.77 Checking the license in the clone channel (P-CHAN-00306)

| **P-CHAN-00306** | **Checking the license in the clone channel** |
|---|---|
| Description | This parameter checks the license in the clone channel. When this parameter is set, a check is made during start-up whether all the axes configured in the channel are clone axes. Only then is this channel license-free. The check is also made at RESET and axis exchange. If an axis is not a clone axis, an error message is output. |
| Parameter | jog_of_path_only |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter allows the 2nd channel to be used as a Jog-Of-Path channel even with a single-channel license.<br>The parameter is available as of Build V3.1.3037.17. |

## 1.78 Positioning of modulo axes on shortest path (P-CHAN-00346)

| **P-CHAN-00346** | **Positioning of modulo axes on shortest path** |
|---|---|
| Description | This parameter always permits the positioning of modulo rotary axes on the shortest path. It is not necessary to program the sign when defining the direction of rotation. |
| Parameter | enable_mod_axis_always_shortest_way |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.79 Optimised insertion of #FLUSH CONTINUE (P-CHAN-00341)

| **P-CHAN-00341** | **Optimised insertion of #FLUSH CONTINUE** |
|---|---|
| Description | If #FLUSH CONTINUE is programmed between two motion blocks and when this parameter is set to 1, the command #FLUSH CONTINUE is automatically offset to the middle of the first path motion. The transition of the two motion blocks can then be changed, e.g. by polynomial smoothing. |
| Parameter | opt_insert_flush_continue |
| Data type | UNS32 |

| | |
|---|---|
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.80 Enabling 2-path programming (P-CHAN-00261)

| **P-CHAN-00261** | **Enable 2-path programming** |
|---|---|
| Description | This parameter enables the NC syntax for 2-path programming. It permits the programming of two synchronous motions (paths) for 2 axis groups in the same NC block. The motions are separated in the NC block by a colon '**:**'.<br>**`<global>   <separator>   <path1>   <separator>   <path2>`**<br>`Nxx G01 G90 F100    :   X100 Y100 Z0      :      U100 V100 W0` |
| Parameter | multi_path_configuration |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | When 2-path programming is active, the colon ':' signifies a separator in the syntax. Jump marks to block numbers Nxx: (also called expression labels) is not possible, but only jumps on string labels are allowed. |

# 1.81 Configuration of general display data

## 1.81.1 Definition of channel display information

### 1.81.1.1 Channel name (P-CHAN-00174)

| P-CHAN-00174 | Define a channel name for display on the HLI |
|---|---|
| Description | This parameter permits the user-specific definition of a channel name which is provided for display in the PLC interface (HLI). |
| Parameter | channel_name |
| Data type | STRING |
| Data range | <empty_string>: No display of a channel name (default).<br><Channel_name>: Name of the NC channel (string 16 characters long, application-specific). |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameterisation example:<br>*channel_name CHANNEL-1 (name of NC channel for display)*<br>* Note: The default value of variables is a blank string. |

### 1.81.1.2 Channel type (P-CHAN-00175)

| P-CHAN-00175 | Definition of a channel type for display on the HLI |
|---|---|
| Description | This parameter permits the user-specific definition of a channel identifier which is provided for display in the PLC interface (HLI). |
| Parameter | channel_type |
| Data type | UNS32 |
| Data range | 0 ≤ channel_type ≤ MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*channel_type 123 (NC channel identifier for display)* |

## 1.81.2 Suppressing display of file name/file offset (P-CHAN-00180)

| P-CHAN-00180 | Suppress display of file name/file offset |
|---|---|
| Description | This parameter defines the program level up to which the active file name and file offset are updated for display. The display of file name and file offset is then frozen at all program levels below this limit. If the parameter is not defined or is set to 0, the file name and the file offset are displayed for all permissible program levels. |
| Parameter | suppress_prg_display_level |
| Data type | UNS16 |
| Data range | 0 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 50 * |
| Remarks | If the parameter is assigned the value 1, the data is only updated for the main program. Each time the parameter is increased, the data is updated for a further subroutine level.<br>* Maximum permissible nesting depth of NC programs |

### 1.81.3 Diameter display (P-CHAN-00256)

| P-CHAN-00256 | Diameter display during turning |
|---|---|
| Description | This parameter switches over the display of the (PCS) position values provided by the PLC interface (HLI) for turn machining. When the parameter is set to 1, the display shows the average value when diameter programming (G51) is active. |
| Parameter | display_diameter_pos |
| Data type | BOOLEAN |
| Data range | 0: Radius display with active G51 (default).<br>1: Diameter display with active G51. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.81.4 Display format during machining simulation (P-CHAN-00121)

| P-CHAN-00121 | Display format during machining simulation |
|---|---|
| Description | This parameter switches over the format of display data for the coordinate system at the interface of the machining simulation. |
| Parameter | simu_output_wcs |
| Data type | BOOLEAN |
| Data range | 0: Display of axis coordinates including offsets (machine coordinates)<br>1: Display of absolute coordinates without offsets (programmed coordinates) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

### 1.81.5 Display of active path feed (P-CHAN-00328)

| P-CHAN-00328 | Display of active path feed |
|---|---|
| Description | This parameter selects the displayed active path feed. |
| Parameter | mode_feed_display |
| Data type | UNS32 |
| Data range | 0: Default feed of the feed axes in the channel. The displayed path feed results from the motion of these PCS axes when only main or feed axes are moved. If none of the listed feed axes are moved, the feed of the leading tracked or orientation axis is displayed. This means that the active feed value is always != 0 when axes are moved.<br>1: Real (all feed axes in the channel). The displayed feed is calculated from the PCS velocities of the feed axes.<br>2: TCP (main axes are feed axes)<br>- when kinematics are active. If only rotary orientation axes are moved, the TCP feed is 0. Otherwise, the feed results from the motion of the PCS feed axes.<br>- when kinematics are inactive. Display of default path feed<br>3: Real (as for 1.); in addition to real feed display, the service distance is also displayed or provided here based on the real PCS aggregate contour path. (Tracked axis motions are therefore not included in the aggregate contour path.) |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | |

## 1.81.6 Selecting the coordinate system to display axis positions (P-CHAN-00330)

| **P-CHAN-00330** | **Select the coordinate system for display of axis positions** |
|---|---|
| Description | This parameter selects the coordinate system to display axis positions. |
| Parameter | display_top_coord_sys |
| Data type | STRING |
| Data range | IPO: Axis positions (target position, interpolated command position, actual position and distance to target position) are output in the currently selected coordinate system (default).<br>PCS: Output is in the top-level coordinate system. |
| Dimension | ---- |
| Default value | IPO |
| Remarks | |

## 1.81.7 Display of axis command values and target points in Cartesian coordinate systems (P-CHAN-00331)

| **P-CHAN-00331** | **Display of axis command values and target points in Cartesian coordinate systems** |
|---|---|
| Description | This parameter activates the display of axis command values and target points in Cartesian coordinate systems:<br>• If only kinematic transformation (step 0 and/or step 1) is selected, axis command values and target points are displayed in the Cartesian machine coordinate system.<br>• If Cartesian coordinate systems are (additionally) selected via #CS ON, axis command values and target points are output in the programming machine coordinate system.<br>• If Cartesian coordinate systems are (additionally) defined via #CS ADD, axis command values are output in all defined coordinate systems while the target points are only displayed in the programming coordinate system selected via #CS SELECT. |
| Parameter | kin_trafo_enable_cs_coord_display |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.81.8 Display of transformed actual positions (P-CHAN-00344)

| **P-CHAN-00344** | **Suppressed output of actual positions transformed kinematically and/or in the Cartesian system.** |
|---|---|
| Description | This parameter suppresses the output of actual positions which are executed by kinematic and/or Cartesian transformation. This reduces the computing time in the cyclic task. |
| Parameter | suppress_trafo_curr_pos |
| Data type | BOOLEAN |

| | |
|---|---|
| Data range | 0: Actual positions of the physical axes are cyclically transformed (forward) and made available (coordinate system selected by the P-CHAN-00331 parameter) (default).<br>1: No transformation of actual positions. The command position values are displayed instead. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.81.9 Consider mirroring in display and manual mode (P-CHAN-00434)

| P-CHAN-00434 | Consider mirroring in display and manual mode |
|---|---|
| Description | The G functions G21-G23 and G351 can activated the mirroring of coordinates on axes. Normally, this has no influence on the display in the interpolator or on movement direction in manual mode. When this parameter is activated, the display data is mirrored on an axis if mirroring is selected. In addition, the movement direction is inverted in manual mode. |
| Parameter | mirror_display_positions |
| Data type | BOOLEAN |
| Data range | 0: Active mirroring has no influence on the display in the interpolator and manual mode.<br>1: Active mirroring is considered in the interpolator display data and in manual mode. |
| Dimension | --- |
| Default value | 0 |
| Remarks | **CAUTION:** When this parameter is activated, the motion direction of this axis is inverted in manual mode if the mirroring function is active**.** |

## 1.81.10 Display motion range limits in PCS coordinate system (P-CHAN-00489)

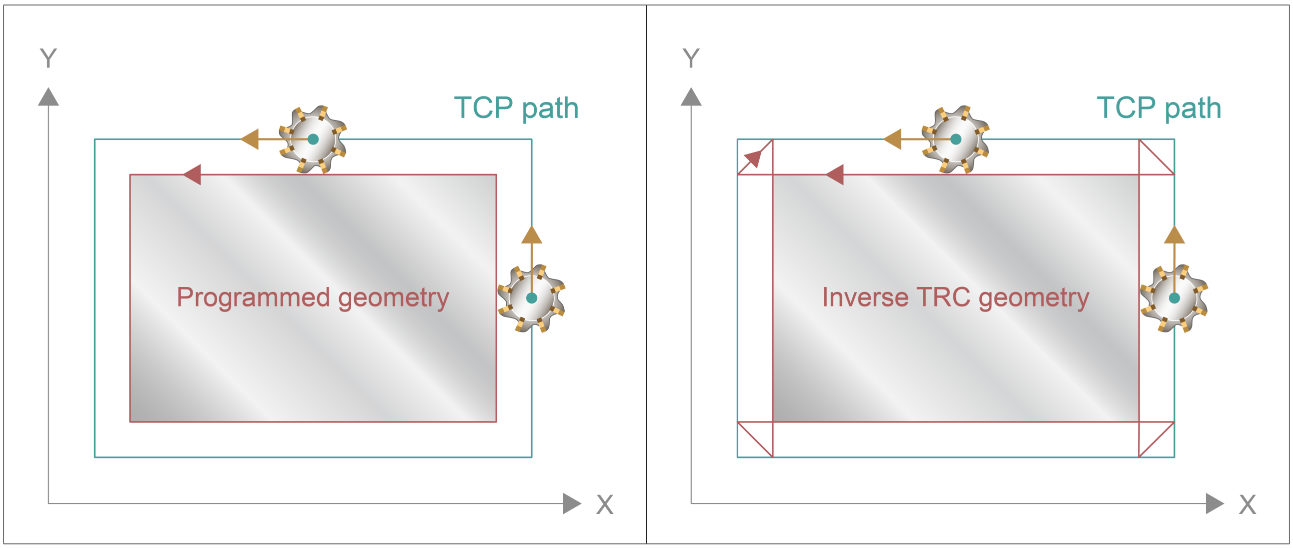
| P-CHAN-00489 | Display motion range limits in PCS coordinate system |
|---|---|
| Description | This parameter enables a display of the software limits in the PCS coordinate system. All offsets are considered (e.g. zero point or tool offsets) and Cartesian transformations (#CS). In order to consider active mirroring functions (G21-G23, G351) the channel parameter P-CHAN-00434 [▶ 285] must also be activated..<br>A display of PCS limits is not possible for axes that are a component of an active kinematic transformation. In this case, the limits are set to +/-1E200.<br>Offset limits for manual mode are not considered in this review.<br>If the display of PCS motion range limits is activated, the limits can be requested by CNC objects in the data of the interpolator axes of the GEO task,<br>• Positive PCS limit<br>• Negative PCS limit<br>• Distance to positive PCS limit<br>• Distance to negative PCS limit |
| Parameter | display_pcs_limits |
| Data type | BOOLEAN |
| Data range | 0: The PCS display of motion range limits is disabled.<br>1: PCS motion range limits are calculated and supplied. |
| Dimension | --- |

| | |
|---|---|
| Default value | 0 |
| Remarks | **NOTE:** A display of PCS limits is not possible for axes that are a component of an active kinematic transformation (#TRAFO) is not possible.<br>This function is available as of CNC Build V3.1.3079.32. |

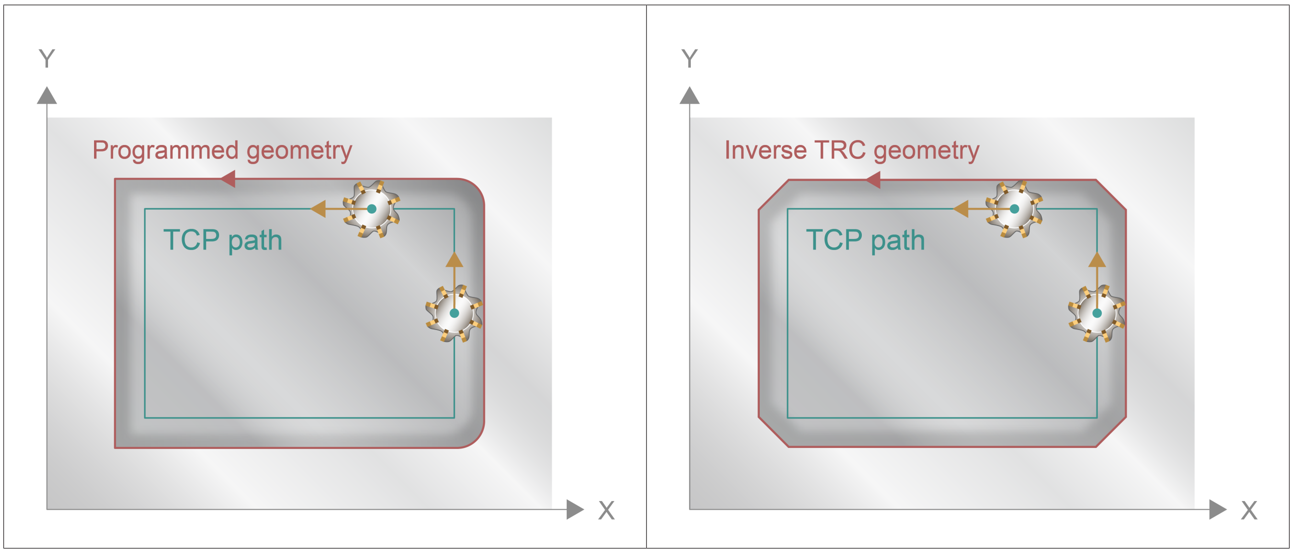
## 1.81.11 Display PCS positions inverse TRC (P-CHAN-00487)

| **P-CHAN-00487** | **TRC inverse display PCS positions** |
|---|---|
| Description | If tool radius compensation (TRC) is active, the three Cartesian PCS positions of the display refer to the TCP, i.e. the tool centre point.<br>If you also require the CNC to display the corresponding PCS coordinates of the inverse TRC for this data in the display, set this parameter to 1.<br>The positions of the two main axes of the place are available using the following CNC objects:<br>• TC: inverse TRC 1ST (Task GEO - index group 0x12130<$C_{ID}$> Index offset 0x13A)<br>• TC: inverse TRC 2ND (Task GEO - index group 0x12130<$C_{ID}$> Index offset 0x13B)<br>See Contour examples [▶ 287] |
| Parameter | trc_inverse_display_pcs_pos |
| Data type | BOOLEAN |
| Data range | 0 , 1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | These PCS positions generally do not correspond to PCS positions that are programmed interpolated.<br>Parameter available as of CNC Build V3.1.3079.33 and higher |

**External contour element with tool radius compensation**



**Internal contour element with tool radius compensation**



# 1.82 Logging of manual block commands

## 1.82.1 Name of manual block log file (P-CHAN-00338)

| **P-CHAN-00338** | **Name of the manual block log file** |
|---|---|
| Description | If the name is specified, each manual block command of the NC channel is logged to this file. In addition to later diagnostics capability, this file is also used for error display. This means that, if a CNC error occurs within the manual block, the commanded manual block is immediately displayed in the error message. |
| Parameter | mdi_log_file |
| Data type | STRING |
| Data range | Maximum 256 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string. |

## 1.82.2 Maximum size of the manual block log file (P-CHAN-00339)

| **P-CHAN-00339** | **Maximum size of the manual block log file** |
|---|---|
| Description | Since the log file grows with every new manual block, this parameter limits the file size. If the size of the log file is exceeded, it is first cleared automatically before the current manual block is logged. |
| Parameter | mdi_log_file_max_size |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | ---- |
| Default value | 0 * |
| Remarks | * no size limitation |

# 1.83 Filter parameters for error handling in the NC channel (error_filter[i].*)

Users/machine manufacturers parameterise the required actions or filtering operations for error messages for each platform/channel/axis. For more information see FCT-M7

| Structure name | Index |
|---|---|
| error_filter[i] | 0 ≤ i ≤ 3 (maximum number of error filters: 4) |

## 1.83.1 Error cause (P-CHAN-00378)

| P-CHAN-00378 | Error cause (filtering error messages in the channel) |
|---|---|
| Description | The individual error codes can be listed as numbers or texts, whereby the entire row must comply with the following syntax:<br>( number \| text ) {, ( number \| text ) }<br><br>where:<br>number:= CNC error number<br>text:=" error-specific text "<br><br>Example:<br>error_filter[0].reason "D012:", 123000, 123001<br><br>If an error is logged, the program looks in the defined platform/channel/axis filters whether a user-specific filter rule is defined for it. |
| Parameter | error_filter[i].reason where i = 0 ... 3 (maximum number of filters, application-specific) |
| Data type | STRING |
| Data range | Maximum of 96 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string. |

## 1.83.2 Error action (P-CHAN-00379)

<table>
<tr><th>P-CHAN-00379</th><th colspan="2">Error action (filtering error messages in the channel)</th></tr>
<tr><td>Description</td><td colspan="2">Action that is to be performed if an error occurs.</td></tr>
<tr><td>Parameter</td><td colspan="2">error_filter[i].action where i = 0 ... 3 (maximum number of filters, application-specific)</td></tr>
<tr><td>Data type</td><td colspan="2">STRING</td></tr>
<tr><td rowspan="6">Data range</td><td colspan="2">ACTION = NONE | DRIVE_STATE_REQ | PRE_RUN_STATE_REQ | RUN_STATE_REQ</td></tr>
<tr><td><b>Keyword</b></td><td><b>Meaning</b></td></tr>
<tr><td>NONE</td><td>No action</td></tr>
<tr><td>DRIVE_STATE_REQ</td><td>Reading out the drive status</td></tr>
<tr><td>PRE_RUN_STATE_REQ</td><td>Error at start-up of the controller bus in PRE-run state</td></tr>
<tr><td>RUN_STATE_REQ</td><td>Error at start-up of the controller bus in Run state</td></tr>
<tr><td>Dimension</td><td colspan="2">----</td></tr>
<tr><td>Default value</td><td colspan="2">*</td></tr>
<tr><td>Remarks</td><td colspan="2">For SERCOS drive profiles:<br>• DRIVE_STATE_REQ S-0-0095 diagnostic</td></tr>
</table>

| | |
|---|---|
| | • PRE_RUN_STATE_REQ S-0-0021: list of unknown operation data in CP2 -> CP3, command 127<br>• RUN_STATE_REQ S-0-0022: list of unknown operation data in CP3 -> CP4, command 128<br>For ProfiDrive profiles:<br>• <all actions> Parameter 945<br>For CANopen profiles<br>• <all actions> Parameter ID603F<br>* Note: The default value of variables is a blank string. |

## 1.83.3 Conditional activation (P-CHAN-00380)

| **P-CHAN-00380** | **Conditional activation (filtering error messages in the channel)** |
|---|---|
| Description | This filter rule is activated when the corresponding bit is set via the user interface or the PLC (HLI::Control Unit- Activating error filter rules - Channel). |
| Parameter | error_filter[i].conditional_activation where i = 0 ... 3 (maximum Number of filters, application-specific) |
| Data type | UNS32 |
| Data range | 32-bit |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameterisation example:<br>*error_filter[0].conditional_activation 0x2*<br>An activation bit = 0 means that the action is always executed. |

## 1.83.4 Conditional action (P-CHAN-00381)

| **P-CHAN-00381** | **Conditional action (filtering error messages in the channel)** |
|---|---|
| Description | Action that is to be executed if an error occurs and if the condition applies. |
| Parameter | error_filter[i].conditional_action where i = 0 ... 3 (maximum number of filters, application-specific) |
| Data type | STRING |
| Data range | ACTION = NONE \| ( [ HIDE ] [ FORCE ] )<br>FORCE = F_WARNING \| F_SYNTAX \| F_ERROR \| F_SEVERE \| F_FATAL<br>HIDE = [ HIDE ] [ HIDE_LOG ] [ HIDE_PRINT ] [ HIDE_REPORT ]<br>NONE: no action<br>HIDE: Suppress every error output<br>HIDE_LOG: Error output to error log file is suppressed<br>HIDE_DISPLAY: Error output is suppressed<br>HIDE_REPORT: Application-specific error output is suppressed<br>F_WARNING: Error is output as a WARNING (remedy class = 0)<br>F_SYNTAX: Error is output as a syntax error (remedy class = 2)<br>F_ERROR: Error due to NC program or other operator action (error remedy class = 5) |

| | |
|---|---|
| | F_SEVERE: Severe error, requires a warm start (remedy class = 6)<br>F_FATAL: Severe error, requires a complete cold start (remedy class = 7) |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string. |

## 1.83.5 Conditional filter activation (P-CHAN-00382)

| **P-CHAN-00382** | **Conditional filter activation (filtering error messages in the channel))** |
|---|---|
| Description | The individual error codes can be listed as numbers or texts, whereby the entire row must comply with the following syntax:<br>( number \| text ) {, ( number \| text ) }<br><br>where:<br>number:= CNC error number<br>text := **"** error-specific text **"** |
| Parameter | error_filter[i].conditional_param where i = 0 ... 3 (maximum number of filters, application-specific) |
| Data type | STRING |
| Data range | Maximum of 96 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | Parameterisation example:<br>*error_filter[0].conditional_param* "D012:", 123, 1001<br><br>Individual error texts are currently only checked when the SERCOS drive error S95 is read out.<br>Error numbers are only checked in case of SERCOS drive errors (S21 and S22) and in case of ProfiDrive drive errors (parameter 945).<br>* Note: The default value of variables is a blank string. |

## 1.83.6 Output of additional error information (P-CHAN-00383)

| **P-CHAN-00383** | **Output of additional error information (filtering error messages in the channel))** |
|---|---|
| Description | This text is forwarded transparently via the CNC_ERROR_INFO data structure if the filter condition applies. This means that the user has the option to output an additional error text conditionally. |
| Parameter | error_filter[i].conditional_output where i = 0 ... 3 (maximum number of filters, application-specific) |
| Data type | STRING |
| Data range | Maximum of 32 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string. |

# 1.84 Settings for programming coordinate systems (coordinate_system.*)

### Definition of the rotation mode (P-CHAN-00393)

| **P-CHAN-00393** | **Definition of rotation mode for a coordinate system** |
|---|---|
| Description | The rotation mode defines the axes about which the coordinate system is to rotate in order to reach the new position. A distinction here is made between intrinsic and extrinsic rotation. |
| Parameter | coordinate_system.rotation_mode_fixed |
| Data type | BOOLEAN |
| Data range | 0: Each rotation about the new axis of the currently rotated coordinate system (intrinsic, default).<br>1: Each rotation about the “fixed) axis of the coordinate system at the start of rotation (extrinsic). |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is available as of CNC Build V3.1.3039.06 and higher. The pre-assignment can be changed in the NC program by #CS MODE ON [ROTATION_MODE_FIXED]. |

## 1.84.1 Coordinate system settings (CS, ACS, BCS)

Settings for machining coordinate systems (#CS), coordinate systems for fixture adaptation compensation (#ACS) and basic coordinate systems (#BCS).

Coordinate systems can be predefined using the parameters below. They can be activated and changed in the NC program.

| **Structure name** | **Index** |
|---|---|
| def[i] | i = 0,…, 9 (Maximum number of coordinate system definitions) |
| path[j] | j = 0.1 (path, For multipath programming 0: First/main path, 1: Second path) |

### 1.84.1.1 Identifier CS/ACS/BCS (P-CHAN-00490)

| **P-CHAN-00490** | **Identifier CS/ACS/BCS** |
|---|---|
| Description | Identifier for use of the offset with the commands #CS, #ACS or #BCS in the NC program |
| Parameter | coordinate_system.def[i].id |
| Data type | STRING |
| Data range | 7 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string.<br>As of CNC Build V3.01.3079.36 |

### 1.84.1.2 Translation of first axis (P-CHAN-00491)

| **P-CHAN-00491** | Translation of first axis |
|---|---|
| Description | First component of the translatory offset vector in [mm]. (These refer to the main axes in the sequence contained in G17) |

| | |
|---|---|
| Parameter | coordinate_system.def[i].path[j].translation.t1 |
| Data type | REAL64 |
| Data range | |
| Dimension | [mm] |
| Default value | 0.0 |
| Remarks | As of CNC Build V3.01.3079.36 |

### 1.84.1.3 Translation of second axis (P-CHAN-00492)

| **P-CHAN-00492** | **Translation of second axis** |
|---|---|
| Description | Second component of the translatory offset vector in [mm]. (These refer to the main axes in the sequence contained in G17). |
| Parameter | coordinate_system.def[i].path[j].translation.t2 |
| Data type | REAL64 |
| Data range | |
| Dimension | [mm] |
| Default value | 0.0 |
| Remarks | As of CNC Build V3.01.3079.36 |

### 1.84.1.4 Translation of third axis (P-CHAN-00493)

| **P-CHAN-00493** | **Translation of third axis** |
|---|---|
| Description | Third component of the translatory offset vector in [mm]. (These refer to the main axes in the sequence contained in G17). |
| Parameter | coordinate_system.def[i].path[j].translation.t3 |
| Data type | REAL64 |
| Data range | |
| Dimension | [mm] |
| Default value | 0.0 |
| Remarks | As of CNC Build V3.01.3079.36 |

### 1.84.1.5 Rotation angle of first rotation (P-CHAN-00494)

| **P-CHAN-00494** | **Rotation angle of first rotation** |
|---|---|
| Description | Rotation angle in [deg] |
| Parameter | coordinate_system.def[i].path[j].rotation.a1 |
| Data type | REAL64 |
| Data range | |
| Dimension | [deg] |
| Default value | 0.0 |
| Remarks | As of CNC Build V3.01.3079.36 |

### 1.84.1.6 Rotation angle of second rotation (P-CHAN-00495)

| **P-CHAN-00495** | **Rotation angle of second rotation** |
|---|---|
| Description | Rotation angle in [deg] |
| Parameter | coordinate_system.def[i].path[j].rotation.a2 |
| Data type | REAL64 |
| Data range | |
| Dimension | [deg] |

| | |
|---|---|
| Default value | 0.0 |
| Remarks | As of CNC Build V3.01.3079.36 |

### 1.84.1.7 Rotation angle of third rotation (P-CHAN-00496)

| **P-CHAN-00496** | **Rotation angle of third rotation** |
|---|---|
| Description | Rotation angle in [deg] |
| Parameter | coordinate_system.def[i].path[j].rotation.a3 |
| Data type | REAL64 |
| Data range | |
| Dimension | [deg] |
| Default value | 0.0 |
| Remarks | As of CNC Build V3.01.3079.36 |

### 1.84.1.8 Parameterisation example

Parameterisation in the channel parameter list

```
coordinate_system.def[0].id                    test1
coordinate_system.def[0].path[0].translation.t1  10
coordinate_system.def[0].path[0].translation.t2  20
coordinate_system.def[0].path[0].translation.t3  30
coordinate_system.def[0].path[0].rotation.a1      0
coordinate_system.def[0].path[0].rotation.a2      0
coordinate_system.def[0].path[0].rotation.a3     90
```

Use in the NC program

```
#CS ON [test1]
#ACS ON [test1]
#BCS ON [test1]
```

Before first use, an IS can be overwritten in the NC program.

```
#CS ON [test1] [0,0,0]
#ACS ON [test1]
#BCS ON [test1]
```

The parameterised values are again valid then the NC program is rebooted.

## 1.84.2 Transformation stack parameters

| **Structure name** | **Index** |
|---|---|
| **trafo_stack[i]** | i = 0,..,4 (maximum number of definable transformation stacks) |

### 1.84.2.1 Name of the transformation stack (P-CHAN-00752)

| **P-CHAN-00752** | **Name of transformation stack** |
|---|---|
| Description | Name of transformation stack |
| Parameter | trafo_stack[i].name |
| Data type | STRING |
| Data range | 7 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string.<br>As of CNC Build V3.01.3079.36 |

### 1.84.2.2 Name of the kinematic (P-CHAN-00753)

| P-CHAN-00753 | Name of the kinematic |
|---|---|
| Description | Name of the kinematic or kinematics contained in the transformation stack. |
| Parameter | trafo_stack[i].kin[j] |
| Data type | STRING |
| Data range | 7 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string.<br>j = 0 corresponds to the kinematic name of the first kinematic step (kin_step= 1)<br>j = 1 corresponds to the kinematic name of the second kinematic step (kin_step=2)<br>As of CNC Build V3.01.3079.34 |

### 1.84.2.3 ID of CS offset (P-CHAN-00754)

| P-CHAN-00754 | ID of CS offset |
|---|---|
| Description | Using the defined identifier of the stored machining coordinate system (#CS) with P-CHAN-00490 [▸ 292]. |
| Parameter | trafo_stack[i].cs.id[k] |
| Data type | STRING |
| Data range | 7 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | *<br>k = 0,..,4. Coordinate system with Index 0 is the lowest CS<br>As of CNC Build V3.01.3079.36 |

### 1.84.2.4 ID of ACS offset (P-CHAN-00755)

| P-CHAN-00755 | ID of ACS offset |
|---|---|
| Description | Using the defined identifier of the stored coordinate system for fixture adaptation compensation (#ACS) with P-CHAN-00490 [▸ 292]. |
| Parameter | trafo_stack[i].acs.id[k] |
| Data type | STRING |
| Data range | 7 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string.<br>k = 0, … ,4. ACS with Index 0 is the lowest ACS<br>As of CNC Build V3.01.3079.36 |

### 1.84.2.5 ID of BCS offset (P-CHAN-00756)

| P-CHAN-00756 | ID of BCS offset |
|---|---|
| Description | Using the defined identifier of the stored basic coordinate system (#BCS) with P-CHAN-00490 [▸ 292]. |
| Parameter | trafo_stack[i].bcs.id[k] |
| Data type | STRING |

| | |
|---|---|
| Data range | 7 characters |
| Dimension | ---- |
| Default value | - |
| Remarks | k = 0, …,4. Basic coordinate system with Index 0 is the lowest BCS<br>As of CNC Build V3.01.3079.36 |

### 1.84.2.6 Name of activated transformation stack at program start (P-CHAN-00757)

| P-CHAN-00757 | Name of activated transformation stack at program start |
|---|---|
| Description | Specify the defined name P-CHAN-00752 [▶ 294]. This stack is enabled at program start.<br>This parameter specifies the name of the transformation stack to be enabled at program start. The name must be specified in P-CHAN-00752 [▶ 294]. |
| Parameter | trafo_stack_name_active_prog_start |
| Data type | STRING |
| Data range | 7 characters |
| Dimension | ---- |
| Default value | * |
| Remarks | * Note: The default value of variables is a blank string.<br>As of CNC Build V3.01.3079.36 |

## 1.84.3 Definition of rotation sequence (P-CHAN-00394)

| P-CHAN-00394 | Definition of rotation sequence for a coordinate system |
|---|---|
| Description | This parameter defines the rotation sequence for coordinate systems. There are 12 different combinations. The default convention is called ZYX or YAW - PITCH - ROLL. |
| Parameter | coordinate_system.rotation_sequence |
| Data type | STRING |
| Data range | XYZ XYX XZY XZX YXZ YXY YZX YZY ZXY ZXZ ZYX ZYZ |
| Dimension | ---- |
| Default value | ZYX |
| Remarks | This parameter is available as of CNC Build V3.1.3039.06 and higher. Pre-assignment can be changed in the program by #CS MODE ON [ROTATION_SEQUENCE..]. |

## 1.84.4 2-path programming: Selection of the display coordinate system (P-CHAN-00395)

| P-CHAN-00395 | Select the display coordinate system of the second path with 2-path programming |
|---|---|
| Description | When this parameter is initialised with the value 0, each path is displayed in its own Cartesian coordinate system in 2-path programming. When it is initialised with the value 1, the second path is displayed in the coordinate system of the first path. |
| Parameter | coordinate_system.display_global |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

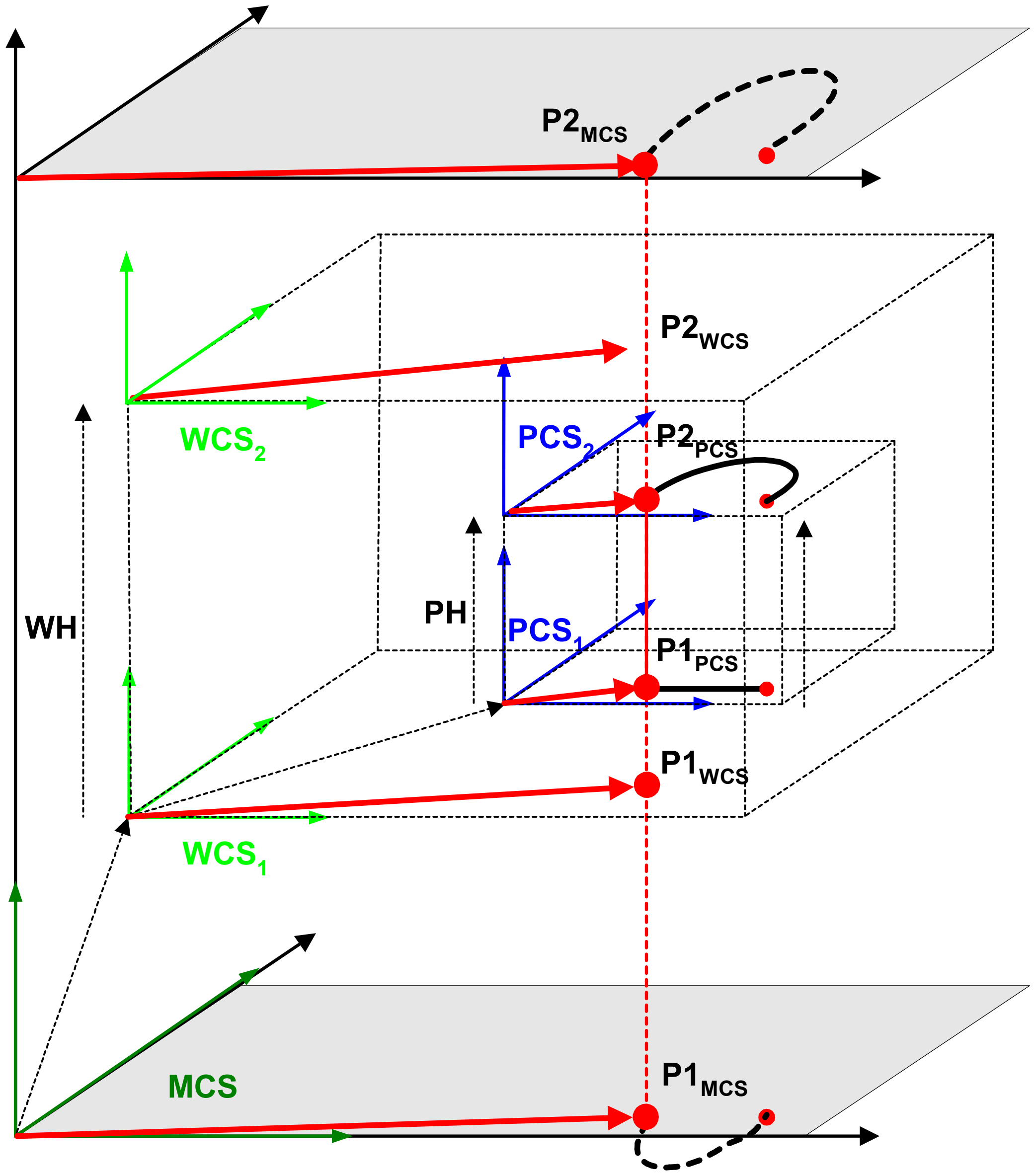


Fig. 51: Display CS for each single path (display_global = 0)

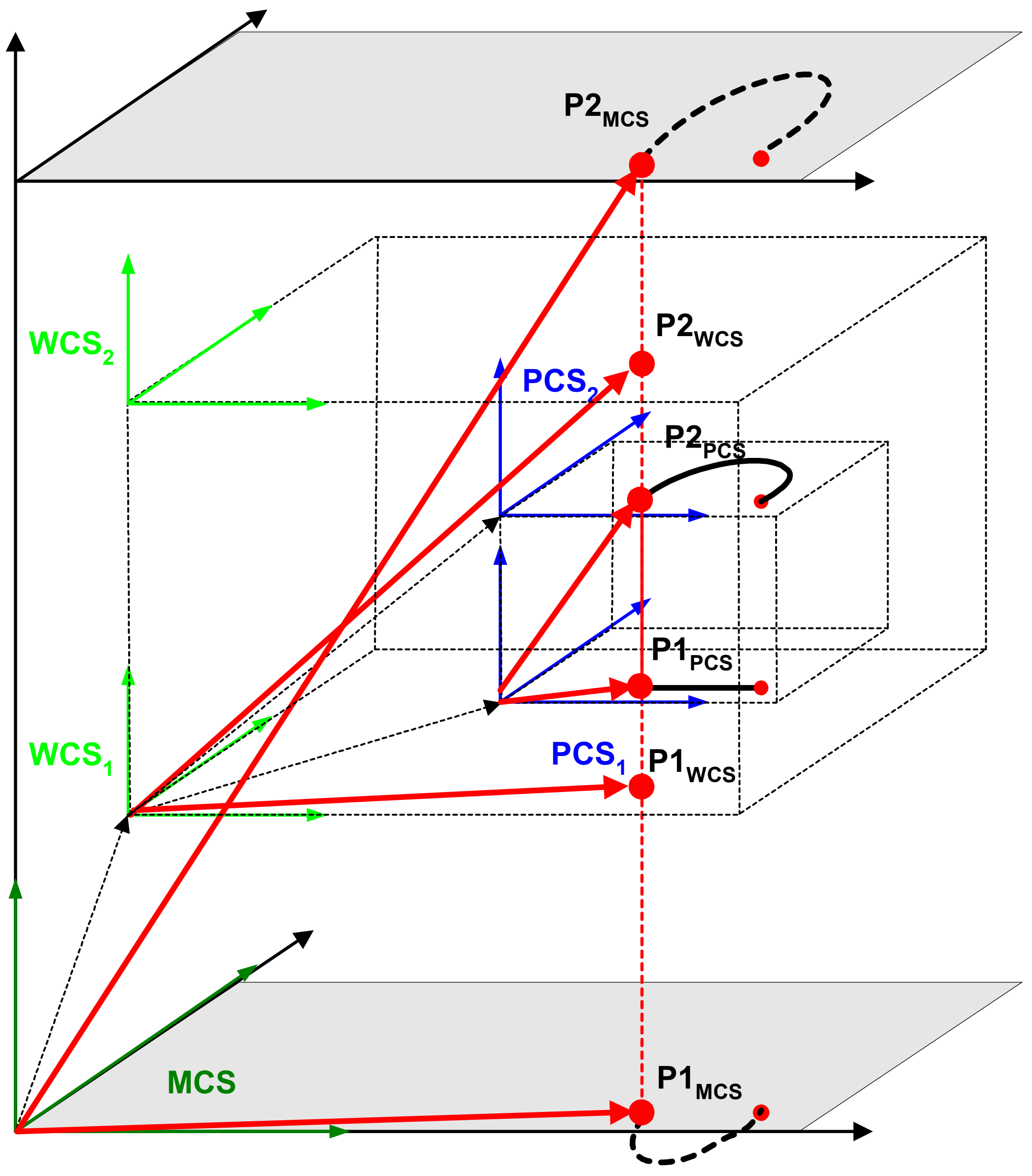


Fig. 52: Display path 2 in CS of path 1 (display_global = 1)

### 1.84.5 2-path programming: Selection of reference coordinate system (P-CHAN-00396)

| **P-CHAN-00396** | **Select the reference coordinate system to define the coordinate system of the second path** |
|---|---|
| Description | When this parameter is initialised with the value 0, the coordinate system of the second path is defined with respect to the coordinate system of the first path belonging to the lower-level coordinate system.<br>When this parameter is initialised with the value 1, the coordinate system of the second path is defined with respect to the coordinate system of the first path. |
| Parameter | coordinate_system.2nd_path_on_actual_1st_path |
| Data type | BOOLEAN |

| | |
|---|---|
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

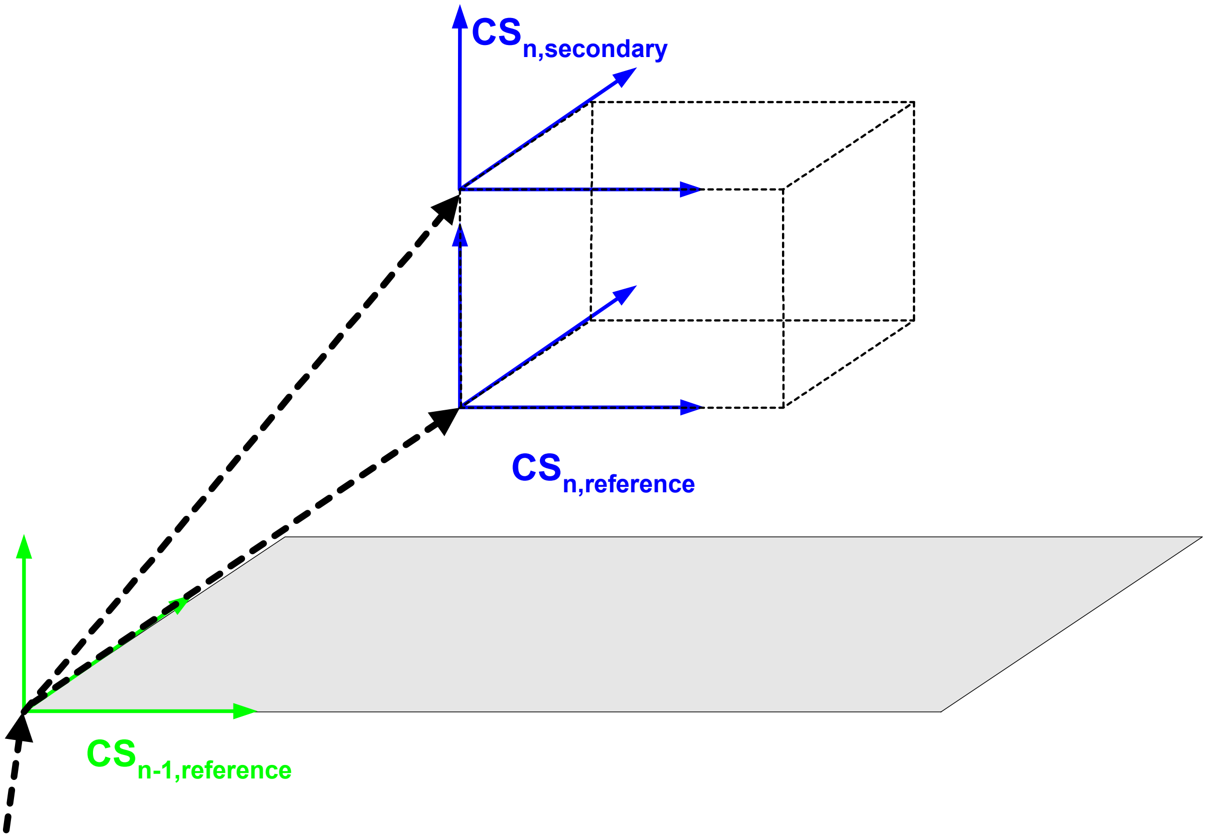


Fig. 53: CS of path 2 sets on basic CS of path 1 on (2nd_path_on_actual_1st_path = 0)

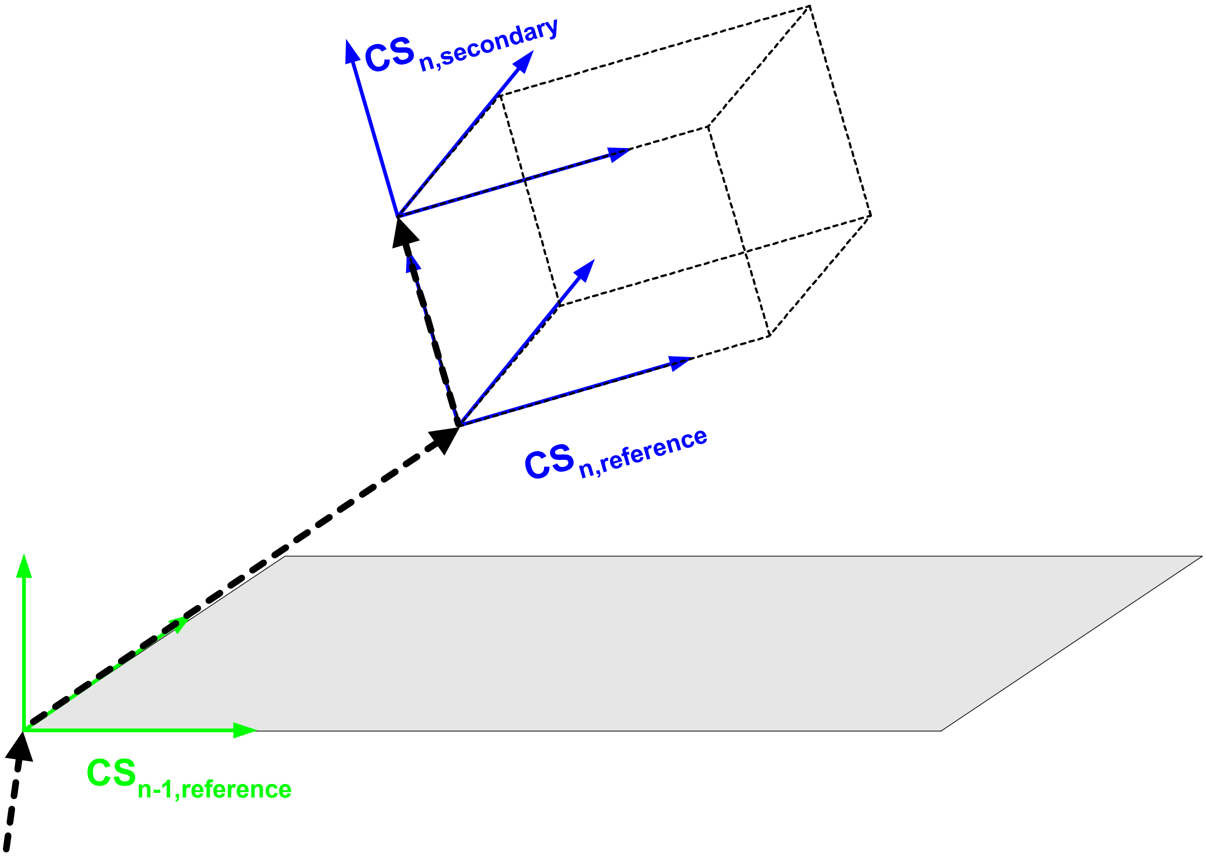
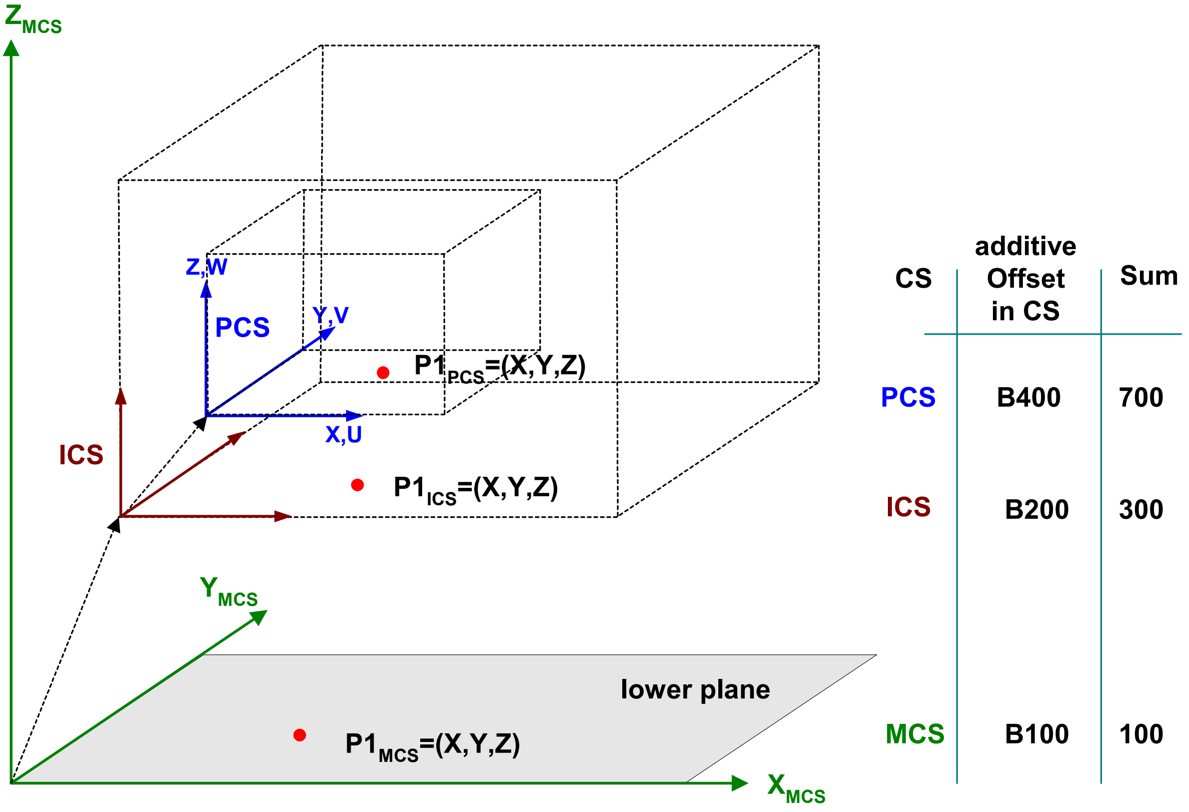


Fig. 54: CS of path 2 refers to CS of path 1 (2nd_path_on_actual_1st_path = 1)

## 1.84.6 2-path programming: Managing axis offsets in tracked axes (P-CHAN-00397)

| P-CHAN-00397 | Manage axis offsets in the tracked axes for a specific coordinate system |
|---|---|
| Description | The axis offsets of tracked axes are managed depending on this parameter. |
| Parameter | coordinate_system.axes_offsets_layer_specific |
| Data type | BOOLEAN |
| Data range | 0: Axis offsets are not managed for a specific coordinate system<br>1: Axis offsets are managed for a specific coordinate system |
| Dimension | ---- |
| Default value | 1 |
| Remarks | |



| CS | additive Offset in CS | Sum |
|---|---|---|
| PCS | B400 | 700 |
| ICS | B200 | 300 |
| MCS | B100 | 100 |

Fig. 55: Offsets of tracked axes in Cartesian coordinate systems

## 1.84.7 2-path programming: Selection/deselection of intersection point calculation (P-CHAN-00398)

| P-CHAN-00398 | Select/deselect calculation of intersecting points with coordinate system planes with 2-path programming |
|---|---|
| Description | With 2-path programming, the point where the tool (e.g. an eroding wire) penetrates through the defined coordinate system planes in the forward direction is calculated automatically (by the default setting). As a result, the Z coordinate is set implicitly to ZERO in the relevant coordinate system. In the backward direction, this intersection point calculation is not performed by default.<br>This parameter activates and deactivates intersection point calculation for Cartesian transformation in the forward and backward directions. |
| Parameter | coordinate_system.intersection |
| Data type | UNS32 |

| | |
|---|---|
| Data range | 1: Intersection point with X, Y and U, V planes with forward transformation<br>2: Intersection point with X, Y and U, V planes with backward transformation |
| Dimension | ---- |
| Default value | 1 |
| Remarks | Intersection point calculation with the coordinate system planes: |

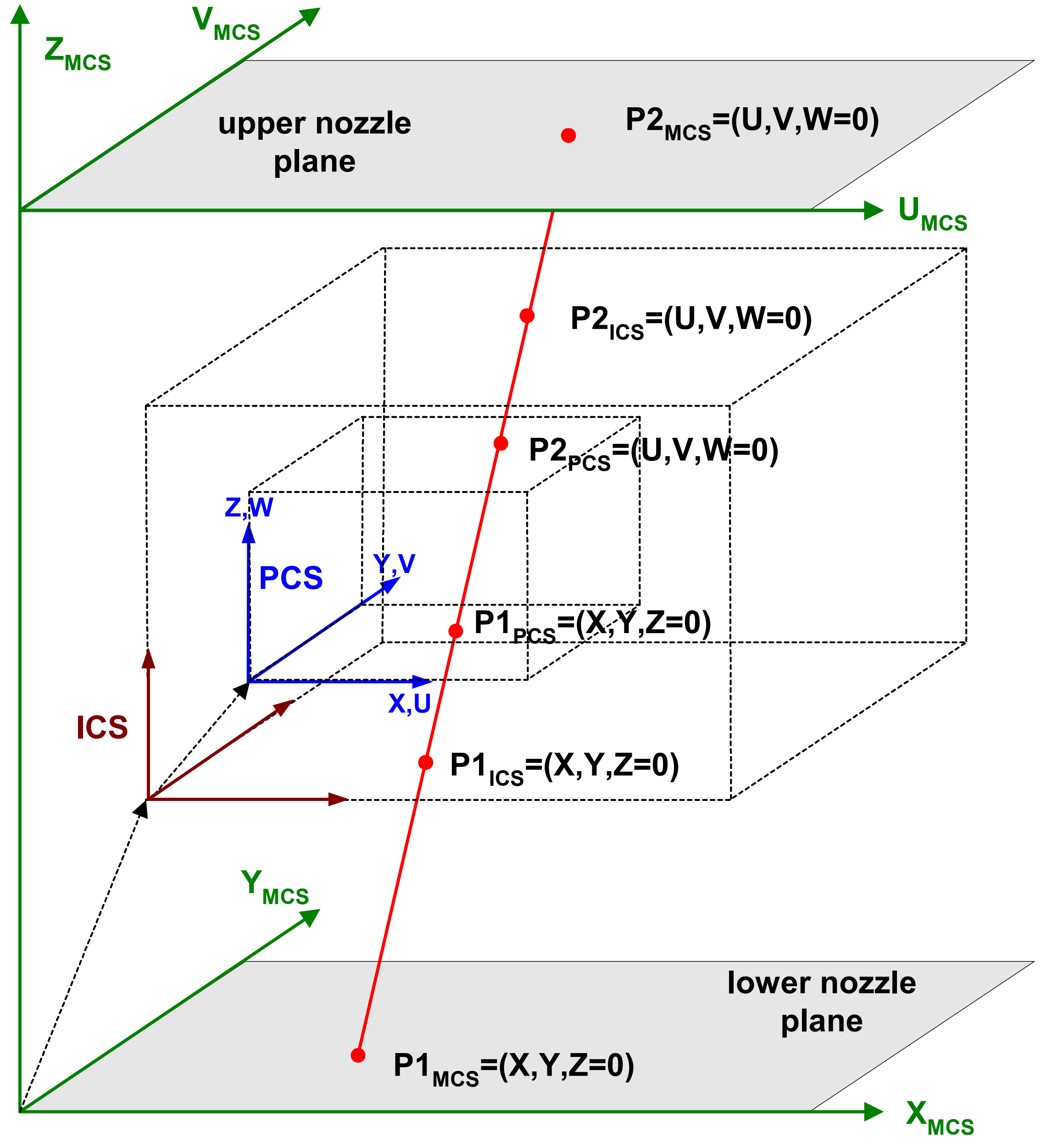
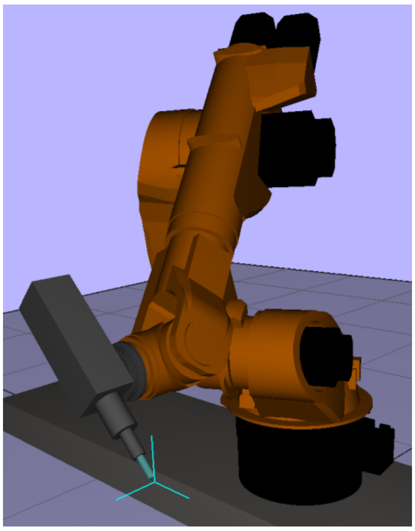
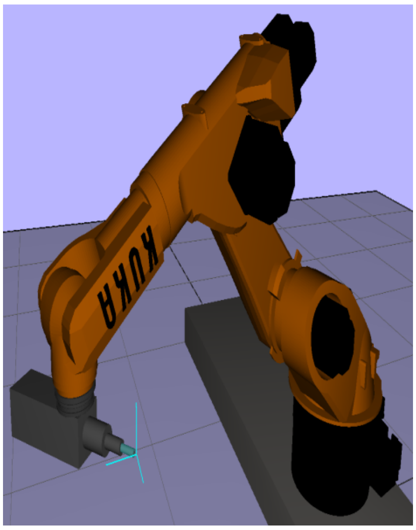


Fig. 56: Intersection point calculation with the coordinate system planes

## 1.85 Response of G91 with orientation axes (P-CHAN-00332)

| | |
|---|---|
| **P-CHAN-00332** | **Response of G91 with orientation axes for complete transformation** |

| | |
|---|---|
| Description | In case of a kinematic sequence with complete transformation (e.g. robot transformations with kinematics ID 45 or 71), this parameter can be used to set the response of G91 when programming orientation axes.<br>For these kinematic sequences, orientation axes are specially handled according to the concatenation of manual systems and the calculation of absolute angles.<br>With default setting (0), a relatively programmed orientation angle is always executed first in the concatenation sequence (rotation about the corresponding Cartesian main axis). If several axes are programmed relatively, a rotation matrix is defined with the relative rotation angles on the basis of the agreed rotation sequence (A(Z) -> B(Y) -> C(X)).<br>Programming example:<br>N010 G90 G00 A0 B180 C0<br>N020 G91 G00 C45 (1st figure)<br>N030 G91 G00 B90 (2nd figure)<br> <br>As an alternative (1), conventional relative orientation programming in the concatenated coordinate system can be selected. In this case, the programmed relative angles are added to the current orientation angle positions. |
| Parameter | ori_prog |
| Data type | UNS16 |
| Data range | 0: G91 acts in the concatenated stationary Cartesian coordinate system (default)<br>1: G91 acts relatively on the programmed angles (calculation of the absolute rotation angles as with the classical relative programming of axes).<br>2: G91 acts concatenated in the Cartesian manual coordinate system. (available as of Build V3.1.3079.36) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Axis identifiers of a Cartesian coordinate system:<br>X, Y, Z: main axes<br>A, B, C: orientation axes |

## 1.86 Use of axis numbers of master channel (P-CHAN-00282)

| P-CHAN-00282 | Use of axis numbers of master channel in clone channel |
|---|---|
| Description | When parameter AXNR is used in the #DRIVE command, the programmed axis number must be known in the channel. This parameter defines that the axis numbers in the master channel must be be used in a clone channel instead of the axis numbers known in the clone channel. |
| Parameter | drive_cmd_use_physical_axis_number |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.87 Buffering motion blocks to optimise the feed rate profile (P-CHAN-00329)

| P-CHAN-00329 | Buffer motion blocks to optimise the feed rate profile |
|---|---|
| Description | This parameter delays the start of motion (from standstill) by the set number of interrupt cycles. During this time, all motion blocks processed by path preparation are buffered in the interpolator so that the motion can be started at the optimum feed rate profile. |
| Parameter | ipo_start_wait_cycles |
| Data type | UNS32 |
| Data range | 0 ≤ ipo_start_wait_cycles ≤ 10 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.88 Units used in PLC Open functions (P-CHAN-00182)

| P-CHAN-00182 | Units used in PLC Open functions |
|---|---|
| Description | When PLC Open functions (S[MC_..] are programmed, see[PROG//section PLCopen functions]), the corresponding specified units must be used in the default setting. Normally, these are value settings in so-called internal units, such as 0.1µm or mm/s² etc.<br>By setting the parameter 'plcopen_std_unit', the values of PLCopen functions can be programmed in the units usually used in the NC program, such as mm, mm/s, etc. Based on the configured resolution factors (*) these values are then recalculated in the internal controller. |
| Parameter | plcopen_std_unit |
| Data type | BOOLEAN |
| Data range | 0: Specify values of PLCopen functions in the specified internal units (0.1µm or µm/s) (default).<br>1: Define values of PLCopen functions in default units (mm, mm/s], etc.). |
| Dimension | ---- |
| Default value | 0 |

| | |
|---|---|
| Remarks | * As of CNC Build V2.11.2026.09 P-CHAN-00315 [▶ 92] replaces the parameters lin_aufloes, rund_aufloes and spind_aufloes (P-CHAN-00034 [▶ 93], P-CHAN-00035 [▶ 140] and P-CHAN-00036 [▶ 145]). It is recommended to only use P-CHAN-00315 [▶ 92] as of this CNC Build. |

## 1.89 Starting NC programs without initialisation (P-CHAN-00347)

| **P-CHAN-00347** | **Start NC programs without initialisation** |
|---|---|
| Description | When an NC program is started, all settings are initialised and assigned default values. Therefore, settings that were changed by a previous NC program are again reset and do not influence the next main program.<br>However, if settings possibly changed by the previous NC program are to remain active in NC programs started afterwards, this can be achieved by setting this parameter. Processing successively executed NC programs then responds like calling a single NC program with local and global subroutines. |
| Parameter | no_init_prog_start |
| Data type | BOOLEAN |
| Data range | 0: When the program is started, all settings are initialised and assigned default values.<br>1: When the program is started, no initialisations take place and changes of previous NC programs remain active. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.90 Logical number of a NC channel (P-CHAN-00400)

| **P-CHAN-00400** | **Logical number of an NC channel for CNC objects** |
|---|---|

| | |
|---|---|
| Description | This parameter is used to assign a distinctive fixed logical number to the NC channel.<br>The logical number is used to address the channel when CNC objects are read and written or in the case of NC commands with channel reference, e.g. #SIGNAL/ #WAIT.<br><br>**Example with implicit assignment of the channel ID (P-CHAN-00400 = 0)**<br>Channel 1: channel_id = channel index + 1 = 1<br>Channel 2: channel_id = channel index + 1 = 2<br>Channel 3: channel_id = channel index + 1 = 3<br>Channel numbering after deactivation of the 2nd channel:<br>Channel 1: channel_id = channel index + 1 = 1<br>Channel 3: channel_id = channel index + 1 = 2<br><br>**Example with explicit assignment of the channel ID (P-CHAN-00400 != 0)**<br>Channel 1: channel_id = 1<br>Channel 2: channel_id = 2<br>Channel 3: channel_id = 3<br>Channel numbering after deactivation of the 2nd channel:<br>Channel 1: channel_id = 1<br>Channel 3: channel_id = 3<br><br>Channel-specific error messages also receive this number for channel identification.<br>If this parameter is not assigned (default), the channel number is automatically assigned and is derived from channel index + 1. |
| Parameter | channel_id |
| Data type | UNS16 |
| Data range | 0 ... maximum number of channels (see CNC system parameters / 2.4) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

# 1.91 Settings for dynamic contour control (dcc.*)

## 1.91.1 Selecting/cancellation of the dynamic contour control function (P-CHAN-00384)

| **P-CHAN-00384** | **Select/deselect the Dynamic Contour Control function** |
|---|---|
| Description | This parameter selects/deselects the Dynamic Contour Control function. |
| Parameter | dcc.active |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.91.2 Selection of calculation method (P-CHAN-00385)

| **P-CHAN-00385** | **Selection of calculation method (Dynamic Contour Control)** |
|---|---|
| Description | This parameter selects whether the Dynamic Contour Control function in a COM object is calculated or whether the dynamic control algorithm integrated in the CNC is used. |
| Parameter | dcc.call |
| Data type | STRING |
| Data range | EXTERNAL, BUILTIN |
| Dimension | ---- |
| Default value | EXTERNAL |
| Remarks | |

## 1.91.3 REAL64 dynamic contour control input parameters (P-CHAN-00388)

| **P-CHAN-00388** | **REAL64 dynamic contour control input parameters (Dynamic Contour Control)** |
|---|---|
| Description | A total of four REAL64 input parameters can be defined for dynamic contour control. |
| Parameter | dcc.param.f[i] where i = 0…3 |
| Data type | REAL64 |
| Data range | MIN(REAL64) ≤ f[i] ≤ MAX(REAL64) |
| Dimension | ---- |
| Default value | 0.0 |
| Remarks | |

## 1.91.4 SGN32 dynamic contour control input parameters (P-CHAN-00389)

| **P-CHAN-00389** | **SGN32 Dynamic Contour Control input parameters** |
|---|---|
| Description | A total of four SGN32 input parameters can be defined for Dynamic Contour Control. |
| Parameter | dcc.param.i[i] where i = 0…3 |
| Data type | SGN32 |
| Data range | MIN(SGN32) ≤ i[i] ≤ MAX(SGN32) |
| Dimension | ---- |

| | |
|---|---|
| Default value | 0 |
| Remarks | |

# 1.92 Settings for surface feed rate (geo_feed_adapt.*)

## 1.92.1 Selecting/deselecting the constant surface feed rate function (P-CHAN-00386)

| **P-CHAN-00386** | **Select/deselect the constant surface feed rate function** |
|---|---|
| Description | This parameter selects/deselects the constant surface feed rate. |
| Parameter | geo_feed_adapt.active |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.92.2 Selection of calculation method (P-CHAN-00387)

| **P-CHAN-00387** | **Selection of calculation method (surface feed rate)** |
|---|---|
| Description | This parameter selects whether the constant surface feed rate in a COM object is calculated or whether the algorithm integrated in the CNC is used. |
| Parameter | geo_feed_adapt.call |
| Data type | STRING |
| Data range | EXTERNAL, BUILTIN |
| Dimension | ---- |
| Default value | EXTERNAL |
| Remarks | |

## 1.92.3 REAL64 input parameters for surface feed rate (P-CHAN-00390)

| **P-CHAN-00390** | **REAL64 input parameters for the surface feed rate** |
|---|---|
| Description | A total of four REAL64 input parameters can be defined for surface feed rate. |
| Parameter | geo_feed_adapt.param.f[i] where i = 0…3 |
| Data type | REAL64 |
| Data range | MIN(REAL64) ≤ f[i] ≤ MAX(REAL64) |
| Dimension | ---- |
| Default value | 0.0 |
| Remarks | |

## 1.92.4 SGN32 input parameters for surface feed rate (P-CHAN-00391)

| **P-CHAN-00391** | **SGN32 input parameters for surface feed rate** |
|---|---|
| Description | A total of four SGN32 input parameters can be defined for surface feed rate. |
| Parameter | geo_feed_adapt.param.i[i] where i = 0…3 |
| Data type | SGN32 |
| Data range | MIN(SGN32) ≤ i[i] ≤ MAX(SGN32) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.93 NC program paths (path[i].*)

This structure element defines the paths to the NC programs for each channel. The path string, the logical path number, the path type and a priority must be specified for each program path.

**If program paths are defined in the start-up and in the channel parameters, only the paths from the channel parameters are adopted. The program paths from the start-up parameters are skipped.**

Further information on program paths in the start-up parameters: [STUP//NC program paths (path[i].*)]

| Structure name | Index |
|---|---|
| path[i] | i = 0 ... 11 (path index, max. number of program paths in the channel: 12, application-specific) |

### 1.93.1 Path specification (P-CHAN-00401)

| P-CHAN-00401 | Path specification to NC programs |
|---|---|
| Description | This parameter defines the path to the NC programs. The CNC employs this path to open an NC program. |
| Parameter | path[i].dir |
| Data type | STRING |
| Data range | Maximum 256 characters (application-specific) |
| Dimension | ---- |
| Default value | * |
| Remarks | This parameter is only available as of CNC Build V3.1.3052.05 and higher.<br>* Note: The default value of variables is a blank string. |

### 1.93.2 Logical path number (P-CHAN-00402)

| P-CHAN-00402 | Logical path number for a program path |
|---|---|
| Description | This parameter defines a logical path number for the program path. Logical path numbers must be unique within the system. |
| Parameter | path[i].id |
| Data type | UNS16 |
| Data range | 1 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is only available as of CNC Build V3.1.3052.05 and higher. |

### 1.93.3 Path type (P-CHAN-00403)

| P-CHAN-00403 | Type of path of a program path |
|---|---|
| Description | This parameter defines the type of the program path bit-encoded. A path specification may also be used for several path types. |
| Parameter | path[i].type |
| Data type | UNS16 |

| | |
|---|---|
| Data range | 0x01 (main program path)<br>0x02 (subroutine path)<br>0x04 (path for #MSG SAVE)<br>0x08 (path for storing debug data *.dbg)<br>Combinations:<br>0x03 (main prog.+ subroutine path)<br>0x05 (main prog. path + path for #MSG SAVE)<br>0x06 (subroutine path + path for #MSG SAVE)<br>0x07 (main prog. + subroutine path + path for #MSG SAVE)<br>0x0B (main prog. path + subroutine path + path for debug data)<br>0x0F (main prog. path + subroutine path + path for #MSG SAVE and debug data) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is available as of CNC Build V3.1.3052.05 and higher. |

## 1.93.4 Priority (P-CHAN-00404)

| **P-CHAN-00404** | **Priority of a program path** |
|---|---|
| Description | This parameter defines the priority of the program path. The priority defines the sequence in which the directories of the corresponding path types are searched for in the NC program file. The highest priority level is '0'. If a priority is not specified for a given program path, the path is initialised with priority '0'. An error message is output if the same priority is specified for a program path of the same path type. |
| Parameter | path[i].priority |
| Data type | UNS16 |
| Data range | 0 ... MAX(UNS16) |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is only available as of CNC Build V3.1.3052.05 and higher.<br>When the path types 0x04 and 0x08 are set as actual program paths, the priorities must be continued based on the sub program paths. |

## 1.94 Settings for generating debug data (debug.*)

Debugging NC programs on the controller requires the supply of certain additional information. For example, this includes the stored values or calculated results of the parameters and variables programmed in the NC blocks and of the expressions generated from them.

Systems that support debugging of NC programs can access this debug data and can display it accordingly on their user interfaces.

### 1.94.1 Tracing parameters and variables (P-CHAN-00392)

| **P-CHAN-00392** | **Trace parameters and variables (debug data of NC programs)** |
|---|---|
| Description | In systems that support debugging of NC programs, this parameter controls tracing of the values of parameters and variables and the expressions generated from them. Trace data is written to a file together with other information relevant to the NC block.<br>The output path of this file with the *.dbg extension is defined by P-CHAN-00403 [▶ 310] . The debug file name is formed from the name of the main NC main program and the channel number.<br>**Example**:<br>Debug trace of the NC program prg_main.nc started in channel 1 is stored to file:<br>*prg_main.nc_ch1.dbg*<br>As of **CNC Build V3.1.3100** a program enumerator is added to the file name.<br>Debug trace of the NC program prg_main.nc started in channel 1, first start, is stored to file:<br>*prg_main.nc_ch1_pc1.dbg* |
| Parameter | debug.prg_trace |
| Data type | BOOLEAN |
| Data range | 0: Generate no debug trace data (default).<br>1: Create and generate debug trace data in a program-specific file. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | When an NC program is started, any existing older debug file with the same name is deleted.<br>The parameter is available as of Build V3.1.3037.0. |

## 1.95 Configuration of channel interface (provide_channel_interface.*)

### 1.95.1 Enabling channel interface for synchronous CS operations (P-CHAN-00399)

| **P-CHAN-00399** | **Automatic enable of channel interface for synchronous dynamic CS operations** |
|---|---|
| Description | This parameter automatically activates the supply of data to the dynCS channel interface at program start. This corresponds to programming the command #CHANNEL INTERFACE ON/OFF [DYN_CS] see [FCT-C30] in the NC program. |
| Parameter | provide_channel_interface.track_cs |
| Data type | BOOLEAN |

| | |
|---|---|
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

# 1.96 Settings for 'Time preview' function

## 1.96.1 Reserved distance for time calculation (P-CHAN-00307)

| **P-CHAN-00307** | **Time preview: Reserved distance for time calculation** |
|---|---|
| Description | The time preview function allows the calculation of the time that the CNC needs to travel a certain distance, taking into consideration the current speed profile. This parameter specifies the distance.<br>The calculated time is output on the PLC interface<br>The specified distance refers to the tool centre point path in the default setting but can be switched to the workpiece contour by P-CHAN-00340 [▸ 313] .<br>The parameter value 0 deactivates the time preview. |
| Parameter | position_lookahead_distance |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | |

## 1.96.2 Relating the reserved distance for time preview to contour (P-CHAN-00340)

| **P-CHAN-00340** | **Relating the reserved distance for time preview to contour.** |
|---|---|
| Description | The time preview function permits the calculation of the time required by the CNC to travel a certain distance (P-CHAN-00307 [▸ 313]) taking into consideration the current velocity profile.<br>In the basic setting, the specified distance refers to the tool centre point path. By setting this parameter, the path of the tool engagement point is related to the contour for the time preview. |
| Parameter | position_lookahead_contour_path |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | - |
| Default value | 0 |
| Remarks | |

# 1.97 Merge short blocks for optimised contouring (P-CHAN-00321)

| **P-CHAN-00321** | **Merge short blocks for optimised contouring** |
|---|---|
| Description | This parameter merges single blocks which are shorter than the specified tolerance to the next adjacent block. Merging provides contour smoothing operations with better results since they require fewer and longer motion blocks. |
| Parameter | block_filter_tolerance |

| | |
|---|---|
| Data type | REAL64 |
| Data range | 0 ≤ block_filter_tolerance < MAX(REAL64) |
| Dimension | 0.1µm |
| Default value | 0.0 |
| Remarks | To activate this function, the parameter P-STUP-00060 in the start-up list must contain the following entry.<br>configuration.channel[0].path_preparation.function FCT_DEFAULT\|**FCT_PRECON** |

## 1.98 Contouring method in inch units (P-CHAN-00439)

| **P-CHAN-00439** | **Contouring method in inch units** |
|---|---|
| Description | So far, certain parameter values could only be specified in mm for contour smoothing methods with an active G70, e.g. #CONTOUR MODE, #HSC [BSPLINE..] or #HSC[SURFACE..].<br>This compatibility parameter now permits the specification of values in inches with an active G70. |
| Parameter | contouring_consider_inch |
| Data type | BOOLEAN |
| Data range | 0: Specify certain parameters only in [mm] with active G70.<br>1: Specify certain parameters only in [inch] with active G70. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Parameter available as of Build V2.11.2036.3. Only acts when G70 is active. |

## 1.99 Unit of constant cutting velocity (P-CHAN-00360)

| **P-CHAN-00360** | **Unit of constant cutting velocity** |
|---|---|
| Description | The constant cutting velocity programmed with G06 is always specified in the unit m/min irrespective of G70 (inch) or G71 (metric).<br>If this parameter is set to 1, the value of constant cutting velocity must be specified in ft/min when G70 is active (inch programming). |
| Parameter | enable_unit_feet_cut_speed |
| Data type | BOOLEAN |
| Data range | 0: Constant cutting velocity is always programmed in m/min.<br>1: Constant cutting velocity is programmed in ft/min when G70 is active. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## Synchronisation of PLCopen jobs (P-CHAN-00359)

| **P-CHAN-00359** | **Synchronisation of PLCopen jobs** |
|---|---|

| | |
|---|---|
| Description | This parameter causes an explicit synchronisation of the PLCopen assigned by the NC program to a single axis.<br>Program processing of the next NC line is only continued when the PLCopen job ends.<br>If an MC_MoveAbsolute command is used simultaneously with a path motion where the single axis movement lasts longer than the path motion, a motion stop occurs on the path.<br>The behaviour of the set parameter plcopen_implicit_sync is the identical with the programming of the keyword WAIT_SYN in each PLCopen command.<br>If P-CHAN-00359 is not set, there is no automatic synchronisation of the NC program flow with PLCopen single axis commands. Therefore, synchronisation must be explicit with the keyword WAIT_SYN when the PLCopen command is generated or when the command #WAIT MC_STATUS SYN [ID] occurs in the NC program. |
| Parameter | plcopen_implicit_sync |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.100 Orientation interpolation mode (P-CHAN-00417)

| **P-CHAN-00417** | **Orientation interpolation mode in conjunction with complete kinematic transformations** |
|---|---|
| Description | With a complete transformation the tool target orientation is defined by concatenated rotations in space.<br>If more than one rotary axis is programmed, it may cause the tool to wobble. This is prevented by using a special angular interpolation (orientation interpolation).<br>This parameter influences the mode of functioning of orientation interpolation. |
| Parameter | ori_interpolation_mode |
| Data type | SGN16 |
| Data range | 0: Orientation interpolation of tool axis (default). This is executed in a place which results from the tool direction at the start and target points of the programmed angular motion. While rotating in the plane, the tool rotates about the tool axis. This mode is used to program relative movements of the Cartesian orientation axes at rotation angles phi < \|180°\|. Larger relative angles generally lead to "shortest way" treatment of the orientation movement.<br>1: Orientation interpolation with a spatial axis calculated by the CNC. This spatial axis executes a rotation from start orientation to target orientation in the manual system. This mode is used to program rotation angles phi > \|180°\| when the orientation is programmed relatively ( 0° < phi < \|360°\|).<br>-1: Orientation interpolation inactive, linear interpolation of the programmed spatial angle. |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Geometric smoothing functions such as polynomial smoothing, spline etc. deactivate the effect of orientation interpolation. |

## 1.101 Axis-specific orientation of tool length compensation (P-CHAN-00420)

| **P-CHAN-00420** | **Axis-specific orientation of tool length compensation** |
|---|---|
| Description | This parameter defines whether the tool length remains considered when the plane in the momentary axis changes or is automatically adopted in the new third main axis after a plane change.<br>This pre-setting can be changed in the NC program by the command #TLAX , |
| Parameter | remain_tool_length_in_ax |
| Data type | BOOLEAN |
| Data range | 0: Tool length is automatically adopted in the new third main axis after plane change (default).<br>1: Tool length always remains in the third main axis 'Z' of the G17 plane |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.102 Deactivate feed factor effect (P-CHAN-00422)

| **P-CHAN-00422** | **Deactivate feed factor effect** |
|---|---|
| Description | In the NC program, the feed factor #FF is additionally weighted by the path feed commanded by the PLC [HLI// Control commands of a channel]. Weighting the PLC feed can be deactivated by setting the parameter P-CHAN-00422. #FF is then disabled. |
| Parameter | disable_feed_factor |
| Data type | BOOLEAN |
| Data range | 0: The feed factor #FF can be weighted in the NC program by the path feed commanded by the PLC (default).<br>1: The feed factor #FF is disabled |
| Dimension | ---- |
| Default value | 0 |
| Remarks | The feed factor #FF only acts on the external feed. |

## 1.103 Settings for precalculation

### 1.103.1 Precalculation - time offset (P-CHAN-00324)

| **P-CHAN-00324** | **Default offset time to calculate future states** |
|---|---|
| Description | At a set time greater than 0, the<br>• path velocity<br>• attempts to calculate axis position, velocity and acceleration<br>at the parameterised point in the future. |
| Parameter | esa.time[i] where i = 0 ... 9 |
| Data type | REAL64 |
| Data range | 0 ≤ time[i] ≤ MAX_REAL64 |
| Dimension | s |
| Default value | 0.0 |
| Remarks | The maximum possible number of entries is limited to 10.<br>Axis position, velocity and acceleration only estimated with the entry esa.time[0]. |

### 1.103.2 Precalculation mode (P-CHAN-00325)

| P-CHAN-00325 | Precalculation mode |
|---|---|
| Description | This parameter sets the precalculation mode.<br>• Mode 1: Precalculation of path velocity at up to 10 future points in time<br>• Mode 2: in addition to Mode 1, the precalculation of axis positions, velocities and accelerations of all the axes in the channel is conducted for the first time entry |
| Parameter | esa.mode |
| Data type | UNS32 |
| Data range | 1 / 2 |
| Dimension | ---- |
| Default value | 1 |
| Remarks | |

## 1.104 Dynamic look-ahead for axis polynomials (P-CHAN-00453)

## 1.105 Memory size for contour machining (P-CHAN-00467)

| P-CHAN-00467 | Memory size for contour machining |
|---|---|
| Description | This parameter is used to extend the memory area to save and edit contours (#CONTOUR BEGIN/#CONTOUR END). The value is assigned as bytes. |
| Parameter | contour_processing_memory |
| Data type | UNS32 |
| Data range | 0 < P-CHAN-00467 < MAX_UNS32 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | |

## 1.106 Real-time cycles

### 1.106.1 Activating real-time cycles (P-CHAN-00406)

| P-CHAN-00406 | Activating real-time cycles |
|---|---|
| Description | This parameter enables the real-time cycle function in the NC channel.<br>The controller must be restarted to adopt the change.<br>Example:<br>`configuration.rt_cycles.enable 1` |
| Parameter | configuration.rt_cycles.enable |
| Data type | BOOLEAN |
| Data range | 0/1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | This parameter is available as of CNC Build V3.1.3107.10 or higher.<br>Use of the parameter “rt_cycles.enable“<br>`rt_cycles.enable 1`<br>(as of V3.1.3105) continues to be supported. |

## 1.106.2 Memory for real-time cycles (P-CHAN-00407)

| P-CHAN-00407 | Memory size for real-time cycles |
|---|---|
| Description | This parameter defines the size of the memory for real-time cycles. The size of the memory is specified in bytes.<br>The controller must be restarted to adopt the change. Then the specified memory is also available for real-time cycles.<br>Example:<br>`configuration.rt_cycles.memory 60000` |
| Parameter | configuration.rt_cycles.memory |
| Data type | UNS32 |
| Data range | 0 ... MAX(UNS32) - 1 |
| Dimension | ---- |
| Default value | 48000 |
| Remarks | **Note:**<br>The assignment of P-CHAN-00407 is only necessary if the memory set by default is no longer sufficient due to activation of the real-time cycles (P-CHAN-00406 [▶ 317]).<br>This parameter is available as of CNC Build V3.1.3107.10 or higher.<br>Use of the parameter “rt_cycles.memory“<br>`rt_cycles.memory 60000`<br>(as of V3.1.3105) continues to be supported. |

## 1.106.3 Max. execution time of realtime cycles (P-CHAN-00425)

| P-CHAN-00425 | Max. execution time of realtime cycles per CNC cycle |
|---|---|
| Description | This parameter defines the maximum execution time of realtime cycles in the NC channel. The parameter is specified in percent (%) and refers to the length of the CNC cycle.<br>Example:<br>If the realtime task of the CNC is clocked at 2ms and parameter P-CHAN-00425 is 75, the realtime cycles may require a total execution time of maximum 1.5ms. If this time is exceeded, error ID 50939 is output. |
| Parameter | rt_cycles.max_duration |
| Data type | UNS16 |
| Data range | 0 < P-CHAN-00425 < MAX_UNS16 |
| Dimension | % |
| Default value | 75 |
| Remarks | The user has no restrictions regarding the number of instructions within a realtime cycle.<br>Realtime timeouts will occur if realtime cycles contain too many instructions and cannot be executed in one CNC cycle,<br>The parameters P-CHAN-00426 [▶ 318] and P-CHAN-00427 [▶ 319] represent a safety mechanism to prevent such realtime timeouts as early as possible. |

## 1.106.4 Number of elementary instructions for time check (P-CHAN-00426)

| P-CHAN-00426 | Number of elementary instructions for time check |
|---|---|
| Description | This parameter defines the number of elementary instructions after which another time check is executed. |

| | |
|---|---|
| | A check for the execution time of realtime cycles must be made within a CNC cycle to see whether the permitted execution time is already exceeded. This takes place by checking the expired time after a specific number of elementary instructions in a cycle. Parameter P-CHAN-00426 indicates the number of these elementary instructions. |
| Parameter | rt_cycles.cont_steps |
| Data type | UNS32 |
| Data range | 0 < P-CHAN-00426 < MAX_UNS32 |
| Dimension | ---- |
| Default value | 100 |
| Remarks | The user has no restrictions regarding the number of instructions within a realtime cycle.<br>Realtime timeouts will occur if realtime cycles contain too many instructions and cannot be executed in one CNC cycle,<br>The parameters P-CHAN-00425 [▸ 318] and P-CHAN-00427 [▸ 319] represent a safety mechanism to prevent such realtime timeouts as early as possible. |

## 1.106.5 Max. number of elementary instructions per CNC cycle (P-CHAN-00427)

| **P-CHAN-00427** | **Max. number of elementary instructions per CNC cycle** |
|---|---|
| Description | This parameter defines the maximum number of elementary instructions per CNC cycle.<br>P-CHAN-00427 as well as P-CHAN-00425 [▸ 318] and P-CHAN-00426 [▸ 318] can limit the execution time in realtime cycles in the CNC cycle.<br>If the number of elementary instructions in the current CNC cycle exceeds the value of this parameter, error ID 50854 is output. |
| Parameter | rt_cycles.max_steps |
| Data type | UNS32 |
| Data range | 0 < P-CHAN-00427 < MAX_UNS32 |
| Dimension | ---- |
| Default value | MAX_UNS32 - 1 |
| Remarks | The user has no restrictions regarding the number of instructions within a realtime cycle.<br>Realtime timeouts will occur if realtime cycles contain too many instructions and cannot be executed in one CNC cycle,<br>The parameters P-CHAN-00425 [▸ 318] and P-CHAN-00426 [▸ 318] represent a safety mechanism to prevent such realtime timeouts as early as possible. |

## 1.106.6 Max. number of actions in a real-time cycle (P-CHAN-00480)

| **P-CHAN-00480** | **Max. number of actions in a real-time cycle** |
|---|---|
| Description | This parameter defines the maximum number of possible actions within a real-time cycle.<br>Possible actions include single-axis movement, spindle command, etc.<br>If too many actions are commanded within a real-time cycle, error ID 51028 is output. |
| Parameter | configuration.rt_cycles.buffers |
| Data type | UNS16 |
| Data range | 0 ... MAX(UNS16) - 1 |
| Dimension | ---- |

| | |
|---|---|
| Default value | 5 |
| Remarks | The parameter is available as of V3.1.3107.10 |

## 1.107 Compatibility for measuring with #CS/Transformation (P-CHAN-00440)

| P-CHAN-00440 | Mode of measuring function with coordinate system / kinematics |
|---|---|
| Description | This parameter is used to influence the mode of operation of the measuring function in connection with active coordinate systems and / or kinematics.<br>Value 0:<br>When measuring with a coordinate system (e.g. #CS[]), the three ACS axes corresponding to the coordinate system axes must be parameterised as measuring axes. The measuring signal must be looped through so that all three main axes are latched during the measuring process and the CNC can calculate a PCS measuring point with consistent values in X, Y, Z in space.<br>When measuring with active kinematics (e.g. #TRAFO), the ACS axes affected by the kinematics must be parameterised as measuring axes. The measuring signal must be looped through so that all ACS axes of the kinematics are latched during the measuring process and the CNC can calculate a PCS measuring point X, Y, Z in space.<br>If no measured value is provided in one of the ACS measuring axes affected by the coordinate system or the kinematics, or if it was not parameterised as a measuring axis, the CNC generates a warning message that the PCS measuring position cannot be calculated.<br>Value 1:<br>Use the current ACS command value for ACS axes that are not measuring axes and do not move during the measurement run. The CNC can then calculate the correct PCS measuring point X, Y, Z in space.<br>If ACS measuring axes of the coordinate system or the kinematics move in the measuring block and no measured value is provided, the CNC generates a warning message. The PCS measuring position cannot be calculated. |
| Parameter | meas_trafo_with_cmd_pos |
| Data type | BOOLEAN |
| Data range | 0 / 1 |
| Dimension | ---- |
| Default value | 0 |
| Remarks | Available as of V2.2039.01 |

## 1.108 Calculation mode of resulting ramp time with #CS and #TRAFO (P-CHAN-00758)

| P-CHAN-00758 | Mode of calculation of the resulting ramp time when coordinate systems and transformations are used |
|---|---|
| Description | This parameter affects use of axis jerk with coordinate systems (#CS) and transformations (#TRAFO).<br>Optimisation must be explicitly enabled to avoid different program behaviour when the CNC Release is updated.<br>Optimisation is enabled by the value 0. |
| Parameter | backward_compatibility.axis_spline_ramp_time |

| | |
|---|---|
| Data type | BOOLEAN |
| Data range | 0: Ramp time is selected dependent on the ramp time of the main axes involved and on the portion of the movement.<br>1: Ramp time is selected dependent on the longest ramp time of the main axes involved (default). |
| Dimension | --- |
| Default value | 1 |
| Remarks | Parameter available as of V3.1.3079.40<br>It is recommended to set this parameter to the value 0 to attain a better utilisation of the axis dynamic. |

## 1.109 Tolerance limit for recalculation with dynamic limitation of axis positions (P-CHAN-00751)

| **P-CHAN-00751** | **Tolerance limit for recalculation with dynamic limitation of axis positions** |
|---|---|
| Description | With dynamic limitation of axis positions, changes to the specified limitation lead to a recalculation of the braking point<br>Minor changes to the specified limitation lead to unnecessary recalculations.<br>This parameter defines a tolerance limit up to which a recalculation is not executed. |
| Parameter | dpl_tol_limit_change |
| Data type | UNS32 |
| Data range | 0 ≤ P-CHAN-00751 < MAX(UNS32) |
| Dimension | 0.1µm |
| Default value | 0 |
| Remarks | Parameter available as of CNC Build V3.1.3079.28 and higher |

## 1.110 Settings for thread tapping

### 1.110.1 Tapping with actual positions of the spindle (P-CHAN-00761)

| **P-CHAN-00761** | **Tapping with actual positions of the spindle** |
|---|---|
| Description | An additional function for tapping can be enabled to take into account the actual spindle speed.<br>The linear axes are then coupled to the actual spindle position. This also permits tapping when the spindle speed drops due to the load. |
| Parameter | tapping.use_actual_position |
| Data type | BOOLEAN |
| Data range | 0: The spindle is coupled to the command positions of the linear axes.<br>1: The linear axes are coupled to the actual spindle positions. |
| Dimension | --- |
| Default value | 0 |
| Remarks | Parameter available as of V3.1.3080.04 |

### 1.110.2 Number of filter cycles to filter actual spindle positions (P-CHAN-00762)

| **P-CHAN-00762** | **Number of filter cycles to filter actual spindle positions** |
|---|---|

| | |
|---|---|
| Description | The actual spindle positions may be noisy under certain circumstances. This parameter activates a filter to smooth the actual positions, The parameter specifies the number of filter smoothing cycles.<br>The value 0 deactivates the filter. |
| Parameter | tapping.n_cycles |
| Data type | UNS16 |
| Data range | 0 <= n_cycles <= 20 |
| Dimension | --- |
| Default value | 0 |
| Remarks | P-CHAN-00762 only acts when tapping with actual spindle positions is active (P-CHAN-00761 [▶ 321]).<br>Parameter available as of V3.1.3080.04 |

# 2 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

### Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

### Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

### Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

| | |
|---|---|
| Hotline: | +49 5246 963-157 |
| e-mail: | support@beckhoff.com |

### Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

| | |
|---|---|
| Hotline: | +49 5246 963-460 |
| e-mail: | service@beckhoff.com |

### Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

| | |
|---|---|
| Phone: | +49 5246 963-0 |
| e-mail: | info@beckhoff.com |
| web: | www.beckhoff.com |

# Index

P

P-CAHN-00501 41
P-CHAN-00001 194
P-CHAN-00002 127
P-CHAN-00003 139
P-CHAN-00004 153
P-CHAN-00005 139
P-CHAN-00006 139
P-CHAN-00007 145
P-CHAN-00008 142
P-CHAN-00009 198
P-CHAN-00010 142
P-CHAN-00011 140
P-CHAN-00012 201
P-CHAN-00013 201
P-CHAN-00014 94
P-CHAN-00015 115
P-CHAN-00016 95
P-CHAN-00017 200
P-CHAN-00018 201
P-CHAN-00019 163
P-CHAN-00020 163
P-CHAN-00021 161
P-CHAN-00022 159
P-CHAN-00023 138
P-CHAN-00024 158
P-CHAN-00025 85
P-CHAN-00026 86
P-CHAN-00027 83
P-CHAN-00028 163
P-CHAN-00029 99
P-CHAN-00030 99
P-CHAN-00031 130
P-CHAN-00032 165
P-CHAN-00033 196
P-CHAN-00034 93
P-CHAN-00035 140
P-CHAN-00036 145
P-CHAN-00037 156
P-CHAN-00038 156
P-CHAN-00039 82
P-CHAN-00040 82
P-CHAN-00041 65
P-CHAN-00042 152
P-CHAN-00043 149
P-CHAN-00044 151
P-CHAN-00045 148
P-CHAN-00046 151
P-CHAN-00047 148
P-CHAN-00048 152
P-CHAN-00049 149
P-CHAN-00050 130
P-CHAN-00051 143
P-CHAN-00052 143
P-CHAN-00053 143
P-CHAN-00054 92
P-CHAN-00055 154
P-CHAN-00056 122
P-CHAN-00057 107
P-CHAN-00058 154
P-CHAN-00059 117
P-CHAN-00060 118
P-CHAN-00061 157
P-CHAN-00062 160
P-CHAN-00063 191
P-CHAN-00064 193
P-CHAN-00065 122
P-CHAN-00066 123
P-CHAN-00067 115
P-CHAN-00068 158
P-CHAN-00069 146
P-CHAN-00070 81
P-CHAN-00071 194
P-CHAN-00072 87
P-CHAN-00073 194
P-CHAN-00074 153
P-CHAN-00075 121
P-CHAN-00076 97
P-CHAN-00077 88
P-CHAN-00078 93
P-CHAN-00079 92
P-CHAN-00080 151
P-CHAN-00081 147
P-CHAN-00082 142
P-CHAN-00083 94
P-CHAN-00085 160
P-CHAN-00086 86
P-CHAN-00087 95
P-CHAN-00088 158
P-CHAN-00089 200
P-CHAN-00090 129
P-CHAN-00091 115
P-CHAN-00092 161
P-CHAN-00093 160
P-CHAN-00094 166
P-CHAN-00095 133
P-CHAN-00096 135
P-CHAN-00097 136
P-CHAN-00098 136
P-CHAN-00099 195
P-CHAN-00100 137
P-CHAN-00101 133
P-CHAN-00102 127
P-CHAN-00103 95
P-CHAN-00104 157
P-CHAN-00105 157
P-CHAN-00106 97
P-CHAN-00107 84
P-CHAN-00108 196
P-CHAN-00109 133
P-CHAN-00110 198
P-CHAN-00111 126
P-CHAN-00112 181
P-CHAN-00113 203
P-CHAN-00114 203
P-CHAN-00115 121
P-CHAN-00116 205
P-CHAN-00117 199
P-CHAN-00118 206

P-CHAN-00119 208
P-CHAN-00120 211
P-CHAN-00121 283
P-CHAN-00122 216
P-CHAN-00123 216
P-CHAN-00124 216
P-CHAN-00125 216
P-CHAN-00126 217
P-CHAN-00127 217
P-CHAN-00128 217
P-CHAN-00129 217
P-CHAN-00130 218
P-CHAN-00131 218
P-CHAN-00132 220
P-CHAN-00133 220
P-CHAN-00134 221
P-CHAN-00135 221
P-CHAN-00136 221
P-CHAN-00137 221
P-CHAN-00138 222
P-CHAN-00139 222
P-CHAN-00140 222
P-CHAN-00141 223
P-CHAN-00142 223
P-CHAN-00143 223
P-CHAN-00144 134
P-CHAN-00145 182
P-CHAN-00146 214
P-CHAN-00147 212
P-CHAN-00148 206
P-CHAN-00149 100
P-CHAN-00150 214
P-CHAN-00151 188
P-CHAN-00152 188
P-CHAN-00153 101
P-CHAN-00154 224
P-CHAN-00155 201
P-CHAN-00156 88
P-CHAN-00157 88
P-CHAN-00158 225
P-CHAN-00159 226
P-CHAN-00160 226
P-CHAN-00161 227
P-CHAN-00162 227
P-CHAN-00163 228
P-CHAN-00164 229
P-CHAN-00165 229
P-CHAN-00166 230
P-CHAN-00167 231
P-CHAN-00168 231
P-CHAN-00169 232
P-CHAN-00170 123
P-CHAN-00171 118
P-CHAN-00172 118
P-CHAN-00173 233
P-CHAN-00174 282
P-CHAN-00175 282
P-CHAN-00176 108
P-CHAN-00177 260
P-CHAN-00178 260
P-CHAN-00179 120
P-CHAN-00180 282
P-CHAN-00181 124
P-CHAN-00182 304
P-CHAN-00183 235
P-CHAN-00184 184
P-CHAN-00185 134
P-CHAN-00186 203
P-CHAN-00187 233
P-CHAN-00188 184
P-CHAN-00189 184
P-CHAN-00190 236
P-CHAN-00191 236
P-CHAN-00192 237
P-CHAN-00193 237
P-CHAN-00194 237
P-CHAN-00195 242
P-CHAN-00196 242
P-CHAN-00197 242
P-CHAN-00198 243
P-CHAN-00199 243
P-CHAN-00200 243
P-CHAN-00208 128
P-CHAN-00209 88
P-CHAN-00210 244
P-CHAN-00211 244
P-CHAN-00212 90
P-CHAN-00213 185
P-CHAN-00214 108
P-CHAN-00215 134
P-CHAN-00216 254
P-CHAN-00217 223
P-CHAN-00218 235
P-CHAN-00219 161
P-CHAN-00220 250
P-CHAN-00221 250
P-CHAN-00222 250
P-CHAN-00223 250
P-CHAN-00224 251
P-CHAN-00225 251
P-CHAN-00226 251
P-CHAN-00227 102
P-CHAN-00228 187
P-CHAN-00230 239
P-CHAN-00231 239
P-CHAN-00232 240
P-CHAN-00233 240
P-CHAN-00234 240
P-CHAN-00235 257
P-CHAN-00239 218
P-CHAN-00240 219
P-CHAN-00241 219
P-CHAN-00243 98
P-CHAN-00244 258
P-CHAN-00245 199
P-CHAN-00246 254
P-CHAN-00247 186
P-CHAN-00249 265
P-CHAN-00250 264
P-CHAN-00251 264
P-CHAN-00252 210
P-CHAN-00253 262
P-CHAN-00254 263
P-CHAN-00255 98
P-CHAN-00256 283
P-CHAN-00257 103

P-CHAN-00258 104
P-CHAN-00259 266
P-CHAN-00260 208
P-CHAN-00261 281
P-CHAN-00262 168
P-CHAN-00263 169
P-CHAN-00264 167
P-CHAN-00265 135
P-CHAN-00266 108
P-CHAN-00267 268
P-CHAN-00268 110
P-CHAN-00269 255
P-CHAN-00270 255
P-CHAN-00271 97
P-CHAN-00273 255
P-CHAN-00274 91
P-CHAN-00275 272
P-CHAN-00276 273
P-CHAN-00277 273
P-CHAN-00278 273
P-CHAN-00279 273
P-CHAN-00280 277
P-CHAN-00281 135
P-CHAN-00282 304
P-CHAN-00283 277
P-CHAN-00284 162
P-CHAN-00285 173
P-CHAN-00286 173
P-CHAN-00287 173
P-CHAN-00288 174
P-CHAN-00289 174
P-CHAN-00290 175
P-CHAN-00291 175
P-CHAN-00292 175
P-CHAN-00293 176
P-CHAN-00294 177
P-CHAN-00295 178
P-CHAN-00296 110
P-CHAN-00297 278
P-CHAN-00298 105
P-CHAN-00299 105
P-CHAN-00300 251
P-CHAN-00301 254
P-CHAN-00305 279
P-CHAN-00306 280
P-CHAN-00307 313
P-CHAN-00308 275
P-CHAN-00309 275
P-CHAN-00310 275
P-CHAN-00311 112
P-CHAN-00312 113
P-CHAN-00313 113
P-CHAN-00315 92
P-CHAN-00316 197
P-CHAN-00317 197
P-CHAN-00318 197
P-CHAN-00319 197
P-CHAN-00320 198
P-CHAN-00321 313
P-CHAN-00324 316
P-CHAN-00325 317
P-CHAN-00326 114
P-CHAN-00327 114
P-CHAN-00328 283
P-CHAN-00329 304
P-CHAN-00330 284
P-CHAN-00331 284
P-CHAN-00332 302
P-CHAN-00338 287
P-CHAN-00339 288
P-CHAN-00340 313
P-CHAN-00341 280
P-CHAN-00343 243
P-CHAN-00344 284
P-CHAN-00345 258
P-CHAN-00346 280
P-CHAN-00347 305
P-CHAN-00349 195
P-CHAN-00350 129
P-CHAN-00351 128
P-CHAN-00353 127
P-CHAN-00359 314
P-CHAN-00360 314
P-CHAN-00361 128
P-CHAN-00375 224
P-CHAN-00378 289
P-CHAN-00379 289
P-CHAN-00380 290
P-CHAN-00381 290
P-CHAN-00382 291
P-CHAN-00383 291
P-CHAN-00384 307
P-CHAN-00385 307
P-CHAN-00386 309
P-CHAN-00387 309
P-CHAN-00388 307
P-CHAN-00389 307
P-CHAN-00390 309
P-CHAN-00391 309
P-CHAN-00392 312
P-CHAN-00393 292
P-CHAN-00394 296
P-CHAN-00395 296
P-CHAN-00396 298
P-CHAN-00397 301
P-CHAN-00398 301
P-CHAN-00399 312
P-CHAN-00400 305
P-CHAN-00401 310
P-CHAN-00402 310
P-CHAN-00403 310
P-CHAN-00404 311
P-CHAN-00405 224
P-CHAN-00406 47, 317
P-CHAN-00407 47, 318
P-CHAN-00415 154
P-CHAN-00416 165
P-CHAN-00417 315
P-CHAN-00418 48
P-CHAN-00419 49
P-CHAN-00420 316
P-CHAN-00421 220
P-CHAN-00422 316
P-CHAN-00423 262
P-CHAN-00424 48
P-CHAN-00425 318

P-CHAN-00426 318
P-CHAN-00427 319
P-CHAN-00428 257
P-CHAN-00429 207
P-CHAN-00433 211
P-CHAN-00434 285
P-CHAN-00435 106
P-CHAN-00436 261
P-CHAN-00437 205
P-CHAN-00438 171
P-CHAN-00439 314
P-CHAN-00440 320
P-CHAN-00442 205
P-CHAN-00443 171
P-CHAN-00446 171
P-CHAN-00447 178
P-CHAN-00448 179
P-CHAN-00449 179
P-CHAN-00450 180
P-CHAN-00451 278
P-CHAN-00452 172
P-CHAN-00454 172
P-CHAN-00455 225
P-CHAN-00456 189
P-CHAN-00457 263
P-CHAN-00458 180
P-CHAN-00459 212
P-CHAN-00460 121
P-CHAN-00463 246
P-CHAN-00464 189
P-CHAN-00465 190
P-CHAN-00466 190
P-CHAN-00467 317
P-CHAN-00468 224
P-CHAN-00469 190
P-CHAN-00470 246
P-CHAN-00471 246
P-CHAN-00472 247
P-CHAN-00473 247
P-CHAN-00474 247
P-CHAN-00475 247
P-CHAN-00476 248
P-CHAN-00477 248
P-CHAN-00478 190
P-CHAN-00479 49
P-CHAN-00480 48, 319
P-CHAN-00481 246
P-CHAN-00482 99
P-CHAN-00486 257
P-CHAN-00487 286
P-CHAN-00488 49
P-CHAN-00489 285
P-CHAN-00490 292
P-CHAN-00492 293
P-CHAN-00493 293
P-CHAN-00494 293
P-CHAN-00495 293
P-CHAN-00496 294
P-CHAN-00498 213
P-CHAN-00500 41
P-CHAN-00502 42
P-CHAN-00503 43
P-CHAN-00504 44
P-CHAN-00505 44
P-CHAN-00506 45
P-CHAN-00507 42
P-CHAN-00508 42
P-CHAN-00509 46
P-CHAN-00510 46
P-CHAN-00511 46
P-CHAN-00512 46
P-CHAN-00514 49
P-CHAN-00515 50
P-CHAN-00516 50
P-CHAN-00517 50
P-CHAN-00518 51
P-CHAN-00519 51
P-CHAN-00520 51
P-CHAN-00550 52
P-CHAN-00551 52
P-CHAN-00552 52
P-CHAN-00553 54
P-CHAN-00554 54
P-CHAN-00600 54
P-CHAN-00601 56
P-CHAN-00602 56
P-CHAN-00603 57
P-CHAN-00604 58
P-CHAN-00605 60
P-CHAN-00606 61
P-CHAN-00650 61
P-CHAN-00651 62
P-CHAN-00652 62
P-CHAN-00653 63
P-CHAN-00654 63
P-CHAN-00655 64
P-CHAN-00656 64
P-CHAN-00657 64
P-CHAN-00658 65
P-CHAN-00751 321
P-CHAN-00752 294
P-CHAN-00753 295
P-CHAN-00754 295
P-CHAN-00755 295
P-CHAN-00756 295
P-CHAN-00757 296
P-CHAN-00758 320
P-CHAN-00759 213
P-CHAN-00760 90
P-CHAN-00761 321
P-CHAN-00762 321
P-CHAN-P-CHAN-00491 292

More Information:

www.beckhoff.com/TF5200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com