DOCUMENTATION ISG-kernel

Manual
Startup data

Short Description:
STUP

www.isg-stuttgart.de
support@isg-stuttgart.de

Documentation version: 1.181
08/11/2024

Preface

Legal information

This documentation was produced with utmost care. The products and scope of functions described are under continuous development. We reserve the right to revise and amend the documentation at any time and without prior notice.

No claims may be made for products which have already been delivered if such claims are based on the specifications, figures and descriptions contained in this documentation.

Personnel qualifications

This description is solely intended for skilled technicians who were trained in control, automation and drive systems and who are familiar with the applicable standards, the relevant documentation and the machining application.

It is absolutely vital to refer to this documentation, the instructions below and the explanations to carry out installation and commissioning work. Skilled technicians are under the obligation to use the documentation duly published for every installation and commissioning operation.

Skilled technicians must ensure that the application or use of the products described fulfil all safety requirements including all applicable laws, regulations, provisions and standards.

Further information

Links below (DE)

https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

or (EN)

https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

contains further information on messages generated in the NC kernel, online help, PLC libraries, tools, etc. in addition to the current documentation.

Disclaimer

It is forbidden to make any changes to the software configuration which are not contained in the options described in this documentation.

Trade marks and patents

The name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® and the associated logos are registered and licensed trade marks of ISG Industrielle Steuerungstechnik GmbH.

The use of other trade marks or logos contained in this documentation by third parties may result in a violation of the rights of the respective trade mark owners.

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

➢ 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.

Attention

Restriction or error

This icon describes restrictions or warns of errors.

Notice

Tips and other notes

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

Example

General example

Example that clarifies the text.

Programing Example

NC programming example

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

Release Note

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

List of figures

1 Overview of start-up parameters

The overview of start-up parameters is sorted into a 4-column table.

- Column 1 contains the unambiguous identifier of the start-up parameter called the “ID” which consists of the prefix “P-STUP” and a unique 5-digit number,
  e.g. P-STUP-00018.
- Column 2 represents the data structure which defines the parameter,
  e.g. pfad[i].
  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 name,
  e.g. prg[j]
  The important thing is that “structure”+”parameter” always belong together and must be configured in exactly the same way in the start-up parameter list,
  e.g. pfad[i]. prg[j]
- Column 4 contains the “functionality” in a summarised term/short description,
  e.g. Program path.

ID	Structure	Parameter	Functionality/ short description
P-STUP-00001 [▶ 20]		kanal_anzahl	Number of configured channels
P-STUP-00002 [▶ 20]		sercos_hochlauf	SERCOS start-up
P-STUP-00003 [▶ 21]		sercos_ring_anzahl	Number of SERCOS rings
P-STUP-00005 [▶ 21]		mds_time_slots	SERCOS time slot calculation
P-STUP-00006 [▶ 21]		at_tslot_type	SERCOS-AT time slot calculation
P-STUP-00007 [▶ 22]		konfiguration	Topology selected
P-STUP-00008 [▶ 22]		listen	List type
P-STUP-00009 [▶ 23]		sda_mds[i]	List file name for default channel parameters
P-STUP-00010 [▶ 23]		wrkz_data[i]	Name of the tool data list file
P-STUP-00011 [▶ 24]		nullp_data[i]	Name of the list file for zero point data
P-STUP-00012 [▶ 24]		pzv_data[i]	Name of the list files for clamp position offset data
P-STUP-00013 [▶ 24]		hand_mds	Name of the list file for manual mode parameters
P-STUP-00014 [▶ 25]		zahl_mds	Number of axis machine data records
P-STUP-00015 [▶ 25]		achs_mds[i]	Name of the list file for axis parameters
P-STUP-00016 [▶ 26]		zahl_kw	Number of offset value lists

ID	Structure	Parameter	Functionality/ short description
P-STUP-00017 [▶ 26]		achs_kw[i]	Name of the list files for offset values
P-STUP-00018 [▶ 29]	pfad[i].	prg[j]	Path specification
P-STUP-00019 [▶ 29]	pfad[i].	log_nr[j]	Logical path number
P-STUP-00020 [▶ 30]	pfad[i].	typ[j]	Path type
P-STUP-00021 [▶ 30]	pfad[i].	prioritaet[j]	Priority
P-STUP-00022 [▶ 69]		online_prod_time_opt	Time-optimised setting for on-line machining time calcula-tion simulation
P-STUP-00024 [▶ 33]	hmi[i].	objects	Name of the list file
P-STUP-00025 [▶ 33]	hmi[i].	mode	Mode of the list file
P-STUP-00026 [▶ 34]	channel[i].	objects	Name of the list file
P-STUP-00027 [▶ 34]	channel[i].	mode	Mode of the list file
P-STUP-00029 [▶ 68]		common_t_d_request	Common transmission of T and D numbers
P-STUP-00031 [▶ 22]		optical_intensity	SERCOS master transmit power
P-STUP-00033 [▶ 35]		fb_storage_size[i]	Memory size for backward motion
P-STUP-00034 [▶ 23]		default_sda_mds	Name of the list file for default channel parameters
P-STUP-00035 [▶ 25]		default_achs_mds	Name of the list file for default axis parameters
P-STUP-00036 [▶ 27]		achs_kw_log_ax_nr[i]	Axis assignment of the offset value list
P-STUP-00037 [▶ 35]		ext_var_max	Memory size for external vari-ables
P-STUP-00039 [▶ 36]		contour_visu_ifc_version	Version identifier of visualisa-tion data
P-STUP-00040 [▶ 36]		single_protocol_fifo	Global or channel-specific output of display data
P-STUP-00042 [▶ 41]	configuration.posi-tion_controller.	log_entry_number	Maximum number of entries in the history buffer
P-STUP-00043 [▶ 42]	configuration.posi-tion_controller.	log_level	Defining the type of logged events
P-STUP-00091 [▶ 42]	configuration.axes_man-ager.	log_entry_number	Maximum number of entries in the history buffer
P-STUP-00092 [▶ 43]	configuration.axes_man-ager.	log_level	Defining the type of logged events

ID	Structure	Parameter	Functionality/ short description
P-STUP-00100 [▶ 47]	vol_comp[i].	max_records	Number of records to be read in
P-STUP-00101 [▶ 47]	vol_comp[i].	file_name	Configuration file for Volumetric Compensation
P-STUP-00110 [▶ 37]	configuration.position_controller.	enable_external_compensation_ifc	Enabling the external compensation
P-STUP-00111 [▶ 31]	configuration.diagnosis_upload.	path	File path for diagnosis upload
P-STUP-00112 [▶ 31]	configuration.diagnosis_upload.	default_file	Filename for diagnosis upload
P-STUP-00113 [▶ 31]	configuration.diagnosis_upload.	startup_file	Filename for upload file – start-up
P-STUP-00114 [▶ 32]	configuration.diagnosis_upload.	history_nbr	Number of diagnosis data output files to be stored
P-STUP-00115 [▶ 32]	configuration.diagnosis_upload.	topics	Identifier to specify the diagnosis upload
P-STUP-00117 [▶ 32]	configuration.diagnosis_upload.	mode	Diagnosis upload mode
P-STUP-00118 [▶ 71]	signal_wait.	use_signal_acknowledge	Activate signal acknowledgement
P-STUP-00119 [▶ 71]	signal_wait.	nbr_events	Maximum number of SIGNAL/WAIT events
P-STUP-00120 [▶ 48]	customer.	val[i]	Free values
P-STUP-00130 [▶ 38]		cam_table_loader	Name of the list file for cam tables
P-STUP-00131 [▶ 38]		cam_table_storage_size	Size of cam table memory
P-STUP-00132 [▶ 39]		trace_function	Enabling/disabling the trace function
P-STUP-00133 [▶ 39]		trace_buffer_size	Defining the ring buffer size
P-STUP-00134 [▶ 40]		scheduling_position_controller	Parameterising scheduling
P-STUP-00135 [▶ 28]		rtconf_lis	Name of the list file for CNC real-time settings
P-STUP-00136 [▶ 28]		hw_configuration_list	Name of the list file for the device configuration
P-STUP-00137 [▶ 63]		scene_mds	Name of the list file for scene
P-STUP-00138 [▶ 63]		enable_scene	Enable the scene function
P-STUP-00145 [▶ 37]		ext_var_struct_member_alignment	Alignment of external variables
P-STUP-00146 [▶ 35]		ve_var[i]	Name of the list file for external variables

ID	Structure	Parameter	Functionality/ short description
P-STUP-00158 [▶ 28]		error_message_texts	Name of error message text file
P-STUP-00166 [▶ 52]		no_error_message_at_reset	Logging a CNC reset as event in error message output
P-STUP-00167 [▶ 49]		error_protocol_mode	Logging mode of error output
P-STUP-00168 [▶ 50]		error_text_of_id	Name of the file for error message texts
P-STUP-00169 [▶ 50]		error_text_user_of_id	Name of the file for user-specific error message texts
P-STUP-00170 [▶ 51]		error_log_file_name	Name of the error log file
P-STUP-00171 [▶ 51]		error_log_file_max_size	Maximum size of the error log file in bytes
P-STUP-00172 [▶ 51]		error_plc_wait_cycles	Waiting cycles before evaluation of PLC activation
P-STUP-00173 [▶ 52]		error_ao_name	Additional descriptive text (AO name)
P-STUP-00175 [▶ 48]		ads_32_bit_comp_mode	32-bit compatibility mode for CNC display data
P-STUP-00182 [▶ 41]		schedule_config	Scheduling the CNC
P-STUP-00183 [▶ 77]	configuration.channel[**0**].decoder.	vi_memory	Maximum V.I. user memory in bytes
P-STUP-00184 [▶ 77]	configuration.channel[**0**].decoder.	vi_maximal_var_count	Maximum number of creatable V.I. variables
P-STUP-00186 [▶ 54]	error_filter[i].	reason	Cause of error
P-STUP-00187 [▶ 55]	error_filter[i].	action	Error action
P-STUP-00188 [▶ 55]	error_filter[i].	conditional_activation	Conditional activation
P-STUP-00189 [▶ 56]	error_filter[i].	conditional_action	Conditional action
P-STUP-00190 [▶ 57]	error_filter[i].	conditional_param	Conditional filter activation
P-STUP-00191 [▶ 57]	error_filter[i].	conditional_output	Output of additional error information
P-STUP-00192 [▶ 64]	plcopen_unit.linear.	position	Setting the units of linear axis positions for PLCopen
P-STUP-00193 [▶ 64]	plcopen_unit.linear.	velocity	Setting the linear axis velocity unit for PLCopen
P-STUP-00194 [▶ 65]	plcopen_unit.linear.	acceleration	Setting the linear axis velocity unit for PLCopen
P-STUP-00195 [▶ 65]	plcopen_unit.linear.	jerk	Setting the linear axis jerk unit for PLCopen
P-STUP-00196 [▶ 66]	plcopen_unit.rotatory.	position	Setting the units of rotary axis positions for PLCopen

ID	Structure	Parameter	Functionality/ short description
P-STUP-00197 [▶ 66]	plcopen_unit.rotatory.	velocity	Setting the units of rotary axis speeds for PLCopen
P-STUP-00198 [▶ 67]	plcopen_unit.rotatory.	acceleration	Setting the units of rotary axis speed for PLCopen
P-STUP-00199 [▶ 67]	plcopen_unit.rotatory.	jerk	Setting the units of rotary axis jerk for PLCopen
P-STUP-00200 [▶ 53]		error_text_cycles_of_id	Name of the file for error message texts of CNC cycles
P-STUP-00213 [▶ 69]	task_trace.geo.	max_records	Number of logging entries for logging
P-STUP-00214 [▶ 70]	task_trace.geo.	filename	Name of the output file
P-STUP-00215 [▶ 70]	task_trace.geo.	history_filename	Name of the history file

1.1 Migrated start-up parameters

Release Note

The following start-up parameters are available as channel parameters as of Versions V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10. The reason for this is improved configurability of the NC channel

The previous start-up parameters can still be used for compatibility reasons.

Meaning of the parameter	Previous start-up parameter	New channel parameter
decoder.function	P-STUP-00050 [▶ 74]	P-CHAN-00500
decoder.log_entry_number	P-STUP-00054 [▶ 76]	P-CHAN-00501
decoder.log_level	P-STUP-00055 [▶ 76]	P-CHAN-00502
decoder.max_chache_number	P-STUP-00051 [▶ 75]	P-CHAN-00503
decoder.max_chache_size	P-STUP-00052 [▶ 75]	P-CHAN-00504
decoder.max_local_sub-routine_definitions	P-STUP-00053 [▶ 76]	P-CHAN-00505
decoder.max_vol_comp_measure-ment_records	P-STUP-00185 [▶ 78]	P-CHAN-00506
tool_radius_comp.function	P-STUP-00080 [▶ 79]	P-CHAN-00550
tool_radius_comp.log_entry_num-ber	P-STUP-00081 [▶ 79]	P-CHAN-00551
tool_radius_comp.log_level	P-STUP-00082 [▶ 80]	P-CHAN-00552
path_preparation.function	P-STUP-00060 [▶ 81]	P-CHAN-00600
path_preparation.log_entry_number	P-STUP-00063 [▶ 87]	P-CHAN-00601

path_preparation.log_level	P-STUP-00064 [▶ 87]	P-CHAN-00602
path_preparation.m_pre_output_lookahead	P-STUP-00061 [▶ 83]	P-CHAN-00603
path_preparation.m_pre_output_max_distance	P-STUP-00062 [▶ 85]	P-CHAN-00604
interpolator.function	P-STUP-00070 [▶ 88]	P-CHAN-00650
interpolator.log_entry_number	P-STUP-00072 [▶ 90]	P-CHAN-00651
interpolator.log_level	P-STUP-00073 [▶ 91]	P-CHAN-00652
interpolator.parameter, param, number_blocks_lah	P-STUP-00071 [▶ 90]	P-CHAN-00653
interpolator.blocks_per_call	P-STUP-00075 [▶ 92]	P-CHAN-00654
interpolator.dyn_cs_history_max	P-STUP-00074 [▶ 91]	P-CHAN-00657
interpolator.contour_lookahead_log_max	P-STUP-00076 [▶ 92]	P-CHAN-00658

2 General description

2.1 Links to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

For technical reasons these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

2.2 Structure and classification of start-up parameters

The start-up list parameters contain application-specific data and links to files that are required to start up the NC kernel. The number of axes, the number of channels and path specifications also used to search for specific ASCII files can be specified in these ASCII files.

Value ranges of parameters may also be defined by stating a limit resulting from data width, e.g. MAX(UNS16), etc.

2.3 Syntax and interpretation of ASCII list file

An interpreter copies the entries in the ASCII list file into identical internal structures which are then checked for plausibility. To ensure reliable controller start-up every time, any defective entries found by the plausibility check are replaced by default values.

Unknown entries are not taken over. These irregularities are displayed by warning messages. We advise you to investigate the cause for these warning messages and remove defective entries from the ASCII list file.

Notice

The following agreement applies to BOOLEAN data:

Value	Meaning
0	Definition of FALSE
1	Definition of TRUE

Notice

The following agreement applies to STRING data:

If a character string containing characters with a special meaning in ASCII lists (e.g. comment characters, spaces [▶ 19]) is assigned to a STRING type list parameter, this character string must be defined in inverted commas "..." (available as of V3.1.3081.0, V3.1.3108.0).

```
example[0].name "STRING_WITH_COMMENT( # /*)_CHARACTERS"
```

Trailing spaces are discarded on import. The entry..

```
example[0].name "STRING_WITH_POST_SPACES "
```

..has the same meaning as

```
example[0].name "STRING_WITH_POST_SPACES"
```

If the character string only contains characters without any special meaning, no inverted commas are required.

```
example[0].name STRING_WITH_STANDARD_CHARACTERS!
```

2.4 Comments in the ASCII list file

Comments can be in an entire line or can be added at the end of a line.

With a comment spanning an entire line, the comment character "'#" must be placed at the start of the line and followed by a blank.

If a comment is to be inserted at the end of a line, only a blank is required before the comment. Blank lines are also possible.

Example

Comments in the ASCII list file

```
#
*****************************************************************************
# Data
#
*****************************************************************************
#
# List comments after numerical values

dummy[1] 1 Comment
dummy[2] 1 # Comment
dummy[3] 1 ( Comment
dummy[4] 1 /* Comment
...
...
```

However, if a character string was assigned to the list parameter as a value in the line, any following comment must be opened by a bracket '('. The comment characters space, # and /* are not permitted.

If a '(' itself is part of the character string, the character string must be defined in inverted commas "..." (available as of V3.1.3081.0, V3.1.3108.0).

```
# List comments after strings

beispiel[0].bezeichnung STRING_1 (comment requires a '(' bracket!)

beispiel[1].bezeichnung" STRING_(2)" (comment requires a '(' bracket!)
```

3 Description of elements

Notice

The TwinCAT System Manager enters or changes a variety of parameters automatically. A manual change is overwritten when the configuration is activated.

Parameters that are automatically changed are marked accordingly.

3.1 Number of configured channels (P-STUP-00001)

P-STUP-00001	Number of configured channels
Description	Application-specific definition of the number of NC channels. The number specified in this parameter must correspond to the configured channels. This parameter informs the systems sequence controller of the number of NC channels. This topology description compiled in binary lists must correspond to this data item.
Parameter	kanal_anzahl
Data type	SGN16
Data range	1 - 12
Dimension	----
Default value	0
Remarks	This parameter is used automatically in TwinCAT systems.

3.2 SERCOS start-up (P-STUP-00002)

P-STUP-00002	SERCOS start-up
Description	This parameter defines whether SERCOS drives should also be run up at start-up.
Parameter	sercos_hochlauf
Data type	SGN16
Data range	0 or .1
Dimension	----
Default value	0
Remarks	A SERCOS card must be installed. TwinCAT: SERCOS parameter without effect.

3.3 Number of SERCOS rings (P-STUP-00003)

P-STUP-00003	Number of SERCOS rings
Description	This parameter defines the number of SERCOS rings in the system.
Parameter	sercos_ring_anzahl
Data type	UNS16
Data range	0 or .1
Dimension	----
Default value	0
Remarks	

3.4 SERCOS time slot calculation (P-STUP-00005)

P-STUP-00005	SERCOS time slot calculation
Description	Option to select a SERCOS time slot calculation mode.
Parameter	mds_time_slots
Data type	SGN16
Data range	0: internal calculation 1: Adopting predefined values from axis / device lists
Dimension	----
Default value	0
Remarks	

3.5 SERCOS AT time slot calculation (P-STUP-00006)

P-STUP-00006	SERCOS AT time slot calculation
Description	Option to select an AT time slot calculation mode.
Parameter	at_tslot_type
Data type	STRING
Data range	DEFAULT / OPTION1
Dimension	----
Default value	DEFAULT:
Remarks	

3.6 SERCOS Master transmit power (P-STUP-00031)

P-STUP-00031	SERCOS Master transmit power
Description	This parameter adjusts the light intensity of the SERCOS master hardware transmitter diode. A reduction in transmit power can, for example, prevent a receiver diode overload in the downstream SERCOS ring user.
Parameter	optical_intensity
Data type	UNS16
Data range	1 ... 6
Dimension	----
Default value	6
Remarks	

3.7 Topology selected (P-STUP-00007)

P-STUP-00007	Topology selected
Description	This parameter is used to select a topology description (configuration) for the NC kernel. The selected configuration must be contained in the code in the form of a binary file.
Parameter	konfiguration
Data type	STRING
Data range	EIN_KANAL_KONFIGURIERUNG / ... / ACHT_KANAL_KONFIGURIERUNG
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.8 List type (P-STUP-00008)

P-STUP-00008	List type
Description	This parameter is used to define whether start-up is performed using binary lists or ASCII lists.
Parameter	listen
Data type	STRING
Data range	ASCII / BINAER
Dimension	----
Default value	ASCII
Remarks	TwinCAT: Entry may not be changed.

3.9 List file name for channel parameters (P-STUP-00009)

P-STUP-00009	List file name for default channel parameters
Description	This parameter is used cross-channel to define the name of the file containing channel parameters.
Parameter	sda_mds[i] where i = 0 ... 11 (maximum number of channels: 12, application-specific)
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.9.1 List file name for default channel parameters (P-STUP-00034)

P-STUP-00034	List file name for default channel parameters
Description	This parameter is used cross-channel to define the name of the file containing channel parameters assigned with default values.
Parameter	default_sda_mds
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.10 Name of the tool data list file (P-STUP-00010)

P-STUP-00010	Name of the tool data list file
Description	This parameter defines for each channel the name of the file containing tool data.
Parameter	werkz_data[i] where i = 0 ... 11 (maximum number of channels: 12, application-specific)
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.11 Name of the list file for zero point data (P-STUP-00011)

P-STUP-00011	Name of the list file for zero point data
Description	This parameter defines for each channel the name of the file containing zero point data.
Parameter	nullp_data[i] where i = 0 ... 11 (maximum number of channels: 12, application-specific)
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.12 Name of the list files for clamp position offset data (P-STUP-00012)

P-STUP-00012	Name of the list files for clamp position offset data
Description	This parameter defines for each channel the name of the file containing clamp position offset data.
Parameter	pzv_data[i] where i = 0 ... 11 (maximum number of channels: 12, application-specific)
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.13 Name of the list file for axis parameters (P-STUP-00013)

P-STUP-00013	Name of the list file for manual mode parameters
Description	This parameter is used cross-channel to define the name of the file containing manual mode parameters.
Parameter	hand_mds
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.14 Number of axis machine data records (P-STUP-00014)

P-STUP-00014	Number of axis machine data records
Description	This parameter determines the number of axis data records that are to be interpreted and this defines the number of axes in the system.
Parameter	zahl_mds
Data type	SGN16
Data range	1 ... 32
Dimension	----
Default value	0
Remarks	This parameter is used automatically in TwinCAT systems.

3.15 Name of the list file for axis parameters (P-STUP-00015)

P-STUP-00015	Name of the list file for axis parameters
Description	This parameter is used cross-channel to define the names of axis parameter data files. The number of axis parameter data files must correspond to the number of axis parameter data records. If more files are specified than are contained in P-STUP-00014 [▶ 25] (zahl_mds), the excess files are not considered. Vice versa, if the system attempts to open unknown files, error messages are output and controller start-up is aborted.
Parameter	achs_mds[i] where i = 0 ... 31 (maximum number of axes in the system: 32, application-specific)
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.15.1 Name of the list file for default axis parameters (P-STUP-00035)

P-STUP-00035	Name of the list file for default axis parameters
Description	This parameter is used cross-channel to define the name of the file containing the axis parameters assigned with default values.
Parameter	default_achs_mds
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.16 Number of offset value lists (P-STUP-00016)

P-STUP-00016	Number of offset value lists
Description	This parameter determines the number of offset value lists to be interpreted. The number of offset value lists may not be greater than the number of axes. An offset value list may exist for each axis.
Parameter	zahl_kw
Data type	SGN16
Data range	1 ... 32
Dimension	----
Default value	0
Remarks	This parameter is used automatically in TwinCAT systems.

3.17 Name of the list files for offset values (P-STUP-00017)

P-STUP-00017	Name of the list files for offset values
Description	This parameter is used cross-channel to define the names of offset value files. The number of offset value lists must correspond to the number of list files. If more files are specified than are contained in P-STUP-00016 [▶ 26] (zahl_kw), the excess files are not considered. Vice versa, if the system attempts to open unknown files, error messages are output and controller start-up is aborted.
Parameter	achs_kw[i] where i = 0 ... 31 (maximum number of axes in the system: 32, application-specific)
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.17.1 Axis assignment of the offset value list (P-STUP-00036)

P-STUP-00036	**Axis assignment of the offset value list**
Description	The logical axis number is used cross-channel to define the assignment between axes and offset value lists.
Parameter	achs_kw_log_ax_nr[i] where i = 0 ... 31 (maximum number of axes in the system: 32, application-specific)
Data type	UNS16
Data range	1 ... MAX(UNS16)
Dimension	----
Default value	0
Remarks	This parameter is used automatically in TwinCAT systems.

3.18 Name of the list file for CNC real-time settings (P-STUP-00135)

P-STUP-00135	Name of the list file for CNC real-time settings
Description	This parameter defines the path and filename of the configuration file for CNC real-time settings.
Parameter	rtconf_lis
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	* Note: The default value of variables is a blank string. This parameter is not available under TwinCAT.

3.19 Name of the list file for device configuration (P-STUP-00136)

P-STUP-00136	Name of the list file for device configuration
Description	This parameter defines the path and filename of the device configuration file.
Parameter	hw_configuration_list
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Unit	----
Default value	*
Remarks	* Note: The default value of variables is a blank string. This parameter is not available under TwinCAT.

3.20 Name of error message text file (P-STUP-00158)

P-STUP-00158	Name of error message text file
Description	This parameter defines the path and filename of the error message text file.
Parameter	error_message_texts
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Unit	----
Default value	*
Remarks	* Note: The default value of variables is a blank string. This parameter is not available under TwinCAT.

3.21 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 priority must be specified for each program path.

Notice

As of CNC Build V3.1.3025.05, program paths can also be defined in the channel parameters. In this case, the program paths are removed from the start-up parameters.

Further information on program paths in the channel: [CHAN//NC program paths (path[i].*)]

Structure name	Index
pfad[i]	i = 0 ... 11 (channel index, e.g. Channel 1 -> Index 0, maximum number of channels: 12, application-specific)

3.21.1 Path specification (P-STUP-00018)

P-STUP-00018	Path name
Description	This parameter defines the path to the NC programs. The CNC employs this path to open an NC program.
Parameter	pfad[i].prg[j] where j = 0 ... 11 (maximum number of program paths in the system: 12, application-specific)
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	* Note: The default value of variables is a blank string.

3.21.2 Logical path number (P-STUP-00019)

P-STUP-00019	Logical path number
Description	This parameter defines a logical path number for the program path. Logical path numbers must be unique within the system.
Parameter	pfad[i].log_nr[j] where j = 0 ... 11 (maximum number of program paths in the system: 12, application-specific)
Data type	UNS16
Data range	1 ... MAX(UNS16)
Dimension	----
Default value	0
Remarks	

3.21.3 Path type (P-STUP-00020)

P-STUP-00020	Path type
Description	This parameter defines the type of the program path bit-encoded. A path specification may also be used for several path types.
Parameter	pfad[i].typ[j] where j = 0 ... 11 (maximum number of program paths in the system: 12, application-specific)
Data type	UNS16
Data range	0x01 (main program path) 0x02 (subroutine path) 0x04 (path for #MSG SAVE) 0x08 (path for storing debug data *.dbg) Combinations: 0x03 (main prog.+ subroutine path) 0x05 (main prog. path + path for #MSG SAVE) 0x06 (subroutine path + path for #MSG SAVE) 0x07 (main prog. + subroutine path + path for #MSG SAVE) 0x0B (main prog. path + subroutine path + path for debug data) 0x0F (main prog. path + subroutine path + path for #MSG SAVE and debug data)
Dimension	----
Default value	0
Remarks	

3.21.4 Priority (P-STUP-00021)

P-STUP-00021	Priority
Description	This parameter defines the priority of the program path. Priority determines the sequence of the directories of the corresponding path types when a search is made for 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	pfad[i].prioritaet[j] where j = 0 ... 11 (maximum number of program paths in the system: 12, application-specific)
Data type	UNS16
Data range	0 ... MAX(UNS16)
Dimension	----
Default value	0
Remarks	When the path types 0x04 and 0x08 are set as actual program paths, the priorities must be continued based on the sub program paths.

3.22 Parameter for diagnosis upload

3.22.1 File path for diagnosis upload (P-STUP-00111)

P-STUP-00111	File path for diagnosis upload
Description	This parameter defines the file path for writing the diagnosis data upload file.
Parameter	configuration.diagnosis_upload.path
Data type	STRING
Dimension	---
Default value	Standard CNC programme path, can be set via system manager -> CNC
Remarks	

3.22.2 Filename for diagnosis upload (P-STUP-00112)

P-STUP-00112	Filename for diagnosis upload
Description	This parameter specifies the filename of the diagnosis data upload file. The file path is defined by P-STUP-00111 [▶ 31] .
Parameter	configuration.diagnosis_upload.default_file
Data type	STRING
Dimension	---
Default value	diag_data.txt
Remarks	

3.22.3 Filename for upload file – start-up (P-STUP-00113)

P-STUP-00113	Name of the upload file of the diagnosis data at start-up
Description	This parameter specifies the name of the diagnosis data upload file at start-up. The file path is defined by P-STUP-00111 [▶ 31] .
Parameter	configuration.diagnosis_upload.startup_file
Data type	STRING
Dimension	---
Default value	
Remarks	**Note**: If P-STUP-00113 is unassigned, no diagnosis upload can be commanded at start-up.

3.22.4 Number of diagnosis data output files to be stored (P-STUP-00114)

P-STUP-00114	Number of diagnosis data output files to be saved
Description	This parameter defines the number of diagnosis data output files to be saved. The file path is defined by P-STUP-00111 [▶ 31] .
Parameter	configuration.diagnosis_upload.history_nbr
Data type	UNS16
Dimension	----
Default value	1
Remarks	

3.22.5 Identifier to specify the diagnosis upload (P-STUP-00115)

P-STUP-00115	Identifier to specify the diagnosis upload
Description	This parameter defines the identifiers to specify the diagnosis upload. For an overview of possible identifiers, see TOPICS table.
Parameter	configuration.diagnosis_upload.topics
Data type	STRING
Dimension	----
Default value	MAX
Remarks	

3.22.6 Diagnosis upload mode (P-STUP-00117)

P-STUP-00117	Diagnosis upload mode
Description	This parameter defines the mode for a diagnosis upload. For an overview of the possible settings, see the Mode Table
Parameter	configuration.diagnosis_upload.mode
Data type	STRING
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.

Diagnosis upload mode	Meaning
STANDARD	Default upload with no further functions
REGRESSION	Formatting for regression test
PROTOCOL_INFO	Additional information about the upload procedure
MSG_FLUSH_OFF	Deactivate automatic flush for messages to ISG_DIAG_BED at the start of the diagnosis upload.

3.23 HMI objects (hmi[i].*)

3.23.1 Name of the list file (P-STUP-00024)

P-STUP-00024	Name of the list file
Description	This parameter defines the cross-channel name of the HMI object list.
Parameter	hmi[i].objects
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.23.2 Mode of the list file (P-STUP-00025)

P-STUP-00025	Mode of the list file
Description	This parameter defines the mode for loading the HMI object list.
Parameter	hmi[i].mode
Data type	STRING
Data range	write: The existing list is only read in. write+: The list is first created, then read in. default: The internal default list is used. No lists are generated externally.
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.24 Parameters for the BF Channel (channel[i].*)

3.24.1 Mode of the list file (P-STUP-00027)

P-STUP-00027	Mode of the list file
Description	This parameter defines the mode for loading the BF Channel object list.
Parameter	channel[i].mode
Data type	STRING
Data range	write: The existing list is only read in. write+: The list is first created, then read in. default: The internal default list is used. No lists are generated externally.
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.24.2 Name of the list file (P-STUP-00026)

P-STUP-00026	Name of the list file
Description	This parameter defines the cross-channel name of the BF Channel object list.
Parameter	channel[i].objects
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	* Note: The default value of variables is a blank string.

3.25 Memory size for backward motion (P-STUP-00033)

P-STUP-00033	Memory size for backward motion
Description	This parameter defines the memory size in bytes used for backward motion on the path. During start-up, the NC checks whether the required minimum size is available. If this is not the case, a warning is output and the memory size is set to the required minimum value. If the size is set to 0, the “forward/ backward motion on the path” function is not available. The maximum size is only limited by the resources available on the PC.
Parameter	fb_storage_size[i] where i = 0 to 11 (maximum number of channels: 12, application-specific)
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

3.26 Memory size for external variables (P-STUP-00037)

P-STUP-00037	Memory size for external variables
Description	This parameter dimensions the memory area available for the external variables of each channel on the HLI. A separate memory area of this size is created for global external variables. The number defined here determines the number of 24-byte blocks of which each of these V.E. memory areas consists.
Parameter	ext_var_max
Data type	UNS16
Data range	0 ... MAX(UNS16)
Dimension	----
Default value	0
Remarks	This parameter is used automatically in TwinCAT systems.

3.27 Name of the list file for external variables (P-STUP-00146)

P-STUP-00146	Name of the list file for external variables
Description	This parameter defines for each channel the name of the file containing the external variables.
Parameter	ve_var[i] where i = 0 ... 11 (maximum number of channels: 12, application-specific)
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	This parameter is used automatically in TwinCAT systems. * Note: The default value of variables is a blank string.

3.28 Version identifier of visualisation data (P-STUP-00039)

P-STUP-00039	Version identifier of visualisation data	
Description	The parameter sets the type of data structure which the contour visualisation ([FCT-C17]) supplies. Depending on the setting selected, more or less visualisation data is generated. An overview of existing data structures is contained in [FCT-C17].	
Parameter	contour_visu_ifc_version	
Data type	UNS32	
Data range	contour_visu_ifc_version	Data structure
	0	SOLLKONT_VISU_DATA_V0 (default)
	1	SOLLKONT_VISU_DATA_V1
	2	SOLLKONT_VISU_DATA_V2
	3	SOLLKONT_VISU_DATA_V3
	4	SOLLKONT_VISU_DATA_V4
	5	SOLLKONT_VISU_DATA_V5
	6	SOLLKONT_VISU_DATA_V6
	7	SOLLKONT_VISU_DATA_V7
	8	SOLLKONT_VISU_DATA_V8
	9	SOLLKONT_VISU_DATA_V9
	10	SOLLKONT_VISU_DATA_V10
	11	SOLLKONT_VISU_DATA_V11
Dimension	----	
Default value	0	
Remarks		

3.29 Global or channel-specific output of display data (P-STUP-00040)

P-STUP-00040	Global or channel-specific output of display data
Description	This parameter defines whether visualisation data is written to a FIFO output for each channel or whether the visualisation data of all channels is written to a global FIFO output.
Parameter	single_protocol_fifo
Data type	BOOLEAN
Data range	0: Channel-specific output of visualisation data 1: Common output of visualisation data.
Dimension	----
Default value	0 *
Remarks	* 1 as of CNC Build V3.1.3038

3.30 Alignment of external variables (P-STUP-00145)

P-STUP-00145	Alignment of external variables
Description	This parameter defines the alignment of external variables in the CNC ([EXTV]). IMPORTANT: They must correspond to the alignment setting used in the PLC.
Parameter	ext_var_struct_member_alignment
Data type	UNS08
Data range	Permissible values for this parameter are: 0: The CNC automatically defines the alignment of variables depending on the target platform 1: 1-byte alignment (pragma pack) is used for external variables. No alignment bytes are added. 2: 2-byte alignment is used 4: The CNC uses 4-byte alignment 8: The CNC uses 8-byte alignment
Dimension	----
Default value	0
Remarks	This parameter is only available as of CNC Build V3.1.3019.00 and higher. IMPORTANT: The alignment setting defined here must correspond to the alignment setting used in the PLC. Otherwise, no or incorrect values can be transferred if there is shared access to external variables. This parameter may only be changed in consultation with the controller manufacturer.

The following difference exists for TwinCAT systems:

TwinCAT2 -> 1-byte alignment

TwinCAT3 -> 8-byte alignment

3.31 Enabling the external compensation (P-STUP-00110)

P-STUP-00110	Enabling the external compensation
Description	This parameter enables the function in the NC channel.
Parameter	configuration.position_controller.enable_external_compensation_ifc
Data type	BOOLEAN
Data range	0/1
Dimension	----
Default value	0
Remarks	This parameter can only be used with TwinCAT3. This is because of the use of the McCOM interface that is only available with TwinCAT3. This parameter is available as of CNC Build V3.1.3074

3.32 Parameters for camming

3.32.1 Name of the list file for cam tables (P-STUP-00130)

P-STUP-00130	Name of the list file for cam tables
Description	This parameter defines the name and path of the parameter file specifying the cam table file entries.
Parameter	cam_table_loader
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	*
Remarks	* Note: The default value of variables is a blank string.

3.32.2 Size of cam table memory (P-STUP-00131)

P-STUP-00131	Size of cam table memory
Description	This parameter defines the size of the cam table memory in bytes.
Parameter	cam_table_storage_size
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	50000
Remarks	

3.33 Parameter to trace position and dynamic data

3.33.1 Enabling/disabling the trace function (P-STUP-00132)

P-STUP-00132	Enabling/disabling the trace function
Description	This parameter enables or disables the trace function of the NC kernel.
Parameter	trace_function
Data type	BOOLEAN
Data range	0/1
Dimension	----
Default value	0
Remarks	

3.33.2 Defining the ring buffer size (P-STUP-00133)

P-STUP-00133	Defining the ring buffer size
Description	This parameter defines the size of the ring buffer for the trace function. The size indicates the number of buffer locations.
Parameter	trace_buffer_size
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	20000
Remarks	

3.34 Scheduling in the CNC

3.34.1 Scheduling in the controller (P-STUP-00134)

P-STUP-00134	Scheduling in the controller
Description	This parameter controls CNC scheduling. A distinction is made between 2 methods: **Method 1:** Control runs in the CNC for at least one axis. Scheduling executes the following sequence: - Read actual values - Calculate position lags - Write velocity command values **Method 2:** All axes are position-controlled. Scheduling automatically executes the following changed sequence: - Read actual values - Write position command values - Calculate position for next cycle
Parameter	scheduling_position_controller
Data type	STRING
Data range	DEFAULT: Depending on axis control, the CNC decides on which scheduling mode is selected (mode 1 or 2). OPT_CNC_POS_CONTROL: Control in CNC; scheduling acc. to mode 1. OPT_DRIVE_POS_CONTROL: Control in the drives; scheduling acc. to mode 2
Dimension	----
Default value	DEFAULT
Remarks	

3.34.2 Scheduling the NC channel (P-STUP-00182)

P-STUP-00182	Scheduling the NC channel
Description	This parameter defines channel scheduling for the CNC. For the die-sinking function, this value need only be parameterised in the down channel using DIE_SINKING.
Parameter	schedule_config
Data type	UNS32
Data range	DEFAULT default scheduling DIE_SINKING optimised scheduling for die sinking.
Dimension	-
Default value	DEFAULT
Remarks	Available as of CNC Build V3.1.3105.01

3.35 Parameter for configuration (configuration.*)

3.35.1 Platform scaling

3.35.1.1 Position control (configuration.position_controller.*)

3.35.1.1.1 Maximum number of logged events (P-STUP-00042)

P-STUP-00042	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.position_controller.log_entry_number
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

3.35.1.1.2 Defining the type of logged events (P-STUP-00043)

P-STUP-00043	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.position_controller.log_level
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

3.35.1.2 Axis management (configuration.axes_manager.*)

3.35.1.2.1 Maximum number of logged events (P-STUP-00091)

P-STUP-00091	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.axes_manager.log_entry_number
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

3.35.1.2.2 Defining the type of logged events (P-STUP-00092)

P-STUP-00092	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.axes_manager.log_level
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

3.35.2 General scaling

3.35.2.1 Logging entries of the CNC

The CNC offers the possibility of storing events in a history memory. If requested, these entries can be read out. When diagnostic data is requested, the entries are stored in a file, for example. Recording of events is currently possible in the following CNC architecture models:

- Decoder
- Path preparation
- Tool radius compensation
- Interpolation
- Axis driver (position control)
- Axis management

Notice

The controller manufacturer must already have planned for recording of individual CNC events.

The **log_entry_number** parameter defines the maximum number of recorded events. If more entries occur, the oldest entry is overwritten.

The **log_level** parameter enables users to specifically define the CNC's logging entries to be recorded. 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.

Example parameters

```
configuration.channel[0].decoder.log_level 0x1010102f
configuration.channel[0].decoder.log_entry_number 256

configuration.channel[0].tool_radius_comp.log_entry_number 128
configuration.channel[0].path_preparation.log_entry_number 64

configuration.channel[0].interpolator.log_entry_number 150

configuration.position_controller.log_entry_number 32
configuration.position_controller.log_level 0x10ff20ff

configuration.axes_manager.log_entry_number 20
configuration.axes_manager.log_level 0x000000ff
```

Output example

```
PATH LOGGING CHANNEL NO.: 1
========================================================================
BF 8 logging : 13/150, level ffffffff, index 13

time level  message
------------------------------------------------------------------------
200852 00020000 1) UPLOAD-ind: start size=748
200856 00020000 1) UPLOAD-ind: data received, pos 0 + 748
200856 00020000 1) UPLOAD-ind: finished : position 748, cb lize
272901 00000001 BAHN restart... start
272904 00000001 BAHN restart...finished
279541 00000001 BAHN abort...start
279551 00000001 BAHN abort...finished
…
280622 00000001 BAHN restart...finished

BAVO LOGGING CHANNEL NO.: 1
========================================================================
BF 11 logging : 10/64, level ffffffff, index 10

time level  message
------------------------------------------------------------------------
200851 00020000 1) UPLOAD-req: start size=748, cb=ize, name=
200855 00020000 1) UPLOAD-req: start ackn : size=748
200855 00020000 1) UPLOAD-req: sent data : 0+748
200859 00020000 1) UPLOAD-req: sent data ackn, fini : 0+748=748
272899 00000001 BAVO reset start
…
280641 00000001 BAVO reset finished (no axes returned to AXV)

DECODER LOGGING CHANNEL NO.: 1
========================================================================
BF 10 logging : 0/0, level 1010102f, index 0

time level  message
------------------------------------------------------------------------

LR LOGGING CHANNEL NO.: 1
========================================================================
BF 5 logging : 22/32, level 10ff20ff, index 22

time level  message
------------------------------------------------------------------------
272907 00000001 lr_abort_axis() ok: axis=6
272907 00000001 lr_abort_axis() ok: axis=11
…
279600 00000001 lr_abort_axis() ok: axis=4
------------------------------------------------------------------------
279600 00000001 lr_abort_axis() ok: axis=5
… 280620 00000001 lr_abort_axis() ok: axis=5
```

3.35.2.2 CNC logging events

3.35.2.2.1 Defining logging levels

Bit 31 to Bit 16 for cross-BF log level classes	Description
0x00010000 BF_LOG_STD	Default BF events
0x00020000 BF_LOG_UPLOAD	#COLL/SCENE RESTORE
0x40000000 BF_LOG_HIGH	High priority events
0x80000000 BF_LOG_RESET	Events at BF reset
0xFFFFffff BF_LOG_ALL	All BF log entries are logged.

Bit 15 to Bit 0 for BF-specific log level classes	Description
0x00000001 BAHN_LOG_STD	Default Interpolator
0x00000002 BAHN_LOG_DDTG_	Events at “Delete distance to go”
0x00000004 BAHN_LOG_FBC_	Forward/backward motion
0x00000008 BAHN_LOG_BS_	Block search

0x00000001 BAVO_LOG_STD	Default Bavo

0x00000001 DEC_LOG_STD	Default decoder
0x00000002 DEC_LOG_EXAMPLE_	---
0x00000004 DEC_LOG_VI	Interchannel variables

0x00000001 AXV_LOG_STD	Default axis exchange

0x00000001 LR_LOG_STD	Default position controller
0x00000002 LR_LOG_ALNK	Axis link from IPO to LR during channel output
0x00000004 LR_LOG_BODE_PLOT	Bode plot logging

3.36 Parameters for Volumetric Compensation (vol_comp[i].*)

For every controller, up to five records of compensation settings can be configured. Configuration of more than only one compensation makes sense on multi-channel machines, for example.

Structure name	Index
vol_comp[i]	i = 0 ... 5

3.36.1 Number of records to be read in (P-STUP-00100)

P-STUP-00100	Number of records to be read in
Description	This parameter specifies an upper limit for the number of parameter data records to be read in. It serves to allocate memory during controller start-up. An error is issued if this number is exceeded while reading in the records.
Parameter	vol_comp[i].max_records
Data type	SGN32
Data range	0 … MAX(SGN32)
Dimension	----
Default value	0
Remarks	

3.36.2 Configuration file for Volumetric Compensation (P-STUP-00101)

P-STUP-00101	Configuration file for Volumetric Compensation
Description	The path specified in this parameter refers to a list file which contains the main ith configuration of the volumetric compensation 'i'.
Parameter	vol_comp[i].file_name
Data type	STRING
Data range	<Path to the configuration file>
Dimension	----
Default value	*
Remarks	Parameterisation example: The row vol_comp[0].file_name C:\volcomp\vol_comp_0.lis defines the path to the configuration file. * Note: The default value of variables is a blank string.

3.37 User-specific data (customer.*)

3.37.1 Free values (P-STUP-00120)

P-STUP-00120	Free values
Description	The user can enter any values in this array. The values are not displayed in the controller, only on the HLI in the element gpPform^.nc_config.**customer_val_r[ ]** (see [HLI]). This permits the user to transfer configuration data to the PLC or HMI.
Parameter	customer.val[i] where i = 0 (maximum number of free values 1, application-specific)
Data type	UNS32
Data range	0 … MAX(UNS32)
Dimension	----
Default value	0
Remarks	

3.38 32-bit compatibility mode for CNC display data (P-STUP-00175)

P-STUP-00175	32-bit compatibility mode for CNC display data
Description	As of CNC Build 2807 and CNC Build 3039.06 and higher, the CNC position controller uses 64-bit integer variables with a finer resolution for command and actual positions. To ensure downward compatibility, this data is downscaled for display data in CNC objects and continue to be supplied as 32-bit values. Conversion can be deactivated by setting the ads_32_bit_comp_mode parameter to the value 0. High-resolution position controller variables are then transferred via CNC objects as 64-bit integer values.
Parameter	ads_32_bit_comp_mode
Data type	BOOLEAN
Data range	0: No conversion; high-resolution 64-bit variable. 1: Downward compatibility, conversion and supply of 32-bit integer variables
Dimension	----
Default value	1
Remarks	This parameter is available as of CNC Builds 2.11.2027.01 and V3.1.3039.06 or higher.

3.39 Parameters for error message output

3.39.1 Logging mode (P-STUP-00167)

P-STUP-00167	**Logging mode of error output**	
Description	This parameter controls the output and scope of the error output.	
Parameter	error_protocol_mode	
Data type	STRING	
Data range	**Flag**	**Meaning**
	FILTER_OFF	No filters are evaluated
	VERBOSE	Extended internal diagnostics
	WITHOUT_ERROR_MANAGER	Direct output without error management
	PRINT	Execute print output
	LOG	Log output to log file
	REPORT	Log output to log file
	SEND_TO_PLC_NONE	Suppress output to the PLC
	PRINT_EXTENDED	Extended print output
	LOG_EXTENDED	Extended log output
	REPORT_EXTENDED	Extended application-specific output
	PRINT_NO_WARNINGS	Warnings are suppressed in the print output
	LOG_NO_WARNINGS	Warnings are suppressed in the log output
	REPORT_NO_WARNINGS	Warnings are suppressed in the report output
	SEND_TO_PLC_NO_WARNINGS	Suppress warnings to PLC
	STARTUP_NO_WARNINGS	Suppress warnings during controller start-up
	NO_WARNINGS	Suppress all warnings
	TC3_EVENT_LOGGER	Output to TC3 Event Logger
	TC3_EVENT_LOGGER_CONFIRMED	Output to TC3 Event Logger, automatic confirmation (state confirmed) on deletion of error message
	TC3_EVENT_LOGGER_NO_CONFIRMATION	Output to TC3 Event Logger without confirmation request
Dimension	----	
Default value	LOG | PRINT | REPORT	
Remarks	**Note**: For example, to suppress warnings in the print output, the entire mode must be set accordingly. `error_protocol_mode LOG | PRINT | REPORT | PRINT_NO_WARNINGS`	

Notice

This parameter is provided with reduced scope in the Version V2.2810.xx. The following setting can be made:
error_protocol_mode NO_WARNINGS

Therefore, only warnings can be suppressed.

3.39.2 Name of text file containing error message texts (P-STUP-00168)

P-STUP-00168	Name of the file for error message texts
Description	Name of the file containing the error message texts which belong to the ID (error number). These can be used for output to the log file. This file is used to assign an error number to the related error message text. The file contains one line in the following format for each error ID: <Error-ID> TABULATOR <Error-Text> The default file 'err_text_version_eng.txt' is assumed if no file is specified.
Parameter	error_text_of_id
Data type	STRING
Data range	Maximum 256 characters
Dimension	----
Default value	err_text_version_eng.txt
Remarks	

3.39.3 Name of text file containing user-specific error message texts (P-STUP-00169)

P-STUP-00169	Name of the file for user-specific error message texts
Description	Comparable to default error texts (see P-STUP-00168 [▶ 50]), you can also specify user-specific texts in this file. These texts are for error IDs in the range [1;1000] which the user can define himself with the NC command #ERROR and they are used for errors in the McCOM interfaces. This file is used to assign an error number to the related user-specific error message text. The file contains one line in the following format for each error ID: *<Error-ID>* TABULATOR *<user_specific_error-text>* The filename is configured with relative or absolute path name. For further information see (FCT-M7// Outputting user error messages)
Parameter	error_text_user_of_id
Data type	STRING
Data range	Maximum 256 characters
Dimension	----
Default value	*
Remarks	* Note: The default value of variables is a blank string. The returned error IDs of the McCOM methods are resolved for error values 292030- 292033 (ERR_KIN_TRAFO_CONFIG/ -INITIALIZE/ -FORWARD/ -BACKWARD).

3.39.4 Name of error log file (P-STUP-00170)

P-STUP-00170	Name of the error log file
Description	Name of the error log file (including directory and path information). If no complete name is specified, no log file is generated and the error message ID 296000 is output. If the parameter is not configured, the error log file is generated with the default file name.
Parameter	error_log_file_name
Data type	STRING
Data range	Maximum of 256 characters
Dimension	----
Default value	<TwinCATInstallation>\Components\Mc\CNC\Diagnostics\error.log
Remarks	The default filename and the associated path are application-dependent.

If no path is specified in TwinCAT systems, the configured NC program path is used.

3.39.5 Maximum size of the error log file (P-STUP-00171)

P-STUP-00171	Maximum size of the error log file in bytes
Description	This parameter defines the maximum size of the error log file.
Parameter	error_log_file_max_size
Data type	SGN32
Data range	> 0 :maximum size of the error log file. If this size is exceeded, the original file is copied to a backup file (extension: <name>.bak) and the contents of the original file are deleted. == 0 : no backup file is created.
Dimension	----
Default value	100000
Remarks	

3.39.6 Waiting cycles before evaluation of PLC activation (P-STUP-00172)

P-STUP-00172	Waiting cycles before evaluation of PLC activation
Description	Waiting cycles in CNC ticks after an error has occurred before the PLC's activation mask for the filter rules is evaluated.
Parameter	error_plc_wait_cycles
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	-
Remarks	

3.39.7 Additional descriptive text (AO name) (P-STUP-00173)

P-STUP-00173	Additional descriptive text (AO name)
Description	Descriptive text (architecture object) that is additionally appended in the event of an error message.
Parameter	error_ao_name
Data type	STRING
Data range	Maximum 83 characters
Dimension	----
Default value	*
Remarks	* Note: The default value of variables is a blank string.

3.39.8 Logging a CNC resets (P-STUP-00166)

P-STUP-00166	Logging a CNC reset as event in error message output
Description	This parameter defines whether the CNC reset triggered by the user is included as an event in the error message log. Previous error messages are acknowledged when the CNC is reset. This always occurs regardless of the setting of P-STUP-00166.
Parameter	no_error_message_at_reset
Data type	BOOLEAN
Data range	0: a CNC reset is logged as warning ID 270076 in the error message output. 1 a CNC reset is not logged
Dimension	----
Default value	0
Remarks	

3.39.9 Name of the file for error message texts of CNC cycles

P-STUP-00200	Name of the file for error message texts of CNC cycles
Description	Name of the file containing the error message texts which belong to the ID (error number). These can be used for output to the log file. This file is used to assign an error number to the related error message text. The file contains one line in the following format for each error ID: <Error-ID> TABULATOR <Error-Text>
Parameter	error_text_cycles_of_id
Data type	STRING
Data range	Maximum 256 characters
Dimension	----
Default value	err_text_cycles_eng.txt
Remarks	

3.40 Filter parameters for error handling on the platform (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// Error management]

Structure name	Index
error_filter[i]	0 ≤ i ≤ 3 (maximum number of error filters: 4)

3.40.1 Error cause (P-STUP-00186)

P-STUP-00186	Cause of error
Description	The individual error codes can be listed as numbers or texts, whereby the entire row must comply with the following syntax: (number | text) {, (number | text) } where: number:= CNC error number text:=" error-specific text " Example: error_filter[0].reason "D012:", 123000, 123001 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.

3.40.2 Error action (P-STUP-00187)

P-STUP-00187	Error action
Description	Action that is to be performed if an error occurs.
Parameter	error_filter[i].action where i = 0 ... 3 (maximum number of filters, application-specific)
Data type	STRING
Data range	ACTION = NONE | DRIVE_STATE_REQ | PRE_RUN_STATE_REQ | RUN_STATE_REQ NONE: No action DRIVE_STATE_REQ: Read out drive status PRE_RUN_STATE_REQ: Error at start-up of the controller bus in PRE-run state RUN_STATE_REQ: Error at start-up of the controller bus in Run state
Dimension	----
Default value	*
Remarks	For SERCOS drive profiles: DRIVE_STATE_REQ: S-0-0095 diagnostic PRE_RUN_STATE_REQ: S-0-0021: list of unknown operation data in CP2 -> CP3, command 127 RUN_STATE_REQ: S-0-0022: list of unknown operation data in CP3 -> CP4, command 128 For ProfiDrive profiles: <all actions> Parameter 945 For CANopen profiles <all actions> Parameter ID603F * Note: The default value of variables is a blank string.

3.40.3 Conditional activation (P-STUP-00188)

P-STUP-00188	Conditional activation
Description	This filter rule is activated when the corresponding bit is set via the user interface or the PLC (HLI::ControlUnit- Activating error filter rules - Platform).
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: *error_filter[0].conditional_activation 0x2* An activation bit = 0 means that the action is always executed.

3.40.4 Conditional action (P-STUP-00189)

P-STUP-00189	Conditional action
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]) FORCE = F_WARNING | F_SYNTAX | F_ERROR | F_SEVERE | F_FATAL HIDE = [HIDE] [HIDE_LOG] [HIDE_PRINT] [HIDE_REPORT] NONE: no action HIDE: Suppress every error output HIDE_LOG: Error output to error log file is suppressed HIDE_DISPLAY: Error output is suppressed HIDE_REPORT: Application-specific error output is suppressed F_WARNING: Error is output as a WARNING (remedy class = 0) F_SYNTAX: Error is output as a syntax error (remedy class = 2) 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) 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.

3.40.5 Conditional filter activation (P-STUP-00190)

P-STUP-00190	Conditional filter activation
Description	The individual error codes can be listed as numbers or texts, whereby the entire row must comply with the following syntax: (number | text) {, (number | text) } where: number:= CNC error number 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: *error_filter[0].conditional_param "*D012:", 123, 1001 Individual error texts are only checked when the SERCOS drive error S95 is read out. Error numbers are only checked in case of SERCOS drive errors (S21 and S22) and in case of ProfiDrive drive errors (parameter 945). * Note: The default value of variables is a blank string.

3.40.6 Output of additional error information (P-STUP-00191)

P-STUP-00191	Output of additional error information
Description	This text is forwarded transparently via the CNC_ERROR_INFO data structure if the filter condition applies, i.e. users have a possibility of conditionally also including additional error information in the output.
Parameter	error_filter[i].conditional_output where i = 0 ... 3 (maximum number of filters, application-specific)
Data type	STRING
Data range	Maximum 32 characters
Dimension	----
Default value	*
Remarks	* Note: The default value of variables is a blank string.

3.41 Parameters for the Job Manager (jobmanager.*)

The parameters for the Job Manager are divided into two main structures:

- jobmanager.coding[] → Links to the declaration of data transferred to the PLC (data is transferred in binary form to the PLC) and
- jobmanager.group[] → Configuration of the Job Manager with independent logical groups and their clients (masters) and agents (slaves)

Note: If the jobmanager.group[] is not declared, the Job Manager function is deactivated. Remarks in each of the parameters describe an alternative deactivation of the Job Manager functions.

3.41.1 Number of PLC parameter lists (P-STUP-00203)

P-STUP-00203	**Number of PLC parameter lists**
Description	Number of lists which describe the parameters sent to the PLC units (corresponding to a data structure)
Parameter	jobmanager.codings
Data type	UNS08
Data range	0 to 2 (application-specific)
Dimension	----
Default value	0
Remarks	

3.41.2 Name of the PLC parameter list file (P-STUP-00204)

P-STUP-00204	**Path and name of a PLC parameter list file**
Description	Path and name of a file which describes the data structure of binary start parameters sent to the PLC
Parameter	jobmanager.coding[i].list where i = 0 to 2 (max. number of lists: 2, application-specific)
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	----
Default value	-
Remarks	

3.41.3 Type of PLC parameter list file (P-STUP-00205)

P-STUP-00205	Type of PLC parameter list file
Description	Type/format of the description of the data transferred to the PLC and saved in the file jobmanager.coding[i].list (P-STUP-00204 [▶ 58]).
Parameter	jobmanager.coding[i].list where i = 0 to 2 (max. number of lists: 2, application-specific)
Data type	UNS08
Data range	(Enumeration) 0: The format is identical to the declaration of V.E variables 1: DDL list (application-specific)
Dimension	----
Default value	1
Remarks	

3.41.4 Log. ID of a client in Job Manager group (P-STUP-00206)

P-STUP-00206	Logical ID of a client channel in a Job Manager group
Description	The parameter defines the logical ID of a client channel. This logical ID can be requested in the channel agent by V.G.IP_NR. Due to the necessary uniqueness, no second client channel may use the same logical ID in the same Job Manager group.
Parameter	jobmanager.group[i].master[j].log_id where i = 0, 1 (index of the Job Manager group, max. 1) where j = 0 … n (index of a continuous list element. n: application-specific)
Data type	UNS16
Data range	1 … 65536
Dimension	----
Default value	0
Remarks	IMPORTANT: If the agent is not called by a client, the call of "V.G.IP_NR" returns the "log_id" of the agent.

3.41.5 Client channel in a Job Manager group (P-STUP-00207)

P-STUP-00207	Logical ID of a client channel in a Job Manager group
Description	The client channel (master) designated by this parameter in a Job Manager group corresponds to an existing CNC channel. It cannot be assigned to any other group, neither as an agent (slave) or as a client (master) The number used must correspond to an existing channel number.
Parameter	jobmanager.group[i].master[j].channel_id (applications-specific) where i = 0, 1 (index of Job Manager group, max. 1) where j = 0..n (index of a continuous list element. n: application-specific)
Data type	UNS16
Data range	1 to 12 (application-specific)
Dimension	----
Default value	0 *
Remarks	* The value corresponds to the statement: Parameter is not used. If all jobmanager.group[i].master[j].channel_id parameters in a Job Manager group are "0", the Job Manager is deactivated for the group. The client channel corresponds to a “normal” channel. The Job Manager configuration extends the command set to include Job Manager commands (see related documentation). This also includes commanding all agents (slaves) in the same Job Manager group with jobs.

3.41.6 Log. ID of a channel agent in a Job Manager group (P-STUP-00208)

P-STUP-00208	Logical ID of a channel agent in a Job Manager group
Description	The parameter defines the logical ID of a channel agent. Every agent (slave) in a Job Manager group is invoked by commands from the client by its logical ID "log_id” at the start. Due to the necessary uniqueness, no second channel agent may use the same logical ID in the same Job Manager group.
Parameter	Jobmanager.group[i].cnc_slave[j].log_idwhere where i = 0, 1 (index of Job Manager group, max. 1) where j = 0..n (index of a continuous list element. n: applications-specific)
Data type	UNS16
Data range	1 … 65536
Dimension	----
Default value	0
Remarks	There are two types of agent: CNC channels and PLC units. The logical ID always refers to a particular type

3.41.7 Channel agent in a Job Manager group (P-STUP-00209)

P-STUP-00209	Channel agent in a Job Manager group
Description	The channel agent (slave) designated by this parameter in a Job Manager group corresponds to an existing CNC channel. It cannot be assigned to any other group, neither as agent (slave) or as client (master). The number used must correspond to an existing channel number.
Parameter	jobmanager.group[i]. cnc_slave[j].channel_id (application-specific) where i = 0, 1 (index of Job Manager group, max. 1) where j = 0..n (index of a continuous list element. n: application-specific)
Data type	UNS16
Data range	1 … (application-specific)
Dimension	----
Default value	0
Remarks	The channel agent behaves like a "normal" channel. It has the additional property of being requested by any master in the same Job Manager group to execute a job. Job completion is signalled back to the client in the controller.

3.41.8 Log. ID of a PLC unit agent in a Job Manager group (P-STUP-00210)

P-STUP-00210	Logical ID of a PLC unit agent in a Job Manager group
Description	The parameter defines the logical ID of a PLC unit agent. Every PLC agent (slave) in a Job Manager group is invoked by commands from the client by its logical ID "log_id". Due to the necessary uniqueness, no second PLC unit agent may use the same logical ID in the same Job Manager group.
Parameter	jobmanager.group[i]. Plc_slave[j].log_id where i = 0, 1 (index of the Job Manager group, max. 1) where j = 0…n (index of a continuous HLI list element. n: application-specific)
Data type	UNS16
Data range	1 … 65536
Dimension	----
Default value	0
Remarks	

3.41.9 PLC unit agent in a Job Manager group (P-STUP-00211)

P-STUP-00211	PLC unit agent in a Job Manager group
Description	The PLC unit agent (slave) in a Job Manager group designated by this parameter corresponds to an interface on the HLI. After assignment it cannot be assigned to any other group.
Parameter	jobmanager.group[i].hli_index (application-specific) where i = 0, 1 (index of the Job Manager group, max. 1) where j = 0..n (index of a continuous HLI list element n: application-specific)
Data type	UNS08
Data range	0 to 31 (application-specific)
Dimension	----
Default value	0
Remarks	A PLC unit agent has the property of being requested by any master in the same Job Manager group to execute a job. Job completion is signalled back to the client in the controller.

3.41.10 Param.list of a PLC unit agent in a Job Manager group (P-STUP-00212)

P-STUP-00212	Parameter list of a PLC unit agent in a Job Manager group
Description	The PLC unit agent (slave) in a Job Manager group can receive parameters from the client at the start. The data structure described a declared parameter list in "jobmanager.coding[i].list" (P-STUP-00204 [▶ 58]). The parameter required here corresponds to the index [i] in "jobmanager.coding[i].list".
Parameter	Jobmanager.group[i].plc_slave[j].coding where i = 0, 1 (index of the Job Manager group, max. 1) where j = 0..n (index of a continuous HLI list element n: application-specific)
Data type	UNS08
Data range	0 to 2 (application-specific)
Dimension	----
Default value	0
Remarks	

3.42 Parameter for scene presentation

3.42.1 Enable scene functionality (P-STUP-00138)

P-STUP-00138	Enable scene functionality
Description	This parameter can enable the scene functionality. The required memory for the scene database is created.
Parameter	enable_scene
Data type	BOOLEAN
Data range	TRUE/ FALSE
Dimension	---
Default value	FALSE
Remarks	

3.42.2 Name of the list file for scene presentation (P-STUP-00137)

P-STUP-00137	Name of the list file for scene presentation
Description	This parameter defines the path and filename of the configuration file for scene presentation.
Parameter	scene_mds
Data type	STRING
Data range	Maximum 256 characters (application-specific)
Dimension	---
Default value	*
Remarks	* Note: The default value of variables is a blank string.

3.43 Setting units for PLCopen

Release Note

These functions are available as of CNC Build V2.11.2808.02.

3.43.1 Positions for linear axes (P-STUP-00192)

P-STUP-00192	Setting the units of linear axis positions for PLCopen
Description	This parameter can be set for PLCopen function blocks and transfers the unit of linear axis position specifications to the function block.
Parameter	plcopen_unit.linear.position
Data type	STRING
Data range	um Positions in µm mm Positions in mm m Positions in m
Dimension	----
Default value	*
Remarks	If none of the above mentioned unit specifications is configured, positions are specified in the unit 0.1µm * Note: The default value of variables is a blank string.

3.43.2 Velocities for linear axes (P-STUP-00193)

P-STUP-00193	Setting the linear axis velocity unit for PLCopen
Description	This parameter can be set for PLCopen function blocks and transfers the unit of linear axis velocity specifications to the function block.
Parameter	plcopen_unit.linear.velocity
Data type	STRING
Data range	um/s Velocity specifications in µm/s mm/min Velocity specifications in mm/min mm/min Velocity specifications in m/min m/s Velocity specifications in m/s mm/s Velocity specifications in mm/s
Unit	----
Default value	um/s
Remarks	

3.43.3 Velocities for linear axes (P-STUP-00194)

P-STUP-00194	Setting the linear axis velocity unit for PLCopen
Description	This parameter can be set for PLCopen function blocks and transfers the unit of linear axis velocity specifications to the function block.
Parameter	plcopen_unit.linear.acceleration
Data type	STRING
Data range	mm/s2 Acceleration in mm/s² m/s2 Acceleration in m/s² mm/min2 Acceleration in mm/min²
Unit	----
Default value	mm/s2
Remarks	

3.43.4 Jerk for linear axes (P-STUP-00195)

P-STUP-00195	Setting the linear axis jerk unit for PLCopen
Description	This parameter can be set for PLCopen function blocks and transfers the unit of linear axis jerk specifications to the function block.
Parameter	plcopen_unit.linear.jerk
Data type	STRING
Data range	mm/s3 Jerk in mm/s³ m/s3 Jerk in m/s² mm/min3 Jerk in mm/min²
Unit	----
Default value	mm/s3
Remarks	

3.43.5 Positions for rotary axes (P-STUP-00196)

P-STUP-00196	Setting the units of rotary axis positions for PLCopen
Description	This parameter can be set for PLCopen function blocks and transfers the unit of rotary axis position specifications to the function block.
Parameter	plcopen_unit.rotatory.position
Data type	STRING
Data range	mdeg Positions in mil° deg Positions in ° rev Positions in revolutions U Positions in revolutions
Dimension	----
Default value	*
Remarks	If none of the above mentioned unit specifications is configured, positions are specified in the unit 10^-4° * Note: The default value of variables is a blank string.

3.43.6 Speeds for rotary axes (P-STUP-00197)

P-STUP-00197	Setting the units of rotary axis speeds for PLCopen
Description	This parameter can be set for PLCopen function blocks and transfers the unit of rotary axis speed specifications to the function block.
Parameter	plcopen_unit.rotatory.speed
Data type	STRING
Data range	mdeg/s Speeds in milli°/s U/min Speeds in revolutions/s U/s Speeds in revolutions/s rpm Speeds in revolutions/min rev/min Speeds in revolutions/min rev/s Speeds in revolutions/s deg/min Speeds in °/min deg/s Speeds in °/s
Unit	----
Default value	mdeg/s
Remarks	

3.43.7 Speeds for rotary axes (P-STUP-00198)

P-STUP-00198	Setting the units of rotary axis speed for PLCopen
Description	This parameter can be set for PLCopen function blocks and transfers the unit of rotary axis speed specifications to the function block.
Parameter	plcopen_unit.rotatory.acceleration
Data type	STRING
Data range	m/s2 Acceleration in m/s² rev/s2 Acceleration in revolutions/s² U/s2 Acceleration in revolutions/s² deg/min2 Acceleration in °/s²
Unit	----
Default value	deg/s2
Remarks	

3.43.8 Jerk for rotary axes (P-STUP-00199)

P-STUP-00199	Setting the units of rotary axis jerk for PLCopen
Description	This parameter can be set for PLCopen function blocks and transfers the unit of rotary axis jerk specifications to the function block.
Parameter	plcopen_unit.rotatory.jerk
Data type	STRING
Data range	deg/s3 Jerk in °/s³ rev/s3 Jerk in revolutions/s³ U/s3 Jerk in revolutions/s³ deg/min3 Jerk in °/min³
Unit	----
Default value	deg/s3
Remarks	

3.44 Common transmission of T and D numbers for tool requests (P-STUP-00029)

P-STUP-00029	**Common transmission of T and D numbers for tool requests sent to an external tool management system.**
Description	This parameter changes the protocol of tool requests between the CNC and an external tool management system. When the parameter is active, the programmed D number is sent together with the programmed T number. The D number is then placed in the variant field of the request and the T number is in the basic field. When the parameter is inactive, two separate requests are sent to the external tool management system.
Parameter	common_t_d_request
Data type	BOOLEAN
Data range	0: Transmit T and D numbers in separate requests. 1: Transmit T and D numbers together actively.
Dimension	---
Default value	0
Remarks	Parameter available as of V3.1.3080.13 and V3.1.3107.46 When P-STUP-00029 is active, the parameter P-CHAN-00016 (ext_wzv_vorhanden) must be active in every channel. If this is not the case, the warning ID 22243 is output and the channel parameter P-CHAN-00016 is set to active. P-STUP-00029 does not change the operation mode of the NC commands #TOOL DATA, #TOOL PREP and #TOOL LIFE READ. P-STUP-00029 may not be used with an old external tool management system. The interface must be at least version no. 1.0.0. Otherwise, error ID 22247 is output. The value of the parameter is specified at initial handshake via the interface version of the external tool management system.

3.45 Time-optimised setting for the simulation of online machining time calculation (P-STUP-00022)

P-STUP-00022	Time-optimised setting for channel mode of the simulation of online machining time calculation
Description	This parameter enables the time-optimised setting for the channel mode Simulation of online production time calculation in the channel. When the parameter is enables, functionalities not required in the NC channel are disabled and the supply of display data and control units of the PLC are reduced to the required data.
Parameter	online_prod_time_opt
Data type	BOOLEAN
Data range	0 – No time optimisation 1 – Time optimisation enabled
Dimension	----
Default value	0
Remarks	The parameter is available as of V3.1.3079.19.

3.46 MultiCore trace

The following parameters are used to log scheduling events within a task.

The parameters can be used for any task.

3.46.1 Number of logging entries for logging (P-STUP-00213)

P-STUP-00213	Number of logging entries for logging
Description	This parameter sets the maximum number of log entries for the corresponding task. Real-time events are logged in these entries for diagnostic purposes. After the number is reached, logging stops automatically. With a value=0, no log file is generated at CNC start-up.
Parameter	trace.geo.max_records
Data type	SGN32
Data range	0 <= max_records < MAX_UNS32
Dimension	----
Default value	0
Remarks	Parameter available as of CNC Build V3.1.3077 and higher

3.46.2 Name of the output file (P-STUP-00214)

P-STUP-00214	Name of the output file
Description	This parameter is used to specify the name of the output file for logging the corresponding task. If no path is specified for the output file, the default path or the main directory of the NC controller is used.
Parameter	task_trace.geo.filename
Data type	STRING
Data range	<Filename with relative / absolute path>
Dimension	----
Default value	<TwinCATInstallation>\Components\Mc\CNC\Diagnostics\MultiCore-Startup.log
Remarks	Parameter available as of CNC Build V3.1.3077 and higher

3.46.3 Output file for previous logs (P-STUP-00215)

P-STUP-00215	Name of the history file
Description	This parameter is used to specify the name of the history file for logging the corresponding task. The file is used to output the history logs. If no path is specified for the file, the default path or the main directory of the NC controller is used.
Parameter	task_trace.geo.history_filename
Data type	STRING
Data range	<Filename with relative / absolute path>
Dimension	----
Default value	<TwinCATInstallation>\Components\Mc\CNC\Diagnostics\MultiCore-History.log
Remarks	Parameter available as of CNC Build V3.1.3077 and higher

3.47 Parameters for channel synchronisation (signal_wait.*)

3.47.1 Activate acknowledgement for signals (P-STUP-00118)

P-STUP-00118	Activate acknowledgement for signals
Description	This parameter activates a mode that waits until storage of the signal is acknowledged with default signals sent at decoder level. This does not affect broadcast signals and synchronised signals at interpolator level (#SIGNAL SYN[…]) . The maximum number of stored events can be defined in P-STUP-00119 [▶ 71]. Error ID 250001 can no longer occur at decoder level if P-STUP-00118 is set.
Parameter	signal_wait.use_signal_acknowledge
Data type	BOOLEAN
Data range	0: Function deactivated 1: Function activated
Dimension	---
Default value	0
Remarks	Parameter available as of V3.1.3081.02 and V3.1.3108.02

3.47.2 Number of #SIGNAL/#WAIT events (P-STUP-00119)

P-STUP-00119	Number of #SIGNAL/#WAIT events
Description	This parameter defines the maximum number of #SIGNAL and #WAIT events that can be stored. This refers to the sum of the two both events. Storage is activated by P-STUP-00018 [▶ 71].
Parameter	signal_wait.nbr_events
Data type	UNS32
Data range	Number of channels < P-STUP-00119 < 1000
Dimension	---
Default value	150
Remarks	Parameter available as of V3.1.3081.02 and V3.1.3108.02

4 Example of assigning the start-up list

Configuration with 2 channels and a total of 6 axes:

```
# ****************************************************************
# ****************************************************************
configuration TWO_CHANNEL_CKONFIGURATION
kanal_anzahl 2
ext_var_max 200
plc_mode 0
sercos_hochlauf 1
listen ASCII
# ****************************************************************
# Lists of 1st channel
# ****************************************************************
default_sda_mds ..\listen\default_sda.lis
sda_mds[0] ..\listen\sda_mds1.lis
werkz_data[0] ..\listen\werkz_d1.lis
nullp_data[0] ..\listen\nullp_d1.lis
pzv_data[0] ..\listen\pzv_d1.lis
ve_var[0] ..\listen\ext_var1.lis
fb_storage_size[0] 0x200000
# ****************************************************************
# Lists of 2nd channel
# ****************************************************************
sda_mds[1] ..\listen\sda_mds2.lis
werkz_data[1] ..\listen\werkz_d2.lis
nullp_data[1] ..\listen\nullp_d2.lis
pzv_data[1] ..\listen\pzv_d2.lis
ve_var[1] ..\listen\ext_var2.lis
fb_storage_size[1] 0x200000
# ****************************************************************
# Channel-independent lists
# ****************************************************************
hand_mds ..\listen\hand_mds.lis
hmi[0].objects ..\listen\objects1.lis
hmi[0].mode write+
hmi[1].objects ..\listen\objects2.lis
hmi[1].mode write+
channel[0].objects ..\listen\channel1.lis
channel[0].mode write+
channel[1].objects ..\listen\channel2.lis
channel[1].mode write+
rtconf_lis ..\listen\rtconf.lis
konf_path ..\listen
#
# ****************************************************************
# Axis machine data
# ****************************************************************
zahl_mds 6
default_achs_mds ..\listen\default_mds.lis
achs_mds[0] ..\listen\achsmds1.lis
achs_mds[1] ..\listen\achsmds2.lis
achs_mds[2] ..\listen\achsmds3.lis
achs_mds[3] ..\listen\achsmds4.lis
achs_mds[4] ..\listen\achsmds5.lis
achs_mds[5] ..\listen\achsmds6.lis
#
# ****************************************************************
# Offset value lists
# (masked by comment characters)
```

```
# ****************************************************************
# zahl_kw 4
# achs_kw[0] ..\listen\achskw1.lis
# achs_kw_log_ax_nr[0] 1
# achs_kw[1] ..\listen\achskw2.lis
# achs_kw_log_ax_nr[1] 2
# achs_kw[2] ..\listen\achskw3.lis
# achs_kw_log_ax_nr[2] 3
# achs_kw[3] ..\listen\achskw4.lis
# achs_kw_log_ax_nr[3] 4
#
# ****************************************************************
# Program paths:
# ****************************************************************
# path[ <channel_number> ].prg[ <Index> ]
# prg -> Program path specification
# log_nr -> logical program path number
# typ -> Program path type ( 0x01 main program path )
# ( 0x02 Subroutine path )
# ( 0x03 main program and subroutine path )
# priority -> Specifies the program path priority if
# several program paths of the same type are specified.
#
# Program path Channel 1
pfad[0].prg[0]        x:\nc_prg
pfad[0].log_nr[0]     1
pfad[0].typ[0]        0x03 # Main program and subroutine path
pfad[0].prioritaet[0] 1
#
pfad[0].prg[1]        x:\nc_prg\cycles
pfad[0].log_nr[1]     2
pfad[0].typ[1]        0x02 # Subroutine path
pfad[0].prioritaet[1] 2
#
pfad[0].prg[2]        x:\test
pfad[0].log_nr[2]     3
pfad[0].typ[2]        0x03 # Main program and subroutine path
pfad[0].prioritaet[2] 3
#
# Program path Channel 2
pfad[1].prg[0]        ..\prg
pfad[1].log_nr[0]     1
pfad[1].typ[0]        0x01 # Main program path
pfad[1].prioritaet[0] 1
#
pfad[1].prg[1]        ..\prg\sub
pfad[1].log_nr[1]     2
pfad[1].typ[1]        0x02 # Subroutine path
pfad[1].prioritaet[1] 2
#
End
```

5 Appendix

5.1 Channel scaling (configuration.channel[i].*)

This structure element defines the functions for decoding, path preparation and interpolation for each channel

Structure name	Index
configuration.channel[i]	i = 0 ... 11 (maximum number of channels: 12, application-specific)

5.1.1 Decoding (configuration.channel[i].decoder.*)

5.1.1.1 Defining the decoder functionalities (P-STUP-00050)

P-STUP-00050	Definition of decoder functions
Description	The parameter defines specific functionalities for decoding. This disables specific functions for testing or for performance reasons.
Parameter	configuration.channel[i].decoder.function
Data type	STRING
Data range	FCT_USE_CACHED_FILES: Enabling file caching FCT_VOL_COMP_COMPUTATION: Calculations for machine calibration -: No functionalities defined.
Dimension	----
Default value	*
Remarks	Parameterisation example: Caching of maximal 4 files of maximum 4096 bytes each. *configuration.channel[0].decoder.function FCT_USE_CACHED_FILES* *configuration.channel[0].decoder.max_cache_number 4* *configuration.channel[0].decoder.max_cache_size 4096* * Note: The default value of variables is a blank string.

5.1.1.2 Maximum number of possible cache files (P-STUP-00051)

P-STUP-00051	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.channel[i].decoder.max_cache_number
Data type	UNS32
Data range	0 <= P-STUP-00051 <= MAX(UNS32)
Dimension	----
Default value	0
Remarks	If the File Caching function is active with *FCT_USE_CACHED_FILES*, the default value is 4. Parameterisation example: Caching of maximal 6 files of maximum 6000 bytes each. *configuration.channel[0].decoder.function FCT_USE_CACHED_FILES* *configuration.channel[0].decoder.max_cache_number 6* *configuration.channel[0].decoder.max_cache_size 6000*

5.1.1.3 Maximum size of a cache file (P-STUP-00052)

P-STUP-00052	Maximum size of a cache file
Description	This parameter permits the user-specific definition of the maximum size of an NC program cache.
Parameter	configuration.channel[i].decoder.max_cache_size
Data type	UNS32
Data range	0<= P-STUP-00052 <= MAX(UNS32)
Dimension	----
Default value	0
Remarks	If the File Caching function is active with *FCT_USE_CACHED_FILES*, the default value is 4096. Parameterisation example: Caching of maximal 6 files of maximum 6000 bytes each. *configuration.channel[0].decoder.function FCT_USE_CACHED_FILES* *configuration.channel[0].decoder.max_cache_number 6* *configuration.channel[0].decoder.max_cache_size 6000*

5.1.1.4 Maximum number of local subroutine definitions (P-STUP-00053)

P-STUP-00053	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.channel[i].decoder.max_local_subroutine_definitions
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	50
Remarks	Parameterisation example: *configuration.channel[0].decoder.max_local_subroutine_definitions 70*

5.1.1.5 Maximum number of logged events (P-STUP-00054)

P-STUP-00054	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.channel[i].decoder.log_entry_number
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

5.1.1.6 Defining the type of logged events (P-STUP-00055)

P-STUP-00055	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.channel[i].decoder.log_level
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

5.1.1.7 Maximum V.I. user memory in bytes (P-STUP-00183)

P-STUP-00183	Maximum V.I. user memory in bytes
Description	This parameter defines the maximum memory size in bytes to be provided for V.I. variables at controller start-up.
Parameter	configuration.channel[0].decoder.vi_memory
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	The number and maximum usable memory must be defined in order to use V.I. variables. Select the memory to accommodate all the single variables and arrays.

5.1.1.8 Maximum number of creatable V.I. variables (P-STUP-00184)

P-STUP-00184	Maximum number of creatable V.I. variables
Description	This parameter defines the maximum number of V.I. variables which can be created and used.
Parameter	configuration.channel[0].decoder.vi_maximal_var_count
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	The number and maximum usable memory must be defined in order to use V.I. variables. Set the maximum number of variables so that all individual variables and all arrays each receive an entry. One array always counts as one entry.

5.1.1.9 Maximum number of measurement records for machine calibration (P-STUP-00185)

P-STUP-00185	**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.channel[i].decoder.max_vol_comp_measurement_records
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	Parameterisation example: A maximum of 50 measurement records are logged. *configuration.channel[0].decoder.function FCT_VOL_COMP_COMPUTATION* *configuration.channel[0].decoder.max_vol_comp_measurement_records 50*

5.1.2 Tool radius compensation (configuration.channel[i].tool_radius_comp.*)

5.1.2.1 Defining the functionalities for tool radius compensation (P-STUP-00080)

P-STUP-00080	**Definition of functionalities for tool radius compensation**
Description	This parameter defines individual functionalities for tool radius compensation.
Parameter	configuration.channel[i].tool_radius_comp.function
Data type	STRING
Data range	MULTI_PATH: 2-path configuration and 2-path programming active -: No functionalities defined.
Dimension	----
Default value	*
Remarks	* Note: The default value of variables is a blank string.

5.1.2.2 Maximum number of logged events (P-STUP-00081)

P-STUP-00081	**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.channel[i].tool_radius_comp.log_entry_number
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

5.1.2.3 Defining the type of logged events (P-STUP-00082)

P-STUP-00082	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.channel[i].tool_radius_comp.log_level
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

5.1.3 Path preparation (configuration.channel[i].path_preparation.*)

5.1.3.1 Defining the functionalities for path preparation (P-STUP-00060)

P-STUP-00060	**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.channel[i].path_preparation.function
Data type	STRING
Data range	See Defining the functionalities for path preparation (P-STUP-00060) [▶ 81]
Dimension	----
Default value	FCT_DEFAULT
Remarks	

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). Example: FCT_DEFAULT | FCT_LIFT_UP
FCT_EMF	Edge machining (sharp angle contours). Example: FCT_DEFAULT | FCT_EMF
FCT_EMF_POLY_OFF	Edge machining inactive with polynomials. 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. Example: FCT_DEFAULT | FCT_EMF_POLY_OFF
FCT_SYNC	Synchronisation of an axis on a path group. Example: FCT_DEFAULT | FCT_SYNC
FCT_PRECON	Optimised planning using #HSC[BSPLINE]. Example: FCT_DEFAULT | FCT_PRECON
FCT_LIFT_UP_TIME	Automatic lifting/lowering of an axis (time-based coupling). Example: FCT_DEFAULT | FCT_LIFT_UP_TIME
FCT_PTP	Dynamically optimised contouring of the complete contour. Example: FCT_DEFAULT | FCT_PTP
FCT_M_PRE_OUTPUT	Pre-output of M/H functions (microjoints). Example: FCT_DEFAULT | FCT_M_PRE_OUTPUT
FCT_SURFACE	HSC machining with Surface Optimiser Example: FCT_DEFAULT | FCT_SURFACE
FCT_SEG_CHECK	Block segmentation in combination with path-controlled offset of M functions (dwell time), see P-CHAN-00650 and Defining the functionalities for path preparation (P-STUP-00060) [▶ 81] Example: FCT_DEFAULT | FCT_SEG_CHECK
FCT_NIBBLING	Activate the nibbling function Example: FCT_DEFAULT | FCT_NIBBLING
FCT_PUNCHING	Activate the punching function Example: FCT_DEFAULT | FCT_PUNCHING
FCT_VSM	Activate the velocity smoothing function Example: FCT_DEFAULT | FCT_VSM as of V3.1.3079.21

5.1.3.2 Maximum number of blocks considered for pre-output of M functions (P-STUP-00061)

P-STUP-00061	**Maximum number of blocks considered for pre-output of M functions**
Description	This parameter permits the configuration of the look-ahead range for the pre-output of M functions (see [FCT-C1]).
Parameter	configuration.channel[i].path_preparation.m_pre_output_lookahead
Data type	UNS32
Data range	0 … 1000
Dimension	----
Default value	10
Remarks	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. **Parameterisation example:** *configuration.channel[0].path_preparation.function FCT_DEFAULT | FCT_M_PRE_OUTPUT* *configuration.channel[0].path_preparation.m_pre_output_lookahead* 15

Programing Example

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
```

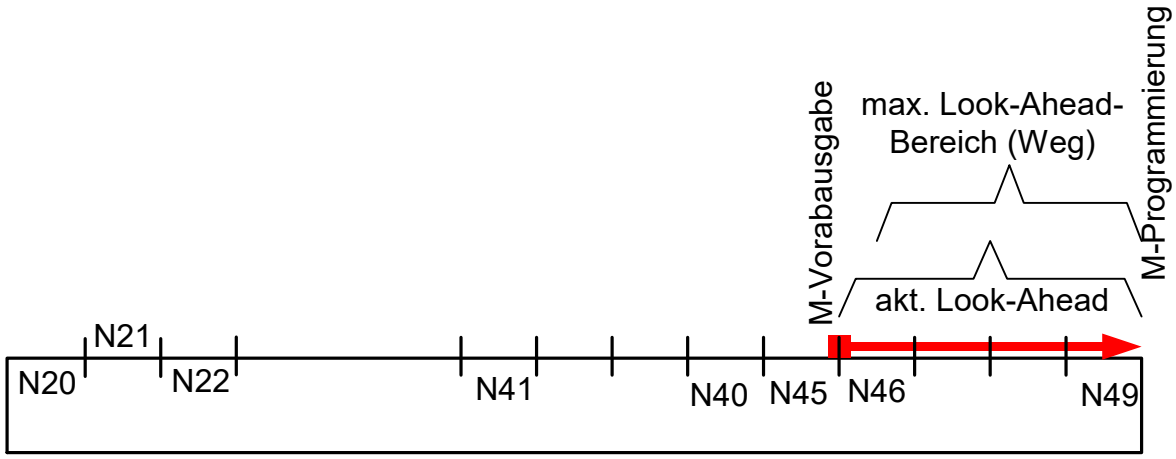


Fig. 1: Limits the pre-output to the maximum look-ahead range (default 10 blocks).

Notice

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

5.1.3.3 Maximum path for pre-output of M functions (P-STUP-00062)

P-STUP-00062	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. 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.channel[i].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 If the maximum number of blocks P-STUP-00061 [▶ 83] 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. Without an explicit specification, the range is not additionally limited (only by the number of blocks P-STUP-00061 [▶ 83]). 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. **Parameterisation example:** *configuration.path_preparation.function FCT_DEFAULT | FCT_M_PRE_OUTPUT* *configuration.* channel[i].*path_preparation.m_pre_output_lookahead 100* *configuration.* channel[i]. *path_preparation.m_pre_output_max_distance 35000 [0.1µm]*

Programing Example

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
```

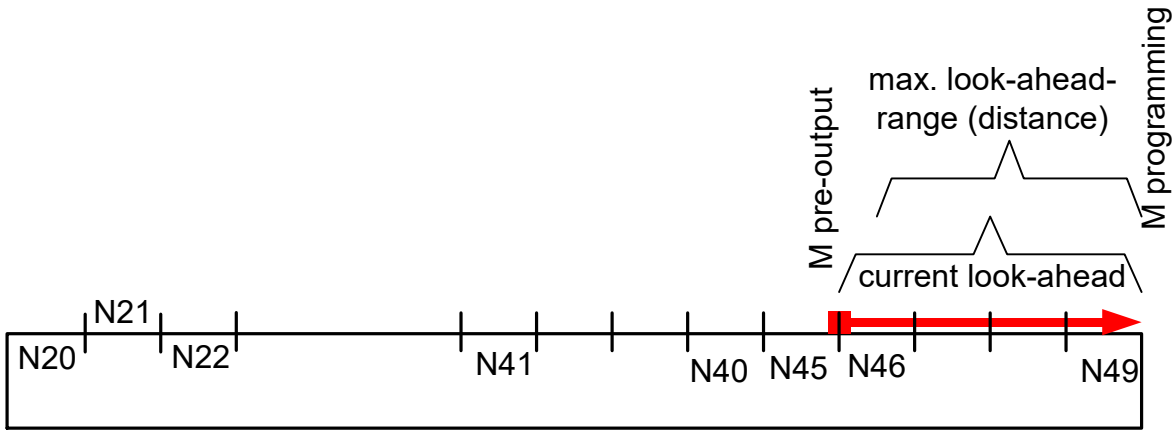


Fig. 2: Distance-related limiting of pre-output to maximum look-ahead range.

5.1.3.4 Maximum number of logged events (P-STUP-00063)

P-STUP-00063	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.channel[i].path_preparation.log_entry_number
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

5.1.3.5 Defining the type of logged events (P-STUP-00064)

P-STUP-00064	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.channel[i].path_preparation.log_level
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

5.1.4 Interpolation (configuration.channel[i].interpolator.*)

Settable functions for position control.

5.1.4.1 Defining interpolator functionalities (P-STUP-00070)

P-STUP-00070	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 to calculate deceleration distance and dynamic planning.
Parameter	configuration.channel[i].interpolator.function
Data type	STRING
Data range	See Defining interpolator functionalities (P-STUP-00070) [▶ 88].
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.
FCT_SYNC	Synchronisation of an axis on a path group. 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). Example: FCT_IPO_DEFAULT | FCT_LIFT_UP_TIME
FCT_SHIFT_NCBL	Path-controlled offset of M functions (dwell time). 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 Example: FCT_IPO_DEFAULT | FCT_CALC_STATE_AT_T
FCT_CALC_TIME	Calculation of interpolation time to next feed block (G01,G02,G03). 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
FCT_EXTENSION_EQUIDIST	Die-sinking EDM Planetary expansion

The look-ahead buffer size values specified above apply as of CNC Builds V2.11.2800 and higher; the following settings apply to CNC Build V2.11.20xx:

FCT_LOOK_AHEAD_LOW	30 blocks
FCT_LOOK_AHEAD_STANDARD	70 blocks
FCT_LOOK_AHEAD_HIGH	120 blocks

5.1.4.2 User-specific size of look-ahead buffer (P-STUP-00071)

P-STUP-00071	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. The parameter is only evaluated if P-STUP-00070 [▶ 88] is set with FCT_LOOK_AHEAD_CUSTOM.
Parameter	configuration.channel[i].interpolator.number_blocks_lah *
Data type	UNS32
Data range	0 ... 10000
Dimension	----
Default value	120
Remarks	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. As of Build V3.1.3067.07 the upper limit of the data range is 500 blocks. If #SLOPE[TYPE=STEP] is used, the upper limit is 10000 blocks as of Build V3.1.3060.0. * P-STUP-00071 in V2.11.20 and higher : configuration.channel[i].interpolator.parameter

5.1.4.3 Maximum number of logged events (P-STUP-00072)

P-STUP-00072	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.channel[i].interpolator.log_entry_number
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	40
Remarks	

5.1.4.4 Defining the type of logged events (P-STUP-00073)

P-STUP-00073	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.channel[i].interpolator.log_level
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	0
Remarks	

5.1.4.5 Number of logs of the dynamic coordinate system (P-STUP-00074)

P-STUP-00074	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 dynCS 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.channel[i].interpolator.dyn_cs_history_max
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	----
Default value	20
Remarks	

5.1.4.6 Reducing interpolator computing time (P-STUP-00075)

P-STUP-00075	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. **Example:** number_blocks_lah = 10000, blocks_per_call = 1000 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.channel[i].interpolator.blocks_per_call
Data type	UNS32
Data range	1 ... The value is defined by P-STUP-00070 [▶ 88] .
Unit	----
Default value	200
Remarks	Parameter available as of V2.11.2033

5.1.4.7 Maximum number of contour elements in the look-ahead contour (P-STUP-00076)

P-STUP-00076	Maximum number of logged contour elements in the look-ahead area.
Description	This parameter can be used to set the maximum number of stored motion blocks that can be supplied to the PLC in advance. The CNC command #CONTOUR LOOKAHEAD LOG [] can be used to activate the save function. This function is only active when FCT_CONTOUR_LAH is enabled in P-STUP-00070 [▶ 88]. `configuration.channel[0].interpolator.function FCT_IPO_DEFAULT|` **`FCT_CONTOUR_LAH`**
Parameter	configuration.channel[i].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 V3.1.3104.07

5.2 Glossary

General abbreviations and terms are documented in the ISG Glossary.

5.3 Suggestions, corrections and the latest documentation

Did you find any errors? Do you have any suggestions or constructive criticism? Then please contact us at documentation@isg-stuttgart.de. The latest documentation is posted in our Online Help (DE/EN):

QR code link: https://www.isg-stuttgart.de/documentation-kernel/

The link above forwards you to:

https://www.isg-stuttgart.de/fileadmin/kernel/kernel-html/index.html

Notice

Change options for favourite links in your browser;

Technical changes to the website layout concerning folder paths or a change in the HTML framework and therefore the link structure cannot be excluded.

We recommend you to save the above "QR code link" as your primary favourite link.

PDFs for download:

DE:
https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

EN:
https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

E-Mail: documentation@isg-stuttgart.de

Keyword index

P

PART 6

[MDS-TOOL] tool_data

6 [MDS-TOOL] tool_data

Preface

Legal information

This documentation was produced with utmost care. The products and scope of functions described are under continuous development. We reserve the right to revise and amend the documentation at any time and without prior notice.

No claims may be made for products which have already been delivered if such claims are based on the specifications, figures and descriptions contained in this documentation.

Personnel qualifications

This description is solely intended for skilled technicians who were trained in control, automation and drive systems and who are familiar with the applicable standards, the relevant documentation and the machining application.

It is absolutely vital to refer to this documentation, the instructions below and the explanations to carry out installation and commissioning work. Skilled technicians are under the obligation to use the documentation duly published for every installation and commissioning operation.

Skilled technicians must ensure that the application or use of the products described fulfil all safety requirements including all applicable laws, regulations, provisions and standards.

Further information

Links below (DE)

https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

or (EN)

https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

contains further information on messages generated in the NC kernel, online help, PLC libraries, tools, etc. in addition to the current documentation.

Disclaimer

It is forbidden to make any changes to the software configuration which are not contained in the options described in this documentation.

Trade marks and patents

The name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® and the associated logos are registered and licensed trade marks of ISG Industrielle Steuerungstechnik GmbH.

The use of other trade marks or logos contained in this documentation by third parties may result in a violation of the rights of the respective trade mark owners.

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

➢ 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.

Attention

Restriction or error

This icon describes restrictions or warns of errors.

Notice

Tips and other notes

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

Example

General example

Example that clarifies the text.

Programing Example

NC programming example

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

Release Note

Specific version information

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

Overview of tool parameters

The overview of tool parameters offsets is sorted into a 4-column table.

- Column 1 contains the unambiguous identifier of the axis parameter called the “ID” which consists of the prefix “P-TOOL” and a unique 5-digit number,
  e.g. P-TOOL-00001.
- Column 2 represents the data structure which defines the parameter,
  e.g. wz[i].
  The structure is a categorisation aid and is described in the following section.
- Column 3 contains the “parameter” with its exact name,
  e.g. typ
  The important thing is that “structure”+”parameter” always belong together and must be configured in exactly the same way in the axis parameter list,
  e.g. wz[i].typ
- Column 4 contains the “functionality” in a summarised term/short description,
  e.g. Differentiation between tool types.

ID	Structure	Parameter	Functionality/short description
P-TOOL-00001 [▶ 105]	wz[i].	typ	Type
P-TOOL-00002 [▶ 106]	wz[i].	srk_lage	Cutter edge position
P-TOOL-00003 [▶ 107]	wz[i].	laenge	Tool length
P-TOOL-00004 [▶ 107]	wz[i].	radius	Tool radius
P-TOOL-00005 [▶ 110]	wz[i].	gueltig	Validity flag
P-TOOL-00006 [▶ 110]	wz[i].	ax_versatz[j]	Axis offsets
P-TOOL-00007 [▶ 111]	wz[i].	param[j]	Additional tool parameters
P-TOOL-00008 [▶ 112]	wz[i].	mass_einheit	Unit of length, radius and axis offsets
P-TOOL-00009 [▶ 112]	wz[i].kinematic.	param[j]	Kinematic parameters
P-TOOL-00010 [▶ 113]	wz[i].	tool_fixed	Tool fixed / alignable
P-TOOL-00011 [▶ 113]	wz[i].	kin_id	Kinematic ID
P-TOOL-00012 [▶ 115]	wz[i].	log_ax_nr_spdl	Logical spindle axis number
P-TOOL-00013 [▶ 115]	wz[i].	vb_min	Minimum rotation speed

ID	Structure	Parameter	Functionality/short description
P-TOOL-00014 [▶ 116]	wz[i].	vb_max	Maximum rotation speed
P-TOOL-00015 [▶ 116]	wz[i].	a_max	Maximum acceleration
P-TOOL-00016 [▶ 116]	wz[i].	gear_ratio_num	Gear ratio of a tool (numerator)
P-TOOL-00017 [▶ 117]	wz[i].	gear_ratio_denom	Gear ratio of a tool (denominator)
P-TOOL-00018 [▶ 117]	wz[i].	gear_inv_direction	Reversal of rotation direction by gear
P-TOOL-00019 [▶ 118]	wz[i].	gear_inv_direction_no_stopm	Reversal of rotation direction without spindle standstill
P-TOOL-00020 [▶ 129]	wz[i].path[j].	radius	Tool radius
P-TOOL-00030 [▶ 119]	wz[i].	grinding_wear_const	Wear constant
P-TOOL-00031 [▶ 119]	wz[i].	grinding_max_infeed	Maximum discrete infeed
P-TOOL-00100 [▶ 121]	wz[i].linkpoint_data.	name	Name
P-TOOL-00101 [▶ 121]	wz[i].linkpoint_data.	mountpoint	Mount point
P-TOOL-00102 [▶ 122]	wz[i].linkpoint_data.	translation[k]	Translation of main axes
P-TOOL-00103 [▶ 122]	wz[i].linkpoint_data.	rotation[k]	Rotation of main axes
P-TOOL-00104 [▶ 122]	wz[i].linkpoint_data.	ax_nr	axis number
P-TOOL-00105 [▶ 123]	wz[i].linkpoint_data.	trans_rot	Translation/rotation of linkpoint
P-TOOL-00106 [▶ 123]	wz[i].linkpoint_data.	inverse	Effect of the movement on the linkpoint
P-TOOL-00107 [▶ 123]	wz[i].linkpoint_data.	visible	Use of the linkpoint
P-TOOL-00108 [▶ 124]	wz[i].linkpoint_data.	fixed	Position of linkpoint after axis exchange
P-TOOL-00109 [▶ 124]	wz[i].linkpoint_data.	arm_len	Maximum lever arm length
P-TOOL-00120 [▶ 125]	wz[i].gobject_data[j].	name	Name of the graphical object
P-TOOL-00121 [▶ 125]	wz[i].gobject_data[j].	linkpoint	Linkpoint name

ID	Structure	Parameter	Functionality/short description
P-TOOL-00122 [▶ 125]	wz[i].gobject_data[j].	group[k]	Group name
P-TOOL-00123 [▶ 126]	wz[i].gobject_data[j].	translation[k]	Translation of object
P-TOOL-00124 [▶ 126]	wz[i].gobject_data[j].	rotation[k]	Rotation of object
P-TOOL-00125 [▶ 126]	wz[i].gobject_data[j].	relative	Effect of shift and rotation
P-TOOL-00126 [▶ 127]	wz[i].gobject_data[j].	file	File containing description of object data
P-TOOL-00127 [▶ 127]	wz[i].gobject_data[j].	key[k]	Key name
P-TOOL-00128 [▶ 127]	wz[i].gobject_data[j].	value[k]	Value name
P-TOOL-00129 [▶ 128]	wz[i].gobject_data[j].	changed	Information about a change
P-TOOL-00138 [▶ 120]	wz[i].	grinding_disc_tilt_angle	Grinding disc tilt angle
P-TOOL-00140 [▶ 130]		tool_data_with_id	Free assignment of the tool number
P-TOOL-00141 [▶ 130]	wz[i].tool_id.	basic	Basic tool number
P-TOOL-00142 [▶ 131]	wz[i].tool_id.	sister	Sister tool number
P-TOOL-00143 [▶ 131]	wz[i].tool_id.	variant	Variant tool number
P-TOOL-00144 [▶ 131]	wz[i].tool_id.	sister_valid	Validity flag of the sister tool
P-TOOL-00145 [▶ 132]	wz[i].tool_id.	variant_valid	Validity flag of the variant tool
P-TOOL-00146 [▶ 108]	wz[i].	orientation_vector[j]	Tool orientation by specifying a vector
P-TOOL-00147 [▶ 114]	wz[i].	additional_settings	Extended tool settings (modes)

6.1 General description

6.1.1 Links to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

For technical reasons these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

6.1.2 Syntax and interpretation of ASCII list file

An interpreter copies the entries in the ASCII list file into identical internal structures which are then checked for plausibility. To ensure reliable controller start-up every time, any defective entries found by the plausibility check are replaced by default values.

Unknown entries are not taken over. These irregularities are displayed by warning messages. We advise you to investigate the cause for these warning messages and remove defective entries from the ASCII list file.

Notice

The following agreement applies to BOOLEAN data:

Value	Meaning
0	Definition of FALSE
1	Definition of TRUE

Notice

The following agreement applies to STRING data:

If a character string containing characters with a special meaning in ASCII lists (e.g. comment characters, spaces [▶ 103]) is assigned to a STRING type list parameter, this character string must be defined in inverted commas "..." (available as of V3.1.3081.0, V3.1.3108.0).

```
example[0].name "STRING_WITH_COMMENT( # /*)_CHARACTERS"
```

Trailing spaces are discarded on import. The entry..

```
example[0].name "STRING_WITH_POST_SPACES "
```

..has the same meaning as

```
example[0].name "STRING_WITH_POST_SPACES"
```

If the character string only contains characters without any special meaning, no inverted commas are required.

```
example[0].name STRING_WITH_STANDARD_CHARACTERS!
```

6.1.3 Comments in the ASCII list file

Comments can be in an entire line or can be added at the end of a line.

With a comment spanning an entire line, the comment character "'#" must be placed at the start of the line and followed by a blank.

If a comment is to be inserted at the end of a line, only a blank is required before the comment. Blank lines are also possible.

Example

Comments in the ASCII list file

```
#
*****************************************************************************
# Data
#
*****************************************************************************
#
# List comments after numerical values

dummy[1] 1 Comment
dummy[2] 1 # Comment
dummy[3] 1 ( Comment
dummy[4] 1 /* Comment
...
...
```

However, if a character string was assigned to the list parameter as a value in the line, any following comment must be opened by a bracket '('. The comment characters space, # and /* are not permitted.

If a '(' itself is part of the character string, the character string must be defined in inverted commas "..." (available as of V3.1.3081.0, V3.1.3108.0).

```
# List comments after strings

beispiel[0].bezeichnung STRING_1 (comment requires a '(' bracket!)

beispiel[1].bezeichnung" STRING_(2)" (comment requires a '(' bracket!)
```

6.2 Description of elements

6.2.1 Tool data (wz[i].*)

The 'wz[i]' structure defines the data of a tool with the number 'i'.

Structure name	Index
wz[i]	i = 0 to 200 (number of tool data records: 201, application-specific)

Notice

The following relationship exists between the D function and the tool data index 'i'.

D function	Tool data index 'i'.
D 0	0
D 1	1
D 2	2
D 3	3
D 4	4
D 5	5
D 6	6
D 7	7

The tool data index 'i' therefore corresponds to the tool number programmed in the NC program using the T or D command.

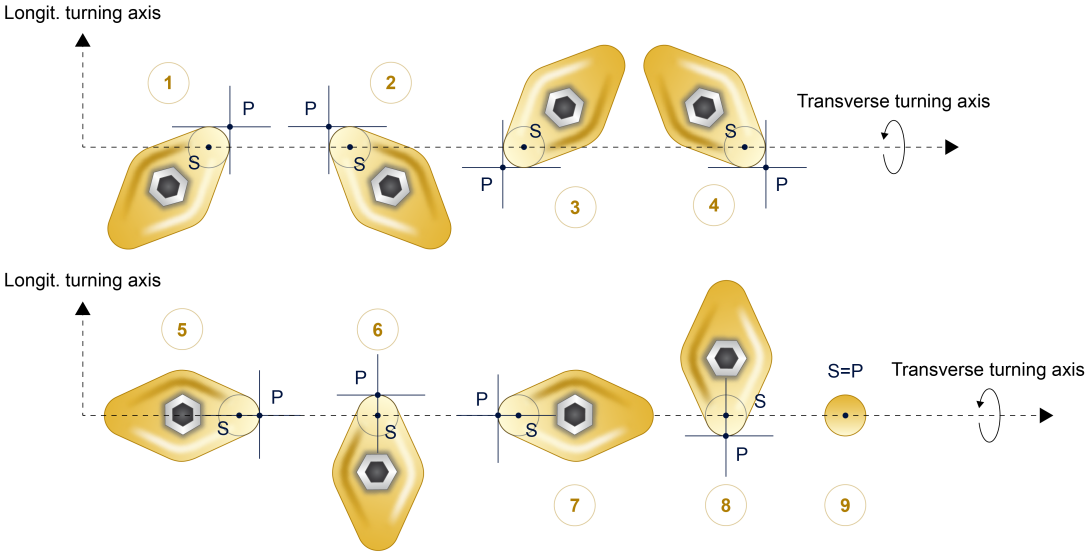
Attention

Data of the tool with index '0' is assigned the value '0' after the ASCII file is interpreted. In this way, all tool axis offsets can be calculated from the path motions by programming 'D0' in the NC program.

6.2.1.1 Type (P-TOOL-00001)

P-TOOL-00001	**Differentiation between tool types**
Description	This parameter is assigned to distinguish between tool types.
Parameter	wz[i].typ
Data type	UNS16
Data range	0: Milling tool 1: Turning tool 2: Grinding tool 3: Wire (erosion)
Unit	----
Default value	0
Remarks	Parameterisation example: Tool 5 is a milling tool *wz[5]typ 0*

6.2.1.2 Cutter position (P-TOOL-00002)

P-TOOL-00002	Cutter edge position
Description	With rotary tools (P-TOOL-00001 [▶ 105] = 1), this parameter specifies the orientation of the cutter relative to the machining plane.  Ident. codes 1 to 9 for orientation of the turning tool tip in the machining plane.
Parameter	wz[i].srk_lage
Data type	UNS16
Data range	0 < srk_lage ≤ 9
Dimension	----
Default value	0
Remarks	For detailed information regarding machining with turning tools, see [PROG]. Parameterisation example: The orientation of the cutter tip of Tool 6 is described by identification code 3. *wz[6].typ 1* *wz[6].srk_lage 3*

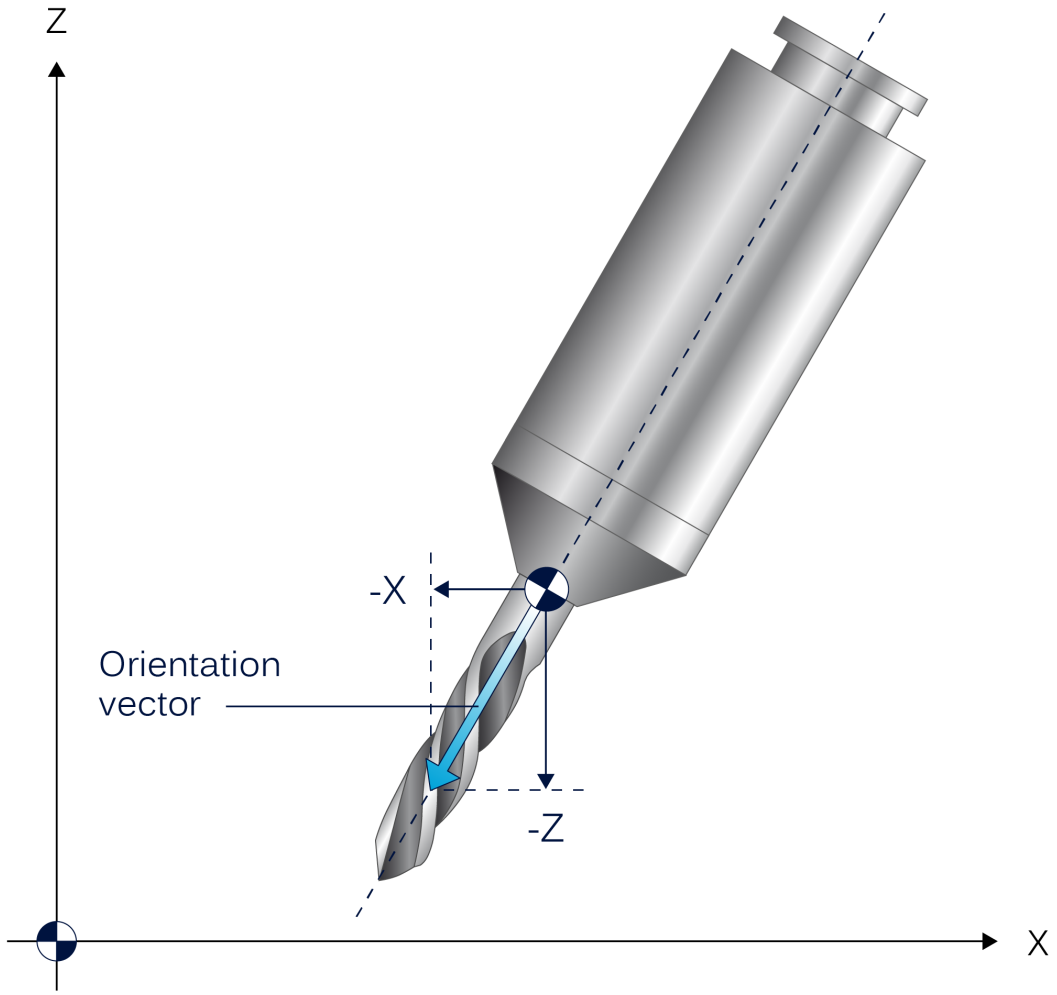
6.2.1.3 Tool length (P-TOOL-00003)

P-TOOL-00003	Tool length
Description	This parameter defines the tool length.
Parameter	wz[i].laenge
Data type	SGN32
Data range	MIN(SGN32) < length < MAX(SGN32)
Dimension	0.1µm or 0.0001 inch
Default value	0
Remarks	Parameterisation example: The length of Tool 5 is 60 mm *wz[5].laenge 600000*

6.2.1.4 Tool radius (P-TOOL-00004)

P-TOOL-00004	Tool radius
Description	This parameter defines the tool radius.
Parameter	wz[i].radius
Data type	SGN32
Data range	MIN(SGN32) ≤ radius < MAX(SGN32)
Dimension	0.1µm or 0.0001 inch
Default value	0
Remarks	Parameterisation example: The radius of Tool 5 is 20 mm *wz[5].radius 200000*

6.2.1.5 Tool orientation (P-TOOL-00146)

P-TOOL-00146	Tool orientation by specifying a vector
Description	In 2.5D mode, if a tool is in parallel or at a fixed angle to the main axes of the machine coordinate system, this orientation can be described as a vector. Vector components are defined in the direction of the tool tip, either scaled or unscaled, starting at the tool mount point. When selecting the tool, the orientation vector is used to form axis-specific offset components of the tool length. 
Parameter	wz[i].orientation_vector[j] where j=0...2
Data type	REAL64
Data range	MIN(REAL64) ≤ orientation_vector[j] ≤ MAX(REAL64)
Dimension	----
Default value	0.0
Remarks	The vector position always refers to the machine coordinate system. If the orientation vector is not assigned (=0), the tool length is considered using the default method (plane-specific, #TLAX). Parameterisation example: Tool T1 is oriented opposite to the positive Z axis direction: *wz[1].orientation_vector[0] 0* *wz[1].orientation_vector[1] 0*

	wz[1].orientation_vector[2] -1

6.2.1.6 Validity flag (P-TOOL-00005)

P-TOOL-00005	**Validity flag**
Description	If the data of a tool is allowed to be used, the validity flag must be set to TRUE.
Parameter	wz[i].gueltig
Data type	BOOLEAN
Data range	0/1
Dimension	----
Default value	0
Remarks	Parameterisation example: The data of Tool 5 is enabled. *wz[5].gueltig 1*

6.2.1.7 Axis offsets (P-TOOL-00006)

P-TOOL-00006	**Axis offsets**
Description	An offset parameter is defined for every tool and for every axis.
Parameter	wz[i].ax_versatz[j] where j = 0 ... 31 (Maximum number of axes per channel: 32, application-specific)
Data type	SGN32
Data range	MIN(SGN32) < ax_versatz[j] < MAX(SGN32)
Dimension	0.1µm or 0.0001 inch
Default value	0

Remarks	With active kinematic transformation, tool axis offsets are only considered in the axes which are not affected by the transformation function. Depending on the transformation type, they typically refer to all axes with index > 2 when RTCP is used.
	The axis-specific tool offsets of the first three axes (index 0, 1, 2) are not considered when transformation is active. If tool offsets are also to exert an effect on these axes during active transformation, enter the values in the specific kinematic offsets of the tool (P-TOOL-00009 [▶ 112]).
	Parameterisation example: The tool axis offsets are: 205 mm for the first axis, 206 mm for the second, 307 mm for the third and -408 mm for the fourth axis.
	`wz[5].ax_versatz[0] 2050000 #Offset 1st axis 205 mm`
	`wz[5].ax_versatz[1] 2060000 #Offset 2nd axis 206 mm`
	`wz[5].ax_versatz[2] 3070000 #Offset 3rd axis 307 mm`
	`wz[5].ax_versatz[3] -4080000 #Offset 4th Axis -408 mm`

6.2.1.8 Additional tool parameters (P-TOOL-00007)

P-TOOL-00007	**Additional tool parameters**
Description	The user can use this parameter to freely define additional parameters for each tool.
Parameter	wz[i].param[j] where j = 0 ... 59 (number of additional freely defined parameters per tool: 60, application-specific)
Data type	REAL64
Data range	MIN(REAL64) ≤ param[j] ≤ MAX(REAL64)
Dimension	----
Default value	0
Remarks	Parameterisation example: Additional free parameters are defined for Tool 5 *wz[5].param[0] 1* *wz[5].param[1] 20.5* *wz[5].param[2] 120*

6.2.1.9 Unit of length, radius and axis offsets (P-TOOL-00008)

P-TOOL-00008	Unit of length, radius and axis offsets
Description	This parameter defines the unit of length, radius and axis offsets of the tool.
Parameter	wz[i].mass_einheit
Data type	UNS16
Data range	0: Dimensional unit of length, radius and axis offsets is millimetres (10^{-4} mm) 1: Dimensional unit of length, radius and axis offsets is inch (10^{-4} inch)
Dimension	----
Default value	0
Remarks	This parameter is only used to adopt tool data for length (P-TOOL-00003 [▶ 107]), radius (P-TOOL-00004 [▶ 107]) and axis offsets (P-TOOL-00006 [▶ 110]) from a tool list. This parameter has no meaning if tool data is adopted from an external tool management system (e.g. from the PLC). Parameterisation example: The tool data of length, radius and axis offsets is specified in the unit 'millimetre'. *wz[5].mass_einheit 0*

6.2.1.10 Kinematic parameters (P-TOOL-00009)

P-TOOL-00009	Kinematic parameters
Description	These parameters are used for tool-dependent parameterisation of kinematic transformation (RTCP / TLC / TOOL ORI CS [PROG]). Assignment is defined application-specific.
Parameter	wz[i].kinematic.param[j] where j = 0 … 74 (maximum number of kinematic parameters, application-specific, syntax as of V263 and higher)
Data type	SGN32
Data range	MIN(SGN32) < param[j] < MAX(SGN32)
Dimension	0.1µm
Default value	0

Remarks	*wz[i].kinematic.wz_kopf_versatz[j] (Syntax up to V260)*
	
	In addition, it is possible to enter offsets for every kinematic parameter in the channel parameters P-CHAN-00094. If an element is assigned in both lists, the CNC performs an addition of the specified values.
	For more details regarding the parameterisation of kinematic transformation for 5-axis machining, see [KITRA] and [PROG].
	Parameterisation example:
	`wz[5].kinematic.param[0] 1538000 #Tool offset 1: 153.8 mm`
	`wz[5].kinematic.param[1] 25000 #Tool offset 2: 2.5 mm`
	`wz[5].kinematic.param[2] 0 #Tool offset 3: 0 mm`
	`wz[5].kinematic.param[5] 900000 #Tool offset 6: 90 mm`

6.2.1.11 Tool fixed / alignable (P-TOOL-00010)

P-TOOL-00010	Tool fixed / alignable
Description	This parameter is assigned the value TRUE if the tool orientation relative to the machine axis coordinate system and in combination with the 'Machining Coordinate System' [PROG] functionality cannot and should not be changed (e.g. because of non existent rotary axes). In this case, tool axis offsets (P-TOOL-00006 [▶ 110]) as components of the tool offset vector always relate to the machine axes.
Parameter	wz[i].tool_fixed
Data type	BOOLEAN
Data range	0/1
Dimension	----
Default value	0
Remarks	Parameterisation example: This parameter indicates whether the measuring probe is aligned in parallel to the axes of the rotated coordinate system for measuring in the machining coordinate system *wz[5].tool_fixed 1 #Tool fixed*

6.2.1.12 Kinematic ID (P-TOOL-00011)

P-TOOL-00011	Kinematic ID
Description	The purpose of kinematic ID is to identify the machine or tool head specific kinematic types implemented in the controller. This parameter defines the kinematic transformation to be used for the tool. When it is assigned the value 0, the kinematic ID results from the default setting as described in P-CHAN-00032 or from the programming in the NC program (#KIN ID [...] or #TRAFO [...]). The kinematic ID always refers to a transformation which must be defined in the channel parameters in a data record of the first kinematic step (kin_step[0].trafo[*].id or trafo[*].id).
Parameter	wz[i].kin_id
Data type	UNS16
Data range	0 ≤ kin_id ≤ MAX(UNS16)
Dimension	----
Default value	0
Remarks	

6.2.1.13 Dynamic and gear data

Depending on the application, it may be necessary to define specific dynamic data for some tools. This data is used in the spindle to limit speed and acceleration during machining. The logical axis number of the spindle must also be specified in order to transmit the dynamic data to the spindle with the assigned tool.

6.2.1.13.1 Extended tool settings (P-TOOL-00147)

P-TOOL-00147	Extended tool settings (modes)
Description	This parameter enables additional modes for the tool. The following modes are currently available: • **NONE**: no additional modes (default) • **GEAR_CHANGE_NO_STOP**: All the functions of the parameter P-TOOL-00019 [▶ 118] can be used when this mode is set. It permits a change in the rotation direction of M3/M4 or M4/M3 when this mode is set. In addition to P-TOOL-00019 [▶ 118] this mode can also differentiate between the gear ratios of the two tools. An “internal” gear change then takes place without the spindle stopping. However, the mode must be set for all the tools participating in the change. **Note:** Only use this parameter if no physical tool change is linked to the D/T word (e.g. several drills/milling cutters on one machine).
Parameter	wz[i].additional_settings
Data type	STRING
Data range	NONE, GEAR_CHANGE_NO_STOP
Unit	----
Default value	NONE
Remarks	Parameter available as of V3.01.3068.06

This parameter may be used for a tool system, for example, where the same spindle drives several drills simultaneously.

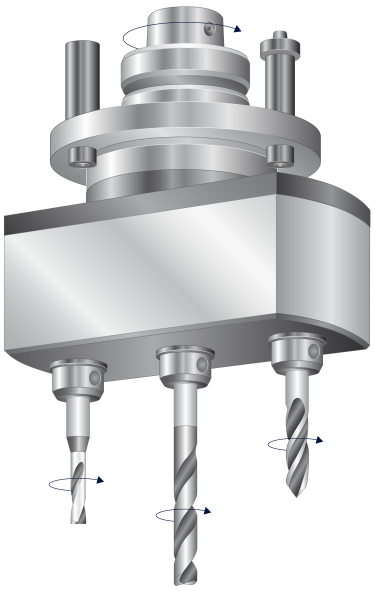


Fig. 3: Tool system with several drills

6.2.1.13.2 Logical spindle axis number (P-TOOL-00012)

P-TOOL-00012	Logical spindle axis number
Description	The dynamic data of the tool is assigned by the logical axis number of the spindle which will carry the tool after it is changed. If the logical axis number is assigned the value 0, no dynamic data is transferred to the spindle when the tool is changed. Dynamic data is modal in the spindle. It can either be overwritten • by the dynamic data of a new tool or • be reset by S[DEFAUL_DYNAMIK_DATA] to the default values configured.
Parameter	wz[i].log_ax_nr_spdl
Data type	UNS16
Data range	0 ≤ log_ax_nr_spdl ≤ MAX(UNS16)
Dimension	----
Default value	0
Remarks	Parameterisation example: Assigning the dynamic data of Tool 5 *wz[5].log_ax_nr_spdl 6 #Logical axis number of the spindle*

6.2.1.13.3 Minimum rotation speed (P-TOOL-00013)

P-TOOL-00013	Minimum rotation speed
Description	Minimum tool rotation speed.
Parameter	wz[i].vb_min
Data type	REAL64
Data range	0 ≤ vb_min ≤ P-TOOL-00014 [▶ 116]
Unit	0.001°/s
Default value	0
Remarks	The minimum rotation speed is only limited for endless turning. Parameterisation example: Assigning the dynamic data of Tool 5 *wz[5].vb_min 60000 #Minimum rotation speed*

6.2.1.13.4 Maximum rotation speed (P-TOOL-00014)

P-TOOL-00014	Maximum rotation speed
Description	Maximum tool rotation speed.
Parameter	wz[i].vb_max
Data type	REAL64
Data range	1 ≤ vb_max ≤ 2000000000
Unit	0.001°/s
Default value	1
Remarks	Parameterisation example: Assigning the dynamic data of Tool 5 *wz[5].vb_max 3000000 #Maximum rotation speed*

6.2.1.13.5 Maximum acceleration (P-TOOL-00015)

P-TOOL-00015	Maximum acceleration
Description	Maximum tool acceleration.
Parameter	wz[i].a_max
Data type	REAL64
Data range	1 ≤ a_max ≤ 100000000
Unit	°/s2
Default value	1
Remarks	Parameterisation example: Assigning the dynamic data of Tool 5 *wz[5].a_max 3000 #Maximum acceleration*

6.2.1.13.6 Numerator gear ratio of a tool (P-TOOL-00016)

P-TOOL-00016	Gear ratio of a tool (numerator)
Description	If the tool has a gear, this parameter specifies the gear ratio. A gear change occurs in the position controller after the tool is changed. The gear ratio need not then be considered when programming the tool rotary speed.
Parameter	wz[i].gear_ratio_num
Data type	SGN32
Data range	0 ≤ gear_ratio_num ≤ MAX(SGN32)
Dimension	----
Default value	0
Remarks	It is only permitted to change a tool with a gear speed when the spindle is at standstill (P-TOOL-00012 [▶ 115]) If the numerator or the denominator is assigned the value 0, no gear change takes place.

6.2.1.13.7 Denominator gear ratio of tool (P-TOOL-00017)

P-TOOL-00017	Gear ratio of a tool (denominator)
Description	If the tool has a gear, this parameter specifies the gear ratio. A gear change occurs in the position controller after the tool is changed. The gear ratio need not then be considered when programming the tool rotary speed.
Parameter	wz[i].gear_ratio_denom
Data type	SGN32
Data range	0 ≤ gear_ratio_denom ≤ MAX(SGN32)
Dimension	----
Default value	0
Remarks	It is only permitted to change a tool with a gear speed when the spindle is at standstill (P-TOOL-00012 [▶ 115]) If the numerator or the denominator is assigned the value 0, no gear change takes place.

6.2.1.13.8 Reversal of rotation direction by gear (P-TOOL-00018)

P-TOOL-00018	Reversal of rotation direction by gear
Description	If the direction of rotation is reversed by the tool gear, a change in the rotation direction can be defined in the position controller by setting this parameter to the value TRUE. This is executed as soon as a tool change is carried out.
Parameter	wz[i].gear_inv_direction
Data type	BOOLEAN
Data range	0/1
Dimension	----
Default value	0
Remarks	

6.2.1.13.9 Reversal of rotation direction without spindle standstill (P-TOOL-00019)

P-TOOL-00019	Reversal of rotation direction without spindle standstill
Description	When changing a tool with a tool gear which changes the direction of rotation (see P-TOOL-00018 [▶ 117]), the spindle must be at standstill, otherwise the CNC generates the error message P-ERR-60290. If this parameter is assigned the value TRUE, a change in tool rotation direction is only evaluated when a programmed change in spindle direction takes place (M03/M04 or M04/M03). In this case, the spindle does not stop. Example: This is practical for a machine with several tools rotating at the same time.
Parameter	wz[i].gear_inv_direction_no_stop
Data type	BOOLEAN
Data range	0/1
Dimension	----
Default value	0
Remarks	Only use this parameter if no physical tool change is linked to the D/T word (e.g. several drills/milling cutters on one machine).

6.2.1.14 Data for grinding applications

The following parameters are provided for grinding machining and the associated consideration and monitoring of tool wear (e.g. grinding disc).

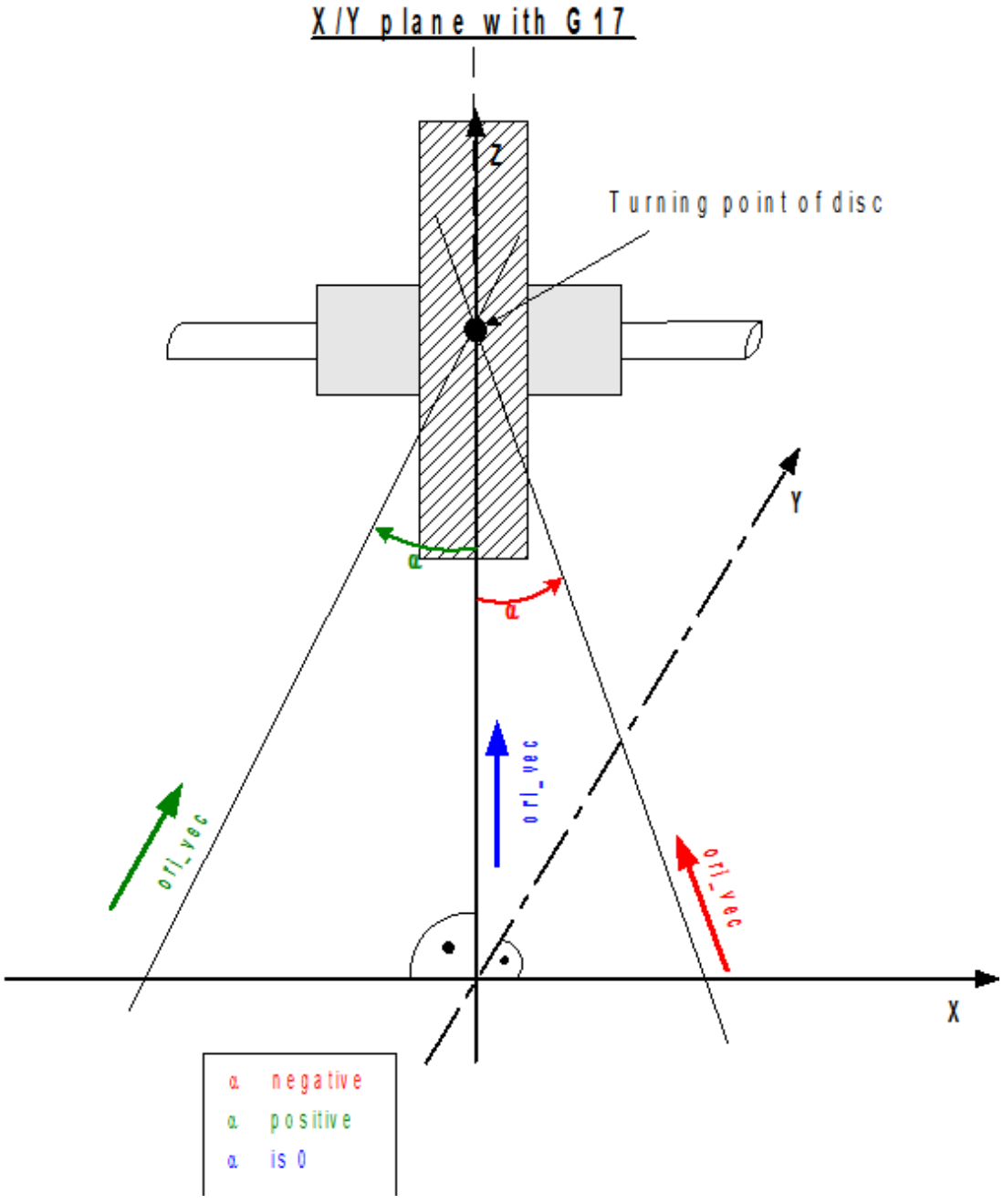
6.2.1.14.1 Wear constant (P-TOOL-00030)

P-TOOL-00030	**Wear constant**
Description	The wear constant is used to calculate continuous tool wear. Alternatively, it can also be defined in the NC program (#OTC […].
Parameter	wz[i].grinding_wear_const
Data type	REAL64
Data range	0 ≤ grinding_wear_const
Dimension	0.1µm/m
Default value	0
Remarks	The wear constant should contain relatively small values. There is no special dynamic consideration based on the actual wear values.

6.2.1.14.2 Maximum discrete infeed (P-TOOL-00031)

P-TOOL-00031	**Maximum discrete infeed**
Description	The maximum discrete infeed defines the greatest relative change which is assignable by the PLC.
Parameter	wz[i].grinding_max_infeed
Data type	REAL64
Data range	MIN(SGN32) ≤ grinding_max_infeed ≤ MAX(SGN32)
Dimension	0.1µm
Default value	0
Remarks	

6.2.1.14.3 Grinding disc tilt angle (P-TOOL-00138)

P-TOOL-00138	Grinding disc tilt angle
Description	The parameter defines the angle between the centre line of the grinding disc and the 3rd main axis for tilted grinding discs.
Parameter	wz[i].grinding_disc_tilt_angle
Data type	REAL64
Data range	-45°≤ grinding_tilt_angle ≤ 45°
Dimension	0.0001°
Default value	0
Remarks	 Grinding disc tilt angle

6.2.1.15 Data for visualisation and collision monitoring

Tools and tool heads are displayed as graphical objects as part of machining simulation. An additional member of the kinematic chain, what is known as a linkpoint, is appended to the connected graphical tool objects for the purpose of visualisation.

The specific data required is transferred in the tool data at tool change and sent to the machining simulation by appropriate commands (#SCENE ..., V.G.WZ_AKT.LINKPOINT...@@[PROG]) in the NC program.

Release Note

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

6.2.1.15.1 Linkpoint data (wz[i].linkpoint_data.*)

Linkpoint data (LINKPOINT) of the kinematic chain to which graphical tool objects are attached is entered in this structure.

6.2.1.15.1.1 Linkpoint name (P-TOOL-00100)

P-TOOL-00100	**Name of linkpoint**
Description	Name of the new linkpoint of the kinematic chain.
Parameter	wz[i].linkpoint_data.name
Data type	STRING
Data range	Maximum 40 characters (application-specific)
Unit	----
Default value	-
Remarks	

6.2.1.15.1.2 Mountpoint (P-TOOL-00101)

P-TOOL-00101	**Mountpoint**
Description	Name of the existing node of the kinematic chain to which the new node is attached.
Parameter	wz[i].linkpoint_data.mountpoint
Data type	STRING
Data range	Maximum 40 characters (application-specific)
Dimension	----
Default value	-
Remarks	

6.2.1.15.1.3 Translation of main axes (P-TOOL-00102)

P-TOOL-00102	Translation of main axes
Description	Shift of the link point in the main axes relative to the mountpoint.
Parameter	wz[i].linkpoint_data.translation[k] where k=0, 1, 2 (application-specific)
Data type	SGN32
Data range	MIN(SGN32) < translation[k] < MAX(SGN32)
Dimension	0.1µm or 0.0001 inch
Default value	0
Remarks	

6.2.1.15.1.4 Rotation of main axes (P-TOOL-00103)

P-TOOL-00103	Rotation of main axes
Description	Eulerian angle by means of which current rotation of the link point is specified with respect to the mountpoint.
Parameter	wz[i].linkpoint_data.rotation[k] where k=0, 1, 2 (application-specific)
Data type	SGN32
Data range	-360° < rotation[k] < 360°
Dimension	0.001°
Default value	0
Remarks	

6.2.1.15.1.5 Axis number (P-TOOL-00104)

P-TOOL-00104	Axis number
Description	Logical axis number that leads to movement of the new linkpoint.
Parameter	wz[i].linkpoint_data.ax_nr
Data type	UNS16
Data range	0 < ax_nr ≤ MAX(UNS16)
Dimension	----
Default value	0
Remarks	

6.2.1.15.1.6 Translation/rotation of linkpoint (P-TOOL-00105)

P-TOOL-00105	Translation/rotation of linkpoint
Description	Definition whether a movement of the axis leads to a translation in X / Y / Z or a rotation about X / Y / Z: 1-TRANS_X, 2-TRANS_Y, ..., 6-ROT_Z.
Parameter	wz[i].linkpoint_data.trans_rot
Data type	SGN32
Data range	MIN(SGN32) < trans_rot < MAX(SGN32)
Dimension	----
Default value	0
Remarks	

6.2.1.15.1.7 Effect of the movement on the linkpoint (P-TOOL-00106)

P-TOOL-00106	Effect of the movement on the linkpoint
Description	This parameter defines whether the programmed axis movements are to exert an inverting effect on the translation or rotation of the linkpoint.
Parameter	wz[i].linkpoint_data.inverse
Data type	BOOLEAN
Data range	0: Axis movement has no inverting effect on the linkpoint 1: Axis movement has an inverting effect on the linkpoint
Dimension	----
Default value	0
Remarks	

6.2.1.15.1.8 Use of the linkpoint (P-TOOL-00107)

P-TOOL-00107	Use of the linkpoint
Description	This parameter defines the method for handling the linkpoint during visualisation.
Parameter	wz[i].linkpoint_data.visible
Data type	BOOLEAN
Data range	0: Linkpoints are only for internal calculation of the chain. This means that the linkpoint is transferred once when added but it is not cyclically displayed. 1: Movements of the linkpoint are cyclically displayed when logged (e.g. via Data Factory).
Dimension	----
Default value	0
Remarks	

6.2.1.15.1.9 Position of linkpoint after axis exchange (P-TOOL-00108)

P-TOOL-00108	Position of linkpoint after axis exchange
Description	This parameter defines the method for handling a linkpoint attached to an axis during collision consideration if this axis is no longer available in the NC channel due to axis exchange.
Parameter	wz[i].linkpoint_data.fixed
Data type	BOOLEAN
Data range	0: Collision monitoring for this linkpoint can no longer be reliably executed. 1: Linkpoint position remains constant after axis exchange. Collision monitoring can be executed.
Dimension	----
Default value	0
Remarks	

6.2.1.15.1.10 Maximum lever arm length (P-TOOL-00109)

P-TOOL-00109	Maximum lever arm length
Description	Maximum lever arm length of the graphical objects attached to the (rotary) linkpoint. This is a segmentation aid for rotary axes.
Parameter	wz[i].linkpoint_data.arm_len
Data type	SGN32
Data range	MIN(SGN32) < arm_len < MAX(SGN32)
Dimension	0.1µm or 0.0001 inch
Default value	0
Remarks	

6.2.1.15.2 Object data (wz[i].gobject_data[j].*)

In this structure, enter data of graphical objects which are linked to the linkpoint (LINKPOINT). Application-specific objects <j> can be defined.

Structure name	Index
gobject_data[j]	j = 0 to 4 (Maximum number of graphical elements: 5, application-specific)

6.2.1.15.2.1 Name Name of graphical object (P-TOOL-00120)

P-TOOL-00120	Name of graphical object
Description	Name of the graphical object.
Parameter	wz[i].gobject_data[j].name
Data type	STRING
Data range	Maximum 40 characters (application-specific)
Unit	----
Default value	-
Remarks	

6.2.1.15.2.2 Linkpoint name (P-TOOL-00121)

P-TOOL-00121	Linkpoint name
Description	Name of the node (LINKPOINT) in the kinematics chain to which the graphical object is appended.
Parameter	wz[i].gobject_data[j].linkpoint
Data type	STRING
Data range	Maximum 40 characters (application-specific)
Dimension	----
Default value	-
Remarks	

6.2.1.15.2.3 Group name (P-TOOL-00122)

P-TOOL-00122	Group name
Description	Name of the group (summarised graphical objects) which is additionally attached to the new graphical object.
Parameter	wz[i].gobject_data[j].group[k] where k=0, 1, 2, 3, 4 (application-specific)
Data type	STRING
Data range	Maximum 40 characters (application-specific)
Dimension	----
Default value	-
Remarks	

6.2.1.15.2.4 Translation of object (P-TOOL-00123)

P-TOOL-00123	Translation of object
Description	Shift of the graphical object or of the group with respect to the linkpoint (LINKPOINT) if specified absolutely or with respect to the previous coordinates if specified as relative (P-TOOL-00125 [▶ 126]).
Parameter	wz[i].gobject_data[j].translation[k] where k=0, 1, 2 (application-specific)
Data type	SGN32
Data range	MIN(SGN32) < translation[k] < MAX(SGN32)
Dimension	0.1µm or 0.0001 inch
Default value	0
Remarks	

6.2.1.15.2.5 Rotation of object (P-TOOL-00124)

P-TOOL-00124	Rotation of object
Description	Eulerian angle by which the static rotation of the graphical object is specified relative to the node point (LINKPOINT).
Parameter	wz[i].gobject_data[j].rotation[k] where k=0, 1, 2 (application-specific)
Data type	SGN32
Data range	-360° < rotation[k] < 360°
Dimension	0.001°
Default value	0
Remarks	

6.2.1.15.2.6 Effect of shift and rotation (P-TOOL-00125)

P-TOOL-00125	Effect of shift and rotation
Description	This parameter defines how the programmed shifts and rotations are to affect the current shifts and rotations.
Parameter	wz[i].gobject_data[j].relative
Data type	BOOLEAN
Data range	0: The specified shift/rotation replaces all previously activated shifts/rotations. 1: Shift and rotation have an accumulative effect on the current shift or rotation.
Dimension	----
Default value	0
Remarks	

6.2.1.15.2.7 File containing description of object data (P-TOOL-00126)

P-TOOL-00126	File containing description of object data
Description	Name of the file which additionally describes the graphical object (e.g. VRML file).
Parameter	wz[i].gobject_data[j].file
Data type	STRING
Data range	Maximum 40 characters (application-specific)
Dimension	----
Default value	-
Remarks	

6.2.1.15.2.8 Key name (P-TOOL-00127)

P-TOOL-00127	Key name
Description	Key name of a key/value pair (value name: P-TOOL-00128 [▶ 127]) which additionally describes the graphical object.
Parameter	wz[i].gobject_data[j].key[k] where k=0, 1, 2, 3, 4 (number of key/value pairs per tool: 5, application-specific)
Data type	STRING
Data range	Maximum 40 characters (application-specific)
Dimension	----
Default value	-
Remarks	

6.2.1.15.2.9 Value name (P-TOOL-00128)

P-TOOL-00128	Value name
Description	Value name of a key/value pair (key name: P-TOOL-00127 [▶ 127]) which additionally describes the graphical object.
Parameter	wz[i].gobject_data[j].value[k] where k=0, 1, 2, 3, 4 (number of key/value pairs per tool: 5, application-specific)
Data type	STRING
Data range	Maximum 40 characters (application-specific)
Dimension	----
Default value	-
Remarks	

6.2.1.15.2.10 Information about a change (P-TOOL-00129)

P-TOOL-00129	Information about a change
Description	This parameter informs about a data change by the external tool management system. The user decides whether to evaluate this information or not.
Parameter	wz[i].gobject_data[j].changed
Data type	BOOLEAN
Data range	0: No change of data 1: Data change by the external tool management system.
Dimension	----
Default value	0
Remarks	

6.2.1.16 Path-specific tool data (wz[i].path[j].*)

In systems with so-called multipath programming, path-specific tool data can be defined in the structure "wz[i].path[j]".

Structure name	Index
path[j]	j = 0 (Path 1), 1 (Path 2)

6.2.1.16.1 Tool radius (P-TOOL-00020)

P-TOOL-00020	Tool radius
Description	Enter the tool radius for each path in this parameter
Parameter	wz[i].path[j].radius
Data type	SGN32
Data range	MIN(SGN32) ≤ radius < MAX(SGN32)
Dimension	0.1 µm or 0.0001 inch
Default value	0
Remarks	Parameterisation example: The tool radius of Tool 5 for Path 1 is to be 10 mm and for Path 2 10.5 mm. *wz[5].path[0].radius 100000 # Tool radius 10 mm in Path 1* *wz[5].path[1].radius 105000 # Tool radius 10.5 mm in Path 2*

6.2.1.17 Settings for the free configuration of the tool number

If the parameter P-TOOL-00140 [▶ 130] is set to 1, any numbers can be entered for the tool in the structure "wz[i].tool_id.*".

Structure name	Meaning
tool_id.*	Structure for any numbers for the basic, sister and variant tools

6.2.1.17.1 Enabling tool number assignment (P-TOOL-00140)

P-TOOL-00140	Enabling tool number assignment
Description	The basic setting defines the tool number <i> by the index of the tool structure wz[<i>]. As a result, the structure size determines the maximum possible number of tools and their numbers. For example, if the structure wz[<i>] has 200 places, the tool number range is from 0 to 199. If P-TOOL-00140 is set to 1, any number can be entered in the structure wz[i].tool_id.* [▶ 129] independent of the index numbering for each tool.
Parameter	tool_data_with_id
Data type	BOOLEAN
Data range	0: Determine the tool number by the index number 1: Any definition of the tool number from the tool ID
Dimension	----
Default value	0
Remarks	The index of wz[i] only determines the number of possible tools when the parameter P-TOOL-00140 is set.

6.2.1.17.2 Defining the tool number(/tool ID (wz[i].tool_id.*)

6.2.1.17.2.1 Basic tool number (P-TOOL-00141)

P-TOOL-00141	Basic tool number
Description	Enter the basic tool number in this parameter.
Parameter	wz[i].tool_id.basic
Data type	SGN32
Data range	MIN(SGN32) < P-TOOL-00141< MAX(SGN32)
Dimension	----
Default value	0
Remarks	

6.2.1.17.2.2 Sister tool number (P-TOOL-00142)

P-TOOL-00142	Sister tool number
Description	Enter the sister tool number in this parameter.
Parameter	wz[i].tool_id.sister
Data type	SGN32
Data range	MIN(SGN32) < P-TOOL-00142 < MAX(SGN32)
Dimension	----
Default value	0
Remarks	

6.2.1.17.2.3 Variant tool number (P-TOOL-00143)

P-TOOL-00143	Variant tool number
Description	Enter the variant tool number in this parameter.
Parameter	wz[i].tool_id.variant
Data type	SGN32
Data range	MIN(SGN32) < P-TOOL-00143 < MAX(SGN32)
Dimension	----
Default value	0
Remarks	

6.2.1.17.2.4 Validity flag of the sister tool (P-TOOL-00144)

P-TOOL-00144	Validity flag of the sister tool
Description	This parameter indicates whether the sister tool P-TOOL-00142 [▶ 131] is valid.
Parameter	wz[i].tool_id.sister_valid
Data type	BOOLEAN
Data range	0 / 1
Dimension	----
Default value	0
Remarks	

6.2.1.17.2.5 Validity flag of the variant tool (P-TOOL-00145)

P-TOOL-00145	**Validity flag of the variant tool**
Description	This parameter indicates whether the variant tool is valid. It is specified by P-TOOL-00143 [▶ 131]
Parameter	wz[i].tool_id.variant_valid
Data type	BOOLEAN
Data range	0 / 1
Dimension	----
Default value	0
Remarks	

6.3 Example of assigning tool data

```
# ************************************************************************
# Tool data
# ************************************************************************
# Important note : Behind the comment character ´#´
# a blank (space) must be added
# CAUTION: Tool dimensions are expected in the unit
# 0.1 µm or 0.0001 inch.
# ************************************************************************
# ========================================================================
# Tool data for tool no. 5
# ========================================================================
Wz[5].laenge 6000 # Tool length
wz[5].radius 54000 # Tool radius
wz[5].gueltig 1 # Tool validity flag TRUE
wz[5].mass_einheit 0 # Tool dimensional unit MM
wz[5].ax_versatz[0] 205000 # Offset 1. Axis
wz[5].ax_versatz[1] 206000 # Offset 2. Axis
wz[5].ax_versatz[2] 307000 # Offset 3. Axis
wz[5].ax_versatz[3] 408000 # Offset 4. Axis
wz[5].log_ax_nr_spdl 6 # Logical axis number of spindle
wz[5].vb_min 60000 # Minimum rotation speed
wz[5].vb_max 3000000 # Maximum rotation speed
wz[5].a_max 3000 # Maximum acceleration
# ========================================================================
# Tool data for tool no. 8
# ========================================================================
Wz[8].laenge 8000 # Tool length
wz[8].radius 45000 # Tool radius
wz[8].gueltig 1 # Tool validity flag TRUE
wz[8].mass_einheit 0 # Tool dimensional unit MM
wz[8].ax_versatz[0] 225000 # Offset 1. Axis
wz[8].ax_versatz[1] -336000 # Offset 2. Axis
wz[8].ax_versatz[2] -457000 # Offset 3. Axis
wz[8].ax_versatz[3] 578000 # Offset 4. Axis
wz[8].kinematic.param[0] 1538000 # Tool offset 1: 153.8 mm
wz[8].kinematic.param[1] 25000 # Tool offset 2: 2.5 mm
wz[8].kinematic.param[2] 0 # Tool offset 3: 0 mm
wz[8].kinematic.param[5] 900000 # Tool offset 6; 90 mm
wz[8].tool_fixed 1 # Tool fixed
wz[8].kin_id 6 # Kinematic 6
# ========================================================================
# Tool data for tool no. 15
# ========================================================================
wz[15].typ 1 # Turning tool
wz[15].srk_lage 5 # Cutter position
wz[15].laenge 8250 # Tool length
wz[15].radius 200 # Tool radius
wz[15].gueltig 1 # Tool validity flag TRUE
wz[15].mass_einheit 0 # Tool dimensional unit MM
wz[15].ax_versatz[0] 0 # Offset 1. Axis
wz[15].ax_versatz[1] 0 # Offset 2. Axis
wz[15].ax_versatz[2] 0 # Offset 3. Axis
wz[15].ax_versatz[3] 0 # Offset 4. Axis
```

```
# ========================================================================
# Tool data for tool no. 23
# ========================================================================
wz[23].laenge 5000 # Tool length
wz[23].radius 10000 # Tool radius
wz[23].gueltig 1 # Tool validity flag TRUE
wz[23].mass_einheit 0 # Tool dimensional unit MM
wz[23].ax_versatz[0] 565000 # Offset 1. Axis
wz[23].ax_versatz[1] 236000 # Offset 2. Axis
wz[23].ax_versatz[2] -233000 # Offset 3. Axis
wz[23].ax_versatz[3] 566400 # Offset 4. Axis
#
# Example of GOBJECT description in internal tool data base
#
wz[1].gobject[0].name GO_NAME
wz[1].gobject[0].linkpoint GO_LINKPOINT
wz[1].gobject[0].group[0] GO_GROUP
wz[1].gobject[0].group[1] GO_GROUP_1
wz[1].gobject[0].group[2] GO_GROUP_2
wz[1].gobject[0].group[3] GO_GROUP_3
wz[1].gobject[0].group[4] GO_GROUP_4
wz[1].gobject[0].translation[0] 10000 # integer in [0.1 um]
wz[1].gobject[0].translation[1] 20000
wz[1].gobject[0].translation[2] 30000
wz[1].gobject[0].rotation[0] 300000 # integer in [0.0001 degree]
wz[1].gobject[0].rotation[1] 600000
wz[1].gobject[0].rotation[2] 900000
wz[1].gobject[0].relative 1
wz[1].gobject[0].changed 1
wz[1].gobject[0].file GO_FILE
wz[1].gobject[0].key[0] GO_KEY
wz[1].gobject[0].key[1] GO_KEY_1
wz[1].gobject[0].key[2] GO_KEY_2
wz[1].gobject[0].key[3] GO_KEY_3
wz[1].gobject[0].key[4] GO_KEY_4
wz[1].gobject[0].value[0] GO_VALUE
wz[1].gobject[0].value[1] GO_VALUE_1
wz[1].gobject[0].value[2] GO_VALUE_2
wz[1].gobject[0].value[3] GO_VALUE_3
wz[1].gobject[0].value[4] GO_VALUE_4
#
wz[1].gobject[1].name GO_1_NAME
wz[1].gobject[1].file GO_1_FILE
:
wz[1].gobject[2].name GO_2_NAME
wz[1].gobject[2].file GO_2_FILE
:
wz[1].gobject[3].name GO_3_NAME
wz[1].gobject[3].file GO_3_FILE
:
wz[1].gobject[4].name GO_4_NAME
wz[1].gobject[4].file GO_4_FILE
```

```
#
# Example of LINKPOINT description in internal tool data base
#
wz[1].linkpoint.name LP_NAME
wz[1].linkpoint.mountpoint LP_MOUNTPOINT
wz[1].linkpoint.translation[0] 1
wz[1].linkpoint.translation[1] 2
wz[1].linkpoint.translation[2] 3
wz[1].linkpoint.rotation[0] 30
wz[1].linkpoint.rotation[1] 60
wz[1].linkpoint.rotation[2] 90
wz[1].linkpoint.ax_nr 1
wz[1].linkpoint.trans_rot 1
wz[1].linkpoint.inverse 0
wz[1].linkpoint.visible 1
wz[1].linkpoint.fixed 0
wz[1].linkpoint.arm_len 1234
#
End
```

6.4 Appendix

6.4.1 Discontinued parameters

6.4.1.1 Name of the assigned partial kinematic (P-TOOL-00148)

P-TOOL-00148	**Name of the assigned partial kinematic**
Description	This parameter notes the name of the assigned partial kinematic. The name specified here is compared with the names of the partial kinematics (P-CHAN-00443) when coupling kinematics are used and a search is made for the corresponding partial kinematic. The parameter data configured in this tool is then transferred to the correct partial kinematic.
Parameter	wz[i].kin_name
Data type	STRING
Data range	
Dimension	----
Default value	*
Remarks	Parameterisation example: *wz[1].hin_name ROBOT* * Note: The default value of variables is a blank string. The parameter is available as of V3.1.3080

6.4.2 References

[CHAN] Documentation of channel parameters

[PROG] CNC programming manual

6.4.3 Suggestions, corrections and the latest documentation

Did you find any errors? Do you have any suggestions or constructive criticism? Then please contact us at documentation@isg-stuttgart.de. The latest documentation is posted in our Online Help (DE/EN):

QR code link: https://www.isg-stuttgart.de/documentation-kernel/

The link above forwards you to:

https://www.isg-stuttgart.de/fileadmin/kernel/kernel-html/index.html

Notice

Change options for favourite links in your browser;

Technical changes to the website layout concerning folder paths or a change in the HTML framework and therefore the link structure cannot be excluded.

We recommend you to save the above "QR code link" as your primary favourite link.

PDFs for download:

DE:
https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

EN:
https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

E-Mail: documentation@isg-stuttgart.de

ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart

www.isg-stuttgart.de
support@isg-stuttgart.de