DOCUMENTATION ISG-kernel

Manual
HLI-Documentation from v2.2800

Short Description:
HLI

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

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

Documentation version: 1.21
13/11/2024

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 Introduction

Extensive data volumes are exchanged between the CNC and the PLC. For example, this comprises

- Commands from the NC program of the CNC to the PLC (e.g. technology commands, such as M, S, T and H commands etc.).
- Acknowledgements of technology commands by the PLC.
- Display data of the CNC (e.g. current axis positions, current and programmed path velocity etc.).
- Jobs from the PLC to the CNC (e.g. mode switchover, set feedhold etc.).
- Jobs from the GUI to the CNC which can be verified and, if applicable, denied by the PLC (e.g. mode switchover, set feedhold etc.).

The interface between CNC and PLC is outlined in the overview diagram below.

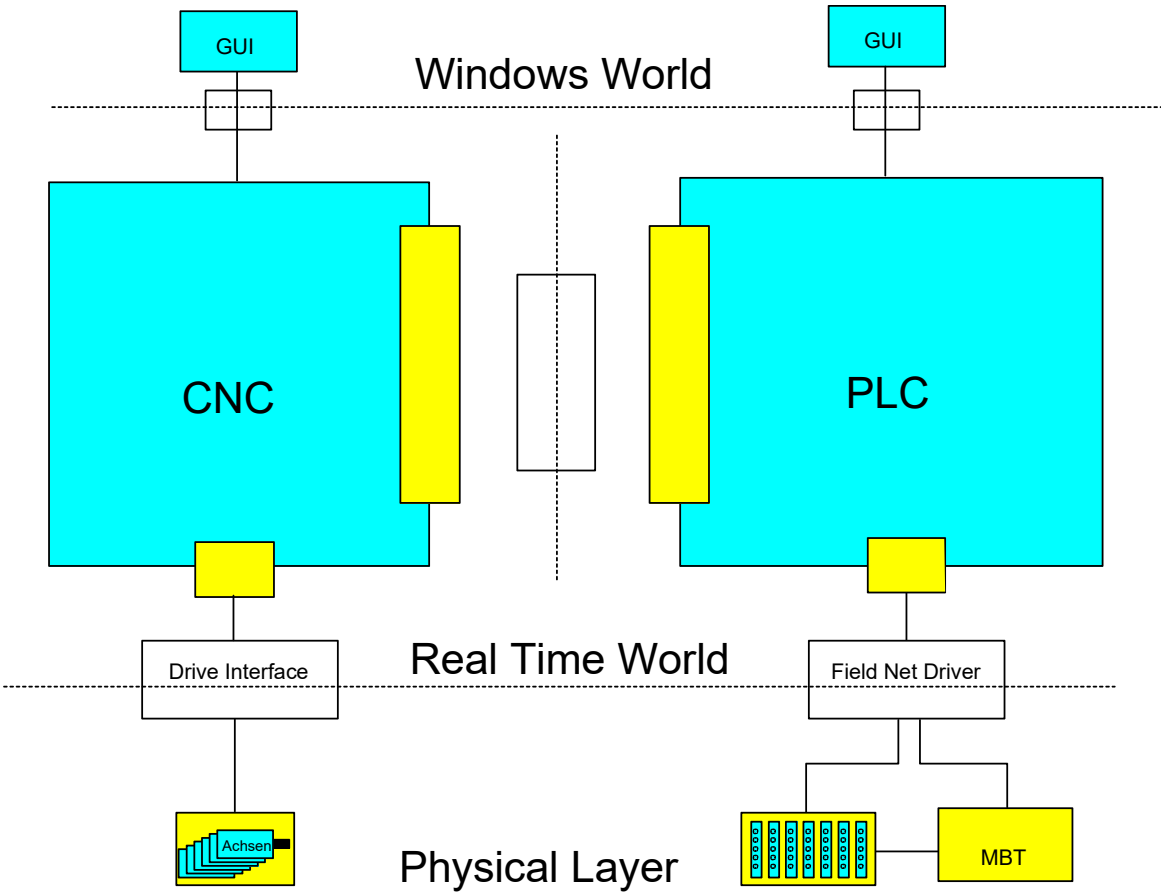


Fig. 1: CNC-PLC interface

This documentation describes the structure of this interface between CNC and PLC referred to below as High-Level Interface (HLI for short).

The term “High-Level Interface”, in this context, refers to the structure of the interface with complex data structures and handshake variables. This has to be seen in contrast to a data interface, which us considered to be at a low level of abstraction (Low-Level Interface).

Attention

Safety restriction for the use of the High Level Interface:

The values transferred between the PLC and TwinCAT CNC can be read and changed by all local users

In order to protect the system against unauthorised access, use the “IPC Security Guide“ as a guideline. This will support you in the management of security risks when using Beckhoff products.

Follow the links to the „IPC Security Guide“ for further measures:

German: https://infosys.beckhoff.com/content/1031/ipc_security/index.html

English: https://infosys.beckhoff.com/content/1033/ipc_security/index.html

1.1 Implementation by High-Level Interface (HLI)

A memory area is created as shared memory which both the CNC and the PLC can access in order to implement data exchange between CNC and PLC. This memory area is referred to as High-Level Interface (HLI).

In order to allow this data access, the relevant views from CNC and PLC onto the memory area must be identical. Data consistency is ensured by means of suitable data exchange mechanisms.

A library containing the structure of the HLI in structured text is provided for simple programming at the PLC end in IEC1131-3. This library is valid in each case for the current HLI format and the current status of the CNC and may change if a version update is implemented. Accordingly, the PLC must be recompiled and loaded if there is new library or if a new CNC is used.

1.2 Organisation of the HLI

As opposed to the usual simple structure of the data to be exchanged in input/output data with PLC applications, the HLI contains complex structures. They reflect the logical structuring of the CNC into channels, axes and platform data.

The logical structure of the HLI is shown in the diagram below:

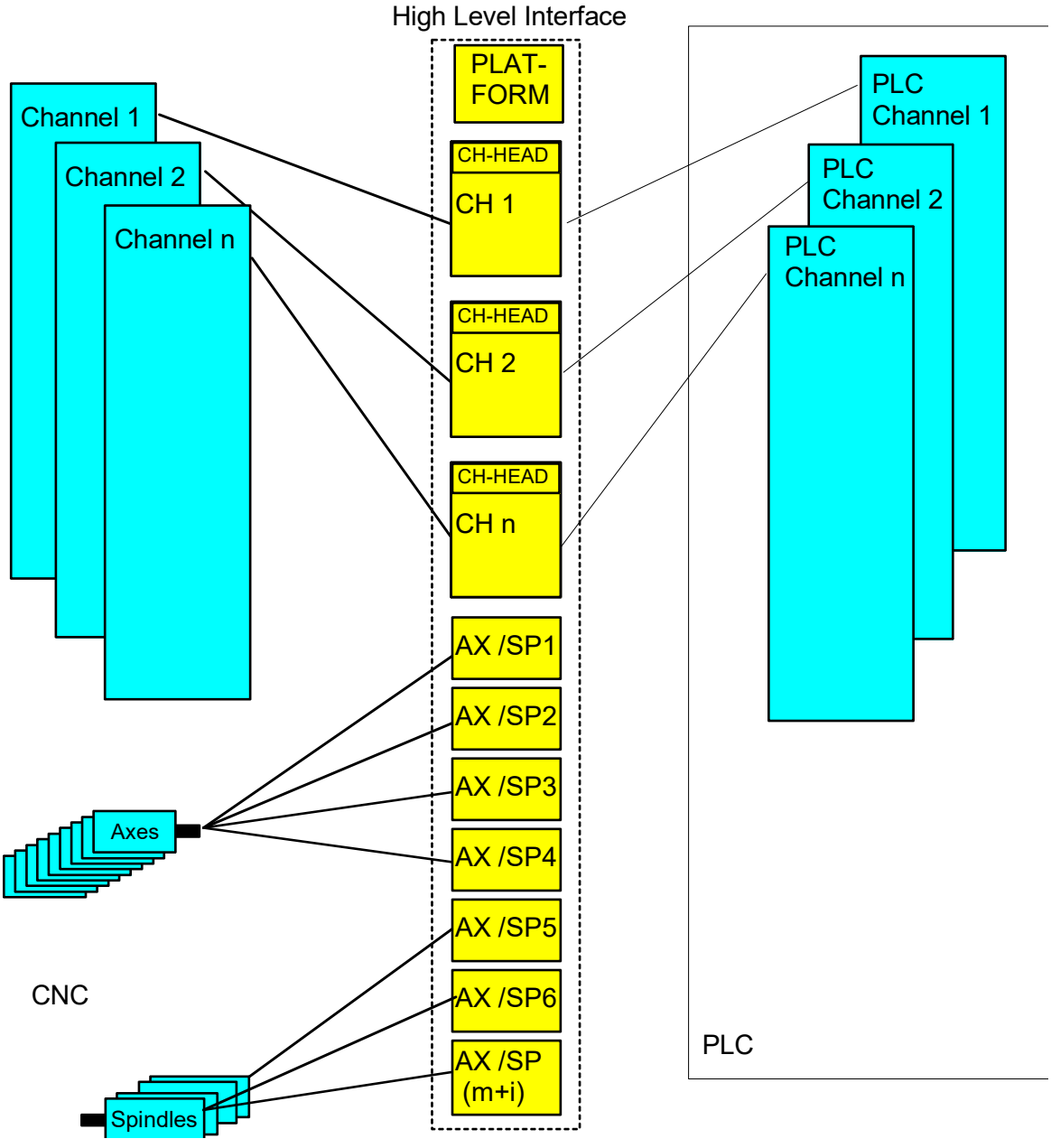


Fig. 2: Structuring of the HLI

The structuring into channel-specific and axis-specific data areas is clearly shown. The channel-specific and axis-specific memory areas on the HLI feature the same logical structure.

The relevant memory areas are subdivided into

- a header area with management data
- and a user data area with status information, control commands and technology data.

The channel/axis-specific memory areas are shown magnified in the diagram below:

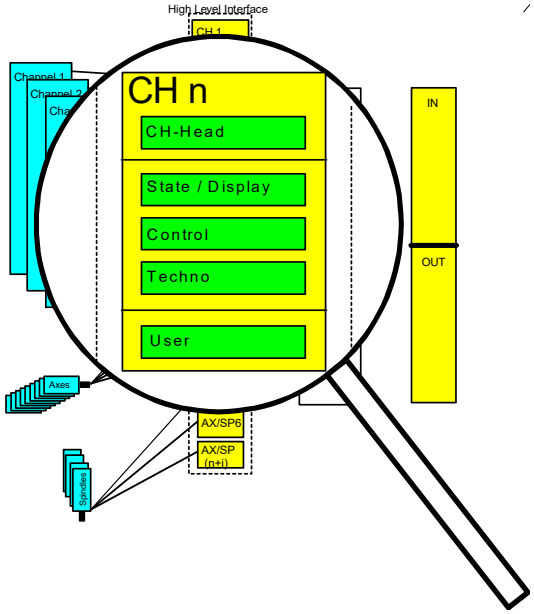


Fig. 3: Channel-specific memory area

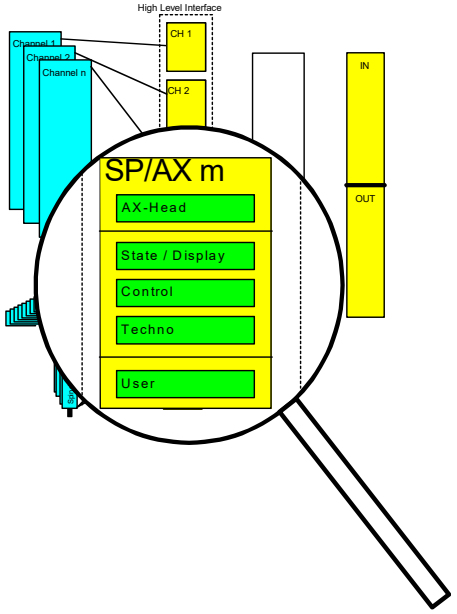


Fig. 4: Axis-specific/spindle-specific memory area

1.2.1 Data of a channel\axis

A distinction is made between the following data areas, whereby channels and axes may be considered the same:

The header areas contain:

- management data such as version information and log-on information

The user data areas contain:

- status and display information (CNC → PLC),
- control interfaces (PLC → CNC),
- technology areas (CNC → PLC and PLC → CNC),
- and any application-specific data.

Specific data, such as status information, is refreshed cyclically by the CNC and can be read by the PLC if needed. However, M functions, so-called usage information, must be read by the PLC. For this purpose, the interface contains suitable mechanisms to ensure that no data is lost and that the order of the data is retained.

1.3 Status and display information

The status information is transferred unidirectionally from the CNC to the PLC. This data is constantly refreshed by the CNC and can be read by the PLC if needed. The status information is updated without informing the PLC. Updating is not protected by a handshake protocol or semaphore mechanism.

With long PLC cycle times, the PLC does not record each short-term change under certain circumstances but is only informed of the current status.

Status information is exchanged on the HLI via memory. The transfer direction is defined for each status information item. The status data is transferred on the HLI as far as possible as single elementary data items (byte, Boolean, integer etc.).

The status information is combined depending on the CNC-internal structure to enhance structuring.

The diagram below shows the internal structure of the CNC in simplified form:

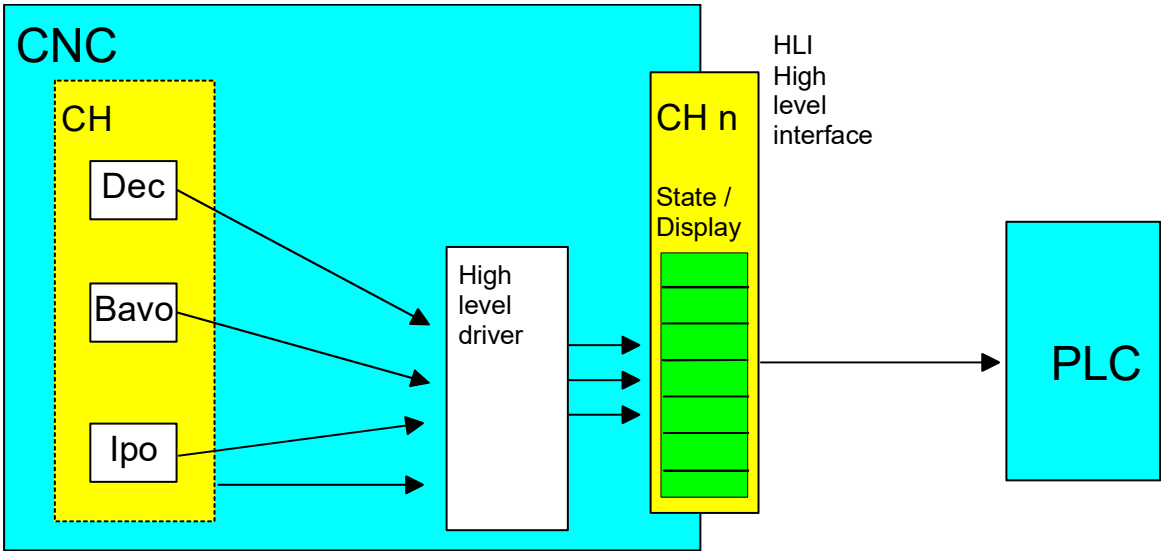


Fig. 5: Transfer of status information

The CNC is subdivided into the modules:

- **Dec** (Decoder): Module for decoding the NC program. This module converts the CNC program into an internal control data format and executes a number of different calculations, such as parameter calculation etc. Timewise, the decoder runs ahead of actual processing and under certain circumstances may be several hundred NC blocks ahead of current processing.
- **Bavo** (path preparation, look-ahead): Module to calculate dynamic limit values based on the set parameters. It detects corners and triggers geometry changes. The Bavo also runs asynchronously with current processing and under certain circumstances may be several hundred NC blocks ahead of current processing.
- **Ipo** (interpolator): Module to generate individual positions for the current cycle, allowing for programmed and maximum velocities etc. . The interpolator represents the current state of the machine.

1.4 Control commands

Control commands are transferred both from the CNC to the PLC and also in the opposite direction.

If a user interface (GUI) communicates via CNC communication objects, there is an option to redirect via the PLC each command which can be operated both by the GUI and the PLC. The PLC then makes the decision regarding the extent to which the GUI command may be transferred to the CNC.

A so-called control unit is created on the HLI for each control command. The control units are differentiated and named based on the target action.

- Control units to influence the CNC are referred to as MC control units (LC acts on MC).
- Control units to influence the PLC are referred to as LC control units (MC acts on LC).

The figure below clearly shows the implemented interaction between users of a control system by means of an MC control unit.

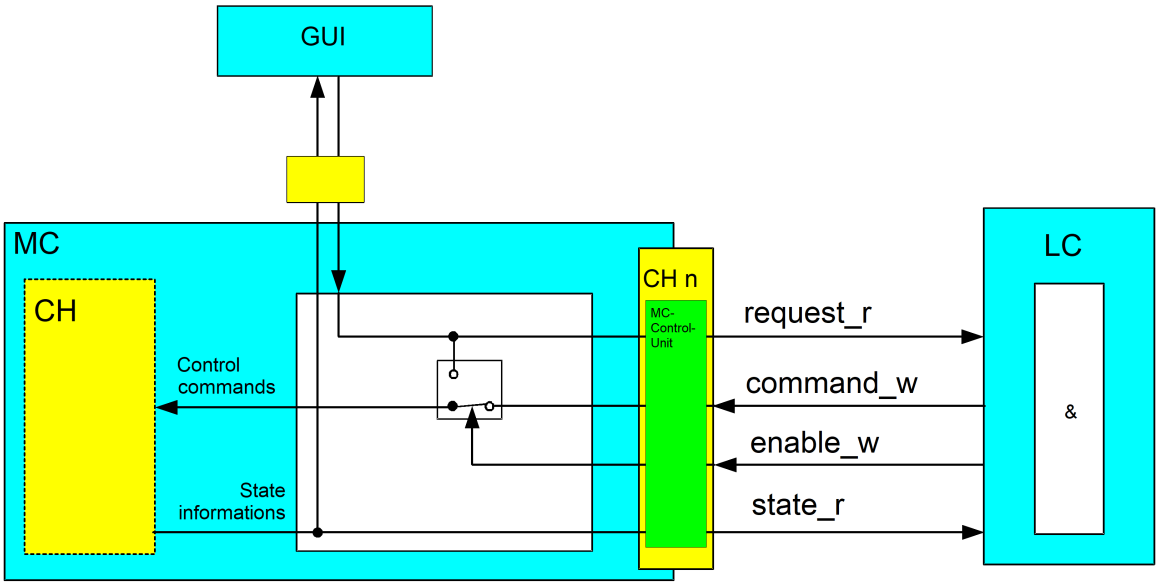


Fig. 6: Application of an MC control unit

If commands from the GUI are to be redirected via the PLC, the PLC must assign the **enable_w** element with the value TRUE. The corresponding control commands from the GUI are then set to the r**equest_r** element of the MC control unit before processing. This means that the PLC has the option of allowing or rejecting this request from the GUI. If a GUI command is allowed by the PLC, the PLC must copy the command from element **request_r** to element **command_w .**

Similarly, the PLC can directly send control commands to the CNC even without a prior job from a GUI by writing element **command_w**.

The element **state_r** is used to check the success of a command. The CNC saves the status corresponding to the command in this element.

An **MC control unit** has the following structure:

```
TYPE MC_CONTROL_UNIT:
STRUCT
   request_r      < DATENTYP A >;         (* data commanded by GUI *)
   enable_w       BOOL;                   (* PLC operates this command *)
   command_w      < DATENTYP A >;         (* data commanded by the PLC *)
   state_r        < DATENTYP B >;         (* CNC feedback *))
END_STRUCT
END_TYPE
```

Note:

- < DATA TYPE A > and < DATA TYPE B > may be identical.
- < DATA TYPE A > and < DATA TYPE B > may also be complex data structures besides the standard data types (e.g. BOOL, INT, UINT...).

1.4.1 Control unit with usage check

If the requested or commanded data is not applied statically and if an individual change should be transferred at this point, the data of the corresponding control unit is provided with an additional usage check.

```
TYPE MC_CONTROL_UNIT:
STRUCT
  request_r          :  < DATATYPE A >; (* data commanded by GUI *)
  enable_w           :  BOOL;             (* PLC operates this command *)
  command_w          :  < DATATYPE A >;(* data commanded by the PLC *)
  request_semaphor_rw :  BOOL;              (* Request valid *)
  command_semaphor_rw :  BOOL;              (* Command valid *)
  state_r            :  < DATATYPE B >; (* CNC feedback *))
END_STRUCT
END_TYPE
```

Example:

CNC accepts the commanded data if the command semaphore has the value TRUE and sets this element to the value FALSE after complete transfer of the data.

The PLC can write command data if the command semaphore has the value FALSE. Once all command data is written, the PLC sets this command semaphore to the value TRUE.

The CNC writes the data requested by the GUI if the request semaphore is FALSE and then sets this element to TRUE.

The PLC reads the data requested by the GUI if the request semaphore is TRUE. After the PLC fully accepts the data, the PLC sets this element to FALSE.

1.5 Operation with and without PLC

Machine start-up can also be performed with partially implemented PLC functionality. The CNC-side driver of the HLI therefore contains mechanisms for the internal simulation of interactions between PLC and CNC.

Example:

The axis should be run in with a rudimentary PLC. The PLC sets only the Axis Enables; other functions are not yet implemented in the PLC.

In order to switch operation modes or perform a reset anyway, the HLI driver simulates the required acknowledgements of the PLC so that the required action can still be performed.

In order to achieve this behaviour and still ensure reliable operation, the PLC issues a corresponding information item to the CNC for each function covered by the PLC, a so-called “present” marker. "Present" means that the PLC claims control of the interface and sends the required acknowledgements.

The PLC must then inform the CNC that it exists, once globally in the HLI. In addition all control commands are enabled individually.

This means that three scenarios are conceivable:

- CNC without PLC
- CNC with PLC which does not use all elements of the HLI
- CNC with PLC which uses all elements of the HLI

The housekeeping sections of each channel or each axis has this “present” element by which the PLC assumes responsibility for this interface.

In addition, each housekeeping unit of a control command, i.e. the control unit, contains an element which the PLC uses to inform the CNC that it supports this command. The **enable_w** element means at the same time that, if the corresponding command is issued via the GUI, it is routed via the PLC and reaches the CNC from there. The CNC can then no longer distinguish whether a command arrives from the PLC or the GUI.

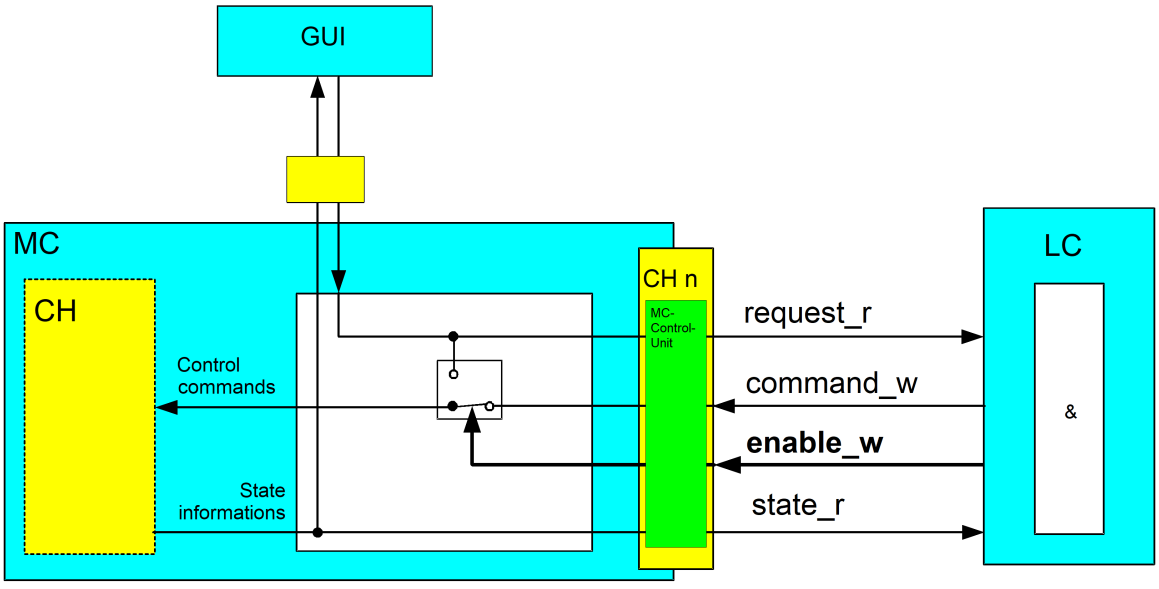


Fig. 7: Interaction, MC control unit and PLC

The connection of the GUI via the CNC is not absolutely necessary. Similarly, an interface can communicate directly with the PLC and the additionally provided communication channel GUI -> CNC -> PLC is not used in such cases.

2 Axes

2.1 Definition of axes

A distinction can be made between various types of axes in a control system. One practical classification is between programmable, logical and physical axes.

A **physical axis** corresponds to a real axis on the machine (translatory or rotary axis). In a control unit a physical axis is represented by software by means of a **logical axis** (1:1 mapping).

A logical axis is the unified representation of an axis in the axis coordinate system (acs). A logical axis in the control unit provides all the relevant information required for the related physical axis. From the point of view of the CNC, each logical axis has its own parameter set and its own interface to the PLC.

A **programmable axis** is an axis which can be programmed or commanded by the user in the part program or MDI mode. A programmable axis either corresponds directly to a logical/physical axis (1:1 mapping) or is mapped by kinematic transformation onto logical/physical axes (1:n mapping).

In multi-channel operation, a logical axis may be swapped between channels (“axis swap“). In this case, the same logical axis may occur in different attributes on the NC channels from the point of the view of the programmer, e.g. as a programmable spindle on one channel (speed preset, speed-controlled) and as a programmable C axis on another channel (position preset, position-controlled).

Programmable, logical and physical axes are generally identical (1: 1 mapping) for simple machines (e.g. a 3-axis milling machine with Cartesian linear axes X, Y and Z). However, with complex machine kinematics or industrial robots, kinematic transformation of programmed to logical/physical axes is required (1:n mapping).

Axes are programmed in the CNC program and moved by the CNC. Alternatively, specific motions may also be triggered directly by the PLC.

The PLC interface for axes has a similar structure as the interface for channels. M functions configured as axis-specific M functions also appear on the axis-specific interface and must be acknowledged by this interface. Similarly, this interface contains axis enables and status information for this axis.

2.2 Definition of coordinate systems

Allowance must be made for various coordinate systems owing to the structure of a machine and its machining programs. The totality of a machine is represented by the reference coordinate system in world coordinates. In turn, the individual axes themselves define coordinate systems, the workpiece and the tool.

The two different coordinate systems mainly used are described below.

Axis coordinate system acs

Each axis has its own coordinate system. Each axis is mounted either on the machine base or on another axis. This means that the machine base or the corresponding axis forms the basis. The axis coordinate system of an axis is fixed with respect to the mounting point of this axis.

Part program coordinate system pcs

This coordinate system is used in the Geometry Description with the DIN 66025 programming language. The data in a part program constitutes program coordinates. Exceptions are G functions which refer to direct axis coordinates.

Other coordinate system designations are listed for the purpose of completeness.

Machine coordinate system mcs

The machine coordinate system represents an abstract coordinate system. It is not bound to a fixed point of the machine. All other coordinate systems refer to this coordinate system.

Workpiece coordinate system wcs

This coordinate system is fixed to a fixed point of the workpiece. The workpiece description by coordinate information refers to this system.

Tool coordinate system tcs

The tool coordinate system has its origin at the clamping point of the tool. Tool information (geometry) refers to this system. Length compensation is therefore specified in tool coordinates. On Cartesian machines, the Z axis may coincide with length compensation.

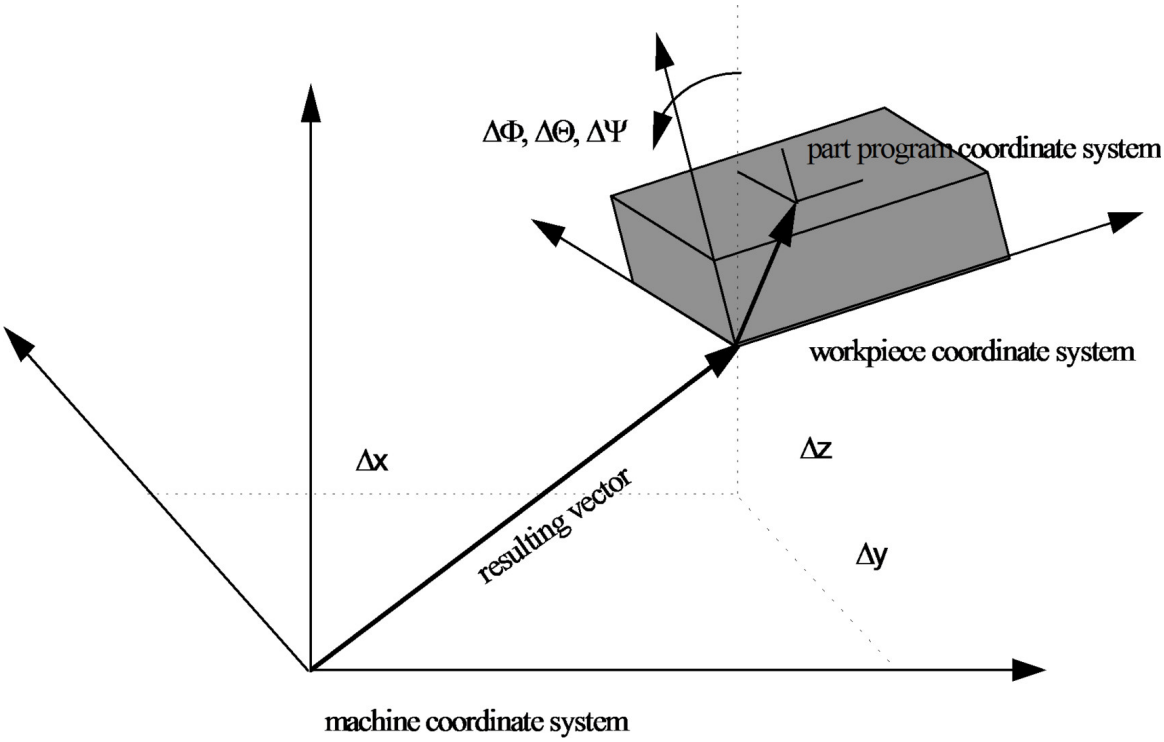


Fig. 8: Coordinate systems

Consequently, data or variables described below always refer to a specific coordinate system. Three coordinate systems are shown in the figure above. The offsets depicted Δx, Δy and Δz and the orientations ΔΦ, ΔΘ and ΔΨ represent the transformation parameters from mcs to wcs. The orientation angles are Euler angles.

Naming must be supplemented by means of an additional suffix in order to allow for the time aspect. The figure below shows the machining direction of a machining block. The **end position** represents the programmed value; the **active position** is the current value of the interpolator; and the **current position** is the actual position including the control error.

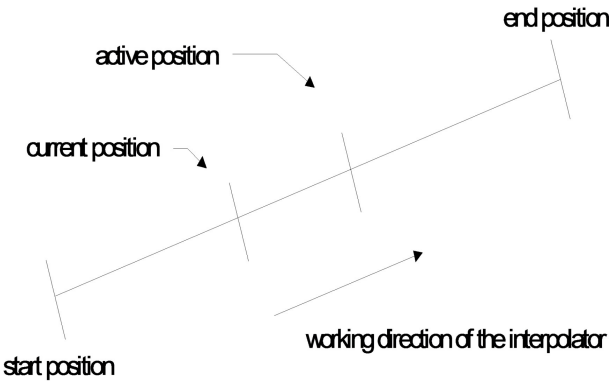


Fig. 9: Position values in PC and AC

2.3 Description of the axis-specific interface

2.3.1 Axis identification

Axis name (PCS)	
Description	Name of the logical axis used for the current reference in the current automatic program / manual block (e.g. X, Y, Z). This may be changed by default when the channel (SDA-MDS list) is programmed or dynamically in the NC program by means of a swap command.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].axis_name_r.**zeichen(HLI_ACHS_NAME_LAENGE)**
Data type	STRING(HLI_ACHS_NAME_LAENGE)
Access	PLC reads

Axis number (PCS)	
Description	Logical axis number which is unique system-wide for the axis which is currently traversed on the channel. The logical axis number is defined arbitrarily in the axis parameter list and is required to identify the axis in case of an axis swap, among other things (e.g. #CALL AX [X, 1, **1,** 0]).
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**log_achs_nr_r** gpAx[*axis_idx*]^.ipo_state.**log_achs_nr_r**
Data type	UINT
Value range	[1, MAX_UNS16] In TwinCAT normally [1, gNrAX]
Access	PLC reads

Axis spindle (ACS)	
Description	Indicates whether the axis is currently interpolated by the path interpolator (e.g. thread drilling, tapping) or the BF spindle.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.ipo_state.**spindle_axis_r**
Data type	BOOL
Value range	[TRUE = the axis is traversed by the BF spindle, FALSE]
Access	PLC reads

Axis type (PCS)	
Description	Type of axis
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**axis_type_r** gpAx[*axis_idx*]^.ipo_state.**type_r**
Data type	UINT
Value range	1 = Translator, 2 = Rotator, 4 = Spindel
Access	PLC reads

Channel number of the axis	
Description	Number of the channel by which the axis is currently commanded. == 0: Axis is currently not on a channel or spindle is not processing any channel command. != 0: Axis belongs to a channel or spindle is executing a channel command.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**cnc_channel_r** gpAx[*axis_idx*]^.ipo_state.**cnc_channel_r**
Data type	UINT
Value range	[1, HLI_SYS_CHNMAX]
Access	PLC reads

2.3.2 Velocities of an axis

Command velocity	
Description	Current command velocity of the axis
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**active_rev_r**
Data type	DINT
Unit	[µm/s]
Access	PLC is reading
Special features	

Actual velocity.	
Description	Current actual velocity of the axis
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state**.current_rev_r**
Data type	DINT
Unit	[µm/s]
Access	PLC is reading
Special features	

2.3.3 Axis positions

2.3.3.1 Axis positions in the PCS

Target position (PCS)	
Description	Target position of the current NC block.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**cmd_position_r**
Data type	DINT
Unit	0,1 µm
Access	PLC reads

Command position (PCS)	
Description	Position preset in the current cycle as setpoint.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**act_position_r**
Data type	DINT
Unit	0.1 µm
Access	PLC is reading

Actual position (PCS)	
Description	Actual ACS position converted in the PCS.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**current_position_r**
Data type	DINT
Unit	0,1 µm
Access	PLC reads

PCS position after a dynamic CS superimposition	
Description	Interpolated position in the selected PCS, including the superimposition of the dynamic coordinate system on the channel interface.
Signal flow	CNC □ PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**active_pos_pcs_dyn_cs_r**
Data type	DINT
Unit	0.1 µm
Access	PLC is reading
Special feature	Available as of V3.1.3105.1

Distance to go to geometry end (PCS)	
Description	Remaining absolute distance to go of the axis until the next programmed #DIST TO GO END.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.coord_r[axis_idx].**dist_to_geom_end_r**
Data type	DINT
Unit	0.1µm
Access	PLC is reading
Special feature	Available as of V3.1.3079.27

2.3.3.2 Position offsets in the PCS

Manual mode offset (PCS)	
Description	Current manual mode offset.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**man_offset_r**
Data type	DINT
Unit	0,1 µm
Access	PLC reads

Total offset (PCS)	
Description	Sum of all active offsets TotalOffset = - Offsets + Tool
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**total_offset_r**
Data type	DINT
Unit	0,1 µm
Access	PLC reads

Special features	Offsets are included in the total as negative values.
	
	For 2.5 D machining:
	TotalOffset = - Offset_G92 - Offset_NP - + WZ_Achsversätze
	
	Calculation of PCS coordinates based on ACS coordinates:
	PCS = ACS + TotalOffset
	
	Example program:
	
	`%total_offset`
	`N00 X100 (ACS = 100mm)`
	`N10 G92 X11 (total_offset = -11mm)`
	`N20 X200 (ACS = 211mm)`
	`N30 D1 (total_offset = 64mm,)`
	`( bei wz[1].ax_versatz[0] 750000[0.1µm])`
	`N40 V.G.WZ_AKT.V[0]=55 (total_offset = 44mm)`
	`N30 X300 (-> ACS coordinate = 256 mm)`
	`M30`

2.3.3.3 MCS axis position

MCS position after all dynamic CS superimpositions	
Description	Interpolated position in the MCS, including the superimposition of all dynamic coordinate systems on the channel interface.
Signal flow	CNC PLC
ST path	gpCh[channel_idx]^.bahn_state.coord_r[axis_idx].**active_pos_mcs_dyn_cs_r**
Data type	DINT
Unit	0.1 µm
Access	PLC is reading
Special feature	Available as of V3.1.3105.1

2.3.3.4 Axis positions in the ACS

Target position (ACS)	
Description	Target position in the current NC block, ACS. This represents the target position of the program coordinate system referred to the axes. It is valid only as long as no transformation is active. Currently, the target position is not transformed back onto the axes.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**end_position_acs_r**
Data type	DINT
Unit	0,1 µm
Access	PLC reads

Actual position (ACS)	
Description	Actual position of the current cycle in the axis coordinate system
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**current_position_acs_r**
Data type	DINT
Unit	0,1 µm
Access	PLC reads

Command position (ACS)	
Description	Command position of current cycle in the axis coordinate system
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**active_position_acs_r**
Data type	DINT
Unit	0.1 µm
Access	PLC is reading

Command position (ACS) from interpolator	
Description	Command position in the axis coordinate system which is refreshed each interpolator cycle.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**acs_position_r**
Unit	0.1 µm
Data type	DINT
Access	PLC is reading

Position offset	
Description	Display of the current offset of the absolute position between the position controller axis (real physical axis) and the interpolator axis (logical axis) currently linked in the axis coordinate system. The display here shows an offset between the interpolator position and the position controller position caused by the use of functions such as “jog of path” (see [FCT-C15]. The offset displayed contains no offsets caused by zero offsets, for example.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**position_offset_r**
Data type	DINT
Unit	0.1 µm
Access	PLC is reading

Look ahead command positions	
Description	This array allows the user to access a limited number of future interpolator command values in the PLC. For example, this can be used to carry out monitoring in the PLC based on a future axis movement that has not yet been executed. The delay is the delayed position output between the interpolator and position controller and therefore the effectiveness on the drive. The maximum possible index in this array is defined in the constant HLI_POS_LOOKAHEAD_MAXIDX. The array element position[HLI_POS_LOOKAHEAD_MAXIDX] contains the setpoint calculated in the current interpolator cycle. The element position[HLI_POS_LOOKAHEAD_MAXIDX **- 1**] contains the setpoint of the previous cycle; position[0] contains the setpoint from before HLI_POS_LOOKAHEAD_MAXIDX cycles. When the axis parameter P-AXIS-00256 is set, the delay between the interpolator and position controller is deactivated; in this case all array elements contain the setpoint calculated in the current interpolator cycle.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.ipo_state.**position_lookahead_r.position[i]**
Data type	ARRAY [0..HLI_POS_LOOKAHEAD_MAXIDX] OF DINT
Unit	0.1 µm
Access	PLC is reading
Special features	**This data item is only available as of Version V3.00 and higher.** This cyclic look ahead must not be confused with the function “Estimation of future data“.

2.3.4 Display data of tool centre point in MCS

Tool centre point position (MCS)	
Description	Command position of tool centre point in machine coordinate system MCS. The value is refreshed in each interpolation cycle.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**w0_position_r**
Data type	DINT
Unit	[0.1 µm]
Access	PLC is reading

Velocity at tool centre point (MCS)	
Description	Velocity value of tool centre point in machine coordinate system MCS. The value is refreshed in each interpolation cycle.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**w0_velocity_r**
Data type	DINT
Unit	[0.1 µm/s]
Access	PLC is reading

Acceleration at tool centre point (MCS)	
Description	Acceleration value of tool centre point in machine coordinate system MCS. The value is refreshed in each interpolation cycle.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**w0_acceleration_r**
Data type	DINT
Unit	[0.1 µm/s²]
Access	PLC is reading

2.3.5 Status information of an axis

Axis state (PCS)			
Description	Axis state, PCS		
Signal flow	CNC → PLC		
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**state_r**		
Data type	UINT		
Value range	**Value**	**PLC constant**	**Meaning**
	1	HLI_AXIS_READY	The axis is ready and moves according to the specified command values after a command.
	3	HLI_AXIS_ACTIVE	The axis is currently moved by the CNC due to an NC command or manual mode.
	5	HLI_AXIS_HOLD	Axis cannot be moved by the CNC since feed-hold is set.
Access	PLC is reading		

Axis state (ACS)			
Description	Axis state, ACS		
Signal flow	CNC → PLC		
ST path	gpAx[axis_idx]^.lr_state.**axis_state_r**		
Data type	UINT		
Value range	**Value**	**PLC constant**	**Meaning**
	1	HLI_AXIS_READY	The axis is ready and moves according to the specified command values after a command.
	3	HLI_AXIS_ACTIVE	The axis is currently moved by the CNC due to an NC command or manual mode.
	5	HLI_AXIS_HOLD	The CNC cannot move the axis because an external signal is set, such as feedhold or tracking mode, or the required drive enables are missing.
	7	HLI_AXIS_ERROR	After an error (in the drive or CNC, e.g. a software limit switch violation) the axis is in error state. Commanding a new motion is only possible after a CNC reset.
Access	PLC is reading		
Special features	**Even if an axis is not moved in the PCS, a corresponding Cartesian or kinematic transformation may nevertheless execute a motion of the physical axis.** **Example: 90° rotation about Z; Y is moved if X is programmed.**		

Synchronisation state, axis			
Description	State of an axis that participates in the synchronisation of a slave axis to a master axis. Due to its configuration the slave axis execute superimposed motions by means of a single axis interpolator. A typical application is the synchronisation of a slave axis to a conveyor belt.		
Signal flow	CNC → PLC		
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**conv_sync_state_r**		
Data type	USINT		
Value range	**Value**	**PLC constant**	**Meaning**
	0	HLI_CONV_SYNC_STATE_IDLE	No synchronisation motion is executed.
	1	HLI_CONV_SYNC_STATE_WAIT	Axis waits for the start of the synchronisation motion.
	2	HLI_CONV_SYNC_STATE_LINKED	Slave axis is synchronous with the conveyor belt.
	3	HLI_CONV_SYNC_STATE_SYNC_IN	Clave axis is synchronised with the conveyor belt position.
	4	HLI_CONV_SYNC_STATE_SYNC_OUT	Axis releases synchronisation to the conveyor belt and stops at the programmed position.
	5	HLI_CONV_SYNC_STATE_STOP	Axis releases synchronisation to the conveyor belt and stops.
	6	HLI_CONV_SYNC_STATE_SYNC_IN_TIME	Master axis stops and slave axis is interpolated over time.
	7	HLI_CONV_SYNC_STATE_ERROR	Axis is in error state.
	8	HLI_CONV_SYNC_STATE_FORCE_SYNC_ERROR	User forced an error.
Access	PLC is reading		

Distance to go (PCS)	
Description	Distance to go in the current NC block, difference between target position and command position.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**dist_to_go_r**
Data type	DINT
Unit	0.1 µm
Access	PLC is reading

Homing done (PCS)	
Description	The axis completed homing successfully and is now referenced.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].**homing_done_r**
Data type	BOOL
Value range	[TRUE = axis is referenced, FALSE]
Access	PLC is reading

Homing done (ACS)	
Description	The axis completed homing successfully and is now referenced.
Signal flow	CNC → PLC
ST path	gpAx[axis_idx]^.lr_state.**homing_done_r**
Data type	BOOL
Value range	[TRUE = axis is referenced, FALSE]
Access	PLC is reading

position lag	
Description	Current position lag of the axis, difference between command position and actual position.
Signal flow	CNC → PLC
ST path	gpAx[axis_idx]^.lr_state.**following_error_r**
Data type	DINT
Unit	0.1 µm
Access	PLC is reading

Axis in control window	
Description	The axis is located in the control window, i.e. the amount of position lag is less than the current effective position window (P-AXIS-00236 or P-AXIS-00472).
Signal flow	CNC → PLC
ST path	gpAx[axis_idx]^.lr_state.**in_window_r**
Data type	BOOL
Value range	[TRUE = axis is located in control window, FALSE]
Access	PLC is reading

Axis in position	
Description	The axis is located in position, i.e. the control window is reached (see above) and the axis is not interpolated.
Signal flow	CNC → PLC
ST path	gpAx[axis_idx]^.lr_state.**in_position_r**
Data type	BOOL
Value range	[TRUE = Axis in position, FALSE]
Access	PLC is reading

Axis is moved	
Description	The axis is currently co-used for the programmed path motion.
Signal flow	CNC → PLC
ST path	gpAx[axis_idx]^.lr_state.**is_moved_r**
Data type	BOOL
Value range	[TRUE = Axis is moved, FALSE]
Access	PLC is reading

Axis is moving	
Description	The axis moves, i.e. a command value is generated for this axis for the current interpolation.
Signal flow	CNC → PLC
ST path	gpAx[axis_idx]^.lr_state.**is_moving_r**
Data type	BOOL
Value range	[TRUE = Axis is traversed, FALSE]
Access	PLC is reading
Special features	If an axis is traversed, e.g. at a very low feed rate in the NC program, the axis is considered as moved for the complete path motion. However, it may not receive a new command position in every cycle due to quantisation. This means that the display “Axis is traversed” is not indicated in each cycle. Similarly, the axis is considered as traversed if override = 0 (traverse at velocity 0). By contrast, no velocity is commanded in the case of a FEEDHOLD, i.e. the axis is considered as not traversed.

Axis is moving forward	
Description	The last output setpoint generated led to a positive direction of motion.
Signal flow	CNC → PLC
ST path	gpAx[axis_idx]^.lr_state.**is_moving_forward_r**
Data type	BOOL
Value range	[TRUE = Axis is moving forward, FALSE]
Access	PLC is reading
Special features	This state display is unable to detect whether the axis is actually moving. Instead, use the state information gpAx[axis_idx]^.lr_state.**is_moving_r**.

Control unit enable, state	
Description	Indicates whether control unit is enabled for the axis.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**control_loop_enabled_r**
Data type	BOOL
Value range	[TRUE = Control unit enabled for axis, FALSE = No control unit enable. Axis cannot be moved by the CNC.]
Access	PLC is reading

Block number	
Description	Current block number of active NC block
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**block_nr_r**
Data type	DINT
Access	PLC is reading

Axis supply	
Description	The command of the physical axes is executed by the logical axis specified in the NC channel.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**link_to_ipo_r**
Data type	UINT
Value range	[= 0 – not linked, > 0 – logical axis number]
Access	PLC is reading

Cyclic drive cyclic values are valid	
Description	The actual values supplied by the drive to the cyclic process data are valid. This signal is reset if the control unit is active and the cyclic transfer of process data is interrupted.
Signal flow	CNC → PLC
ST path	gpAx[axis_idx]^.lr_state.**cyclic_drive_data_ok_r**
Data type	BOOL
Value range	[TRUE = Process data is valid, FALSE]
Access	PLC is reading
Special features	This signal is dependent on the drive type as follows: SERCOS II: Signal is TRUE if the phase of the SERCOS ring is 4 and at least one of the bits 0x4000 or 0x8000 is set in the drive status word. **SERCOS over EtherCAT**: Signal is TRUE if the process datum WcState is 0 and at least one of the bits 0x4000 or 0x8000 is set in the drive status word. **CANopen**: Signal is TRUE if the process datum WcState is 0 and the drive status word contains a valid status according to the CANopen DS402 specification. PROFIBUS: Signal is TRUE if the life counter sign of the PROFIBUS slave is running. For all other drive types, the signal is set to TRUE immediately after CNC start-up.

Configured axis mode	
Description	The axis mode configured in the axis parameter list (P-AXIS-00015) is indicated.
Signal flow	CNC →PLC
ST path	gpAx[*axis_idx*]^.lr_state.**axis_mode_r**
Data type	UDINT
Value range	See description of axis parameter P-AXIS-00015..
Access	PLC is reading

Axis number of gantry master axis	
Description	If the axis is a gantry slave axis, the logical axis number of the master axis in this element (see P-AXIS-00070) is indicated; otherwise 0.
Signal flow	CNC → PLC
ST path	gpAx[axis_idx]^.lr_state.**gantry_master_nr_r**
Data type	UINT
Access	PLC is reading

Axis assignment	
Description	The display shows the physical axes to which the commanded values of the logical axes are of the channel output.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.ipo_state.**link_to_lr**
Data type	UINT
Value range	[= 0 – not linked, > 0 – logical axis number]
Access	PLC is reading

Axis-specific interpolator, state	
Description	This element indicates various states of the axis-specific interpolator. The states are bit-encoded and may be active at the same time.
Signal flow	CNC → PLC
ST path	gpAx[axis_idx]^.ipo_state.**internal_ipo_state_r**
Data type	UDINT
Special features	The element is a bit string. Several states are therefore displayed at the same time.
Value range	The assignment of bits to states is listed in the table below.
Access	PLC is reading

PLC constant	**Value**	**Meaning**
HLI_AX_INDP_INTERPOLATION	16#00000002	This is an independent axis for which setpoints are generated.
HLI_AX_INDP_M_FUNC_PENDING	16#00000004	This is an independent axis which waits for acknowledgement of at least one technology function (M, H, S).
HLI_AX_INDP_ERROR	16#00000008	An error occurred in an independent axis.
HLI_AX_INDP_TIME_INTERPOLATION	16#00000010	Dwell time is active for an independent axis.
HLI_AX_MAN_MV_BACK_TO_START	16#00000020	Axis is moved because it received the command to reduce to 0 the position offset accumulated by motion in manual mode.
HLI_AX_MAN_MV_BACK_WAIT_STOP	16#00000040	The motion to reduce the position offset to 0 and which was generated by a motion of the axis in manual mode was interrupted and the axis is still in the deceleration phase to standstill.
HLI_AX_TRANSM_TO_PLC_IMPOSSIBLE	16#00040000	This is an independent axis where the axis-specific interface is assigned to output technology functions.
HLI_AX_FEEDHOLD	16#001000000	Axis-specific feedhold is active.

States of the axis-specific interpolator

Type of axis coupling			
Description	This indicates whether the axis is a slave axis in an axis coupling; see also [PROG//#AXIS LINK].		
Signal flow	CNC → PLC		
ST path	gpAx[*axis_idx*]^.ipo_state.**axis_link_mode_r**		
Data type	INT		
Value range	**Value**	**PLC constant**	**Meaning**
	-1	HLI_AXIS_LINK_NONE	No axis coupling active for this axis
	0	HLI_AXIS_LINK_NORMAL	The axis is a slave axis of an axis coupling
	1	HLI_AXIS_LINK_SPDL	The axis is a slave axis of a spindle coupling
	2	HLI_AXIS_LINK_GANTRY	The axis is a slave axis of a soft gantry combination
	3	HLI_AXIS_LINK_MIRROR	The axis is a slave axis of an axis coupling; the axis motion mirrors the master axis
Access	PLC is reading		
Special features	The indicated axis coupling in only active when a master axis number is indicated at the same time.		

Axis number of master axis when axis coupling is active	
Description	When the axis is coupled to a master axis (see [PROG//#AXIS LINK]) the logical axis number of the master axis is indicated here.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.ipo_state**.ax_link_master_ax_nr**
Data type	UINT
Value range	0 : No coupling active > 0: logical axis number of the master axis
Access	PLC is reading

Axis type (ACS))		
Description	The configured axis type (P-AXIS-00018) is indicated.	
Signal flow	CNC → PLC	
ST path	gpAx[*axis_idx*]^.ipo_state**.type_r**	
Data type	UINT	
Value range	**Value**	**Meaning**
	0x0001	Linear axis
	0x0002	Rotary axis
	0x0004	Spindle
Access	PLC is reading	
Special features	**Is currently not programmed**	

Lift function is active	
Description	This data item indicates whether a lift function is active or not. An active lift function is associated with execution of a lift movement.
Signal flow	CNC->PLC
ST path	gpAx[axis_idx]^.ipo_state.**lift_active**
Data type	BOOL
Value range	TRUE = Lift function is active FALSE = Lift function is not active
Access	PLC is reading

Lift movement is suppressed	
Description	This data item indicates whether a lift movement is suppressed. This is always the case when the programmed minimum path distance P-CHAN-00244 is not reached.
Signal flow	CNC->PLC
ST path	gpAx[axis_idx]^.ipo_state.**lift_suppressed**
Data type	BOOL
Value range	TRUE = Lift movement is suppressed FALSE = Lift movement is not suppressed.
Access	PLC is reading
Special feature	The following special case of the status display occurs when you use “Advanced Lifting”: If the lift movement is suppressed due to insufficient travel distance and a target position of the Z axis is programmed at the same time, the lift function is active in order to be able to approach the target position of the Z-axis. In this case, the status of the lift function is simultaneously displayed as “active” and “suppressed”.

Axis is moved	
Description	The axis is currently co-used for the programmed path motion.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.ipo_state.**is_moved_r**
Data type	BOOL
Value range	[TRUE = Axis is moved, FALSE]
Access	PLC is reading

Axis is moving	
Description	The axis moves, i.e. a command value is generated for this axis for the current interpolation.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.ipo_state.**is_moving_r**
Data type	BOOL
Value range	[TRUE = Axis is traversed, FALSE]
Access	PLC is reading
Special features	If an axis is traversed, e.g. at a very low feed rate in the NC program, the axis is considered as moved for the complete path motion. However, it may not receive a new command position in every cycle due to quantisation. This means that the display “Axis is traversed” is not indicated in each cycle. This display indicates the status of the interpolator. The axis can still be moved at a later point in time to generate the command value in the CNC. For example, this is possible with a coupled axis [FCT-A9// Axis coupling via HLI]. The variable lr_state.is_moved_r [▶ 31] should be used for these cases.

2.3.5.1 Drive of an axis

Drive type		
Description	Type of axis drive The drive type is specified for each axis in the entry kenngr.antr_typ.	
Signal flow	CNC → PLC	
ST path	gpAx[*axis_idx*]^.lr_state.**drive_type_r**	
Data type	UINT	
Value range	**Value**	**Meaning**
	1	Conventional drive (+-10 V), not used
	2	SERCOS drive
	3	Profidrive
	4	Drive Simulation
	5	Lightbus
	6	Terminal axis (+-10V) via bus
	7	Real time Ethernet
	8	CANopen drive
	16	Virtual drive
	32	CAN drive (option)
Access	PLC reads	

Drive ready for power connection / drive torque enabled	
Description	The axis drives is ready for power connection / drive torque is enabled.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**ready_for_power_on_r**
Data type	BOOL
Value range	[TRUE = Ready for power connection, FALSE]
Access	PLC reads
Special features	**SERCOS drives** With SERCOS drives, this information is derived from bit 14 of the status word. To determine the drive state the datum ready_for_control_loop_on_r [▶ 39] must also be considered. **CANopen drives** With CANopen drives this signal is set when the drive is in the ‘Operation enabled’ state; in this state the drive is torque enabled. This drive state is indicated by the state word xxxx xxxx x01x 0111 transferred by the drive (binary notation, x = don’t care).

Drive ready for operation	
Description	Drive control section and power supply are ready for operation.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**ready_for_control_loop_on_r**
Data type	BOOL
Value range	[TRUE = Drive ready for operation, FALSE]
Access	PLC reads

Special features	**SERCOS drives**
	With SERCOS drives, this information is derived from bit 15 of the status word. In order to evaluate the drive state the datum ready_for_power_on_r [▶ 38] must also be considered. The various combinations are described below.
	If
	`(gpAx[axis_idx]^.lr_state.ready_for_power_on_r == FALSE) AND (gpAx[axis_idx]^.lr_state.ready_for_control_loop_on_r == FALSE)`
	this means that the drive is not ready for power connection.
	If
	`(gpAx[axis_idx]^.lr_state.ready_for_power_on_r == TRUE) AND (gpAx[axis_idx]^.lr_state.ready_for_control_loop_on_r == FALSE)`
	this means that the drive is ready for power connection and the internal drive initialisation procedures are completed.
	If
	`(gpAx[axis_idx]^.lr_state.ready_for_power_on_r == FALSE) AND (gpAx[axis_idx]^.lr_state.ready_for_control_loop_on_r == TRUE)`
	this means that the drives control section and power supply are ready for operation but the drive has no torque.
	If
	`(gpAx[axis_idx]^.lr_state.ready_for_power_on_r == TRUE) AND (gpAx[axis_idx]^.lr_state.ready_for_control_loop_on_r == TRUE)`
	this means that the drive is ready for operation,
	i.e. “feed rate enable, axis” [▶ 49] and “drive on” [▶ 49] are set to true and are active. This means that the drive is torque enabled and follows the position command values in the NC kernel.
	
	CANopen drives
	With CANopen drives this signal is set when the drive is in a valid state not equal to ‘Not ready to switch on’. The ‘Not ready to switch on’ state is indicated in the drive state word by the value xxxx xxxx x0xx 0000 (binary notation x=don’t care).
	Drive has completed self-test and initialisation successfully.

Drive error	
Description	An error occurred in the drive. The drive is therefore interlocked.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**error_r**
Data type	BOOL
Value range	[TRUE = Error occurred, FALSE]
Access	PLC reads
Special features	**SERCOS drives** With SERCOS drives, this information is derived from bit 13 of the status word. **CANopen drives** With CANopen drives this information is derived from bit 3 (bitmask 0x0008) of the state word.

Drive state		
Description	State information delivered by the drive via the fieldbus. This element contains the following data dependent on the drive type:	
	Drive type (P-AXIS-00020**)**	**Datum**
	SERCOS	Value of SERCOS parameter S-0-135 (drive state).
	CANopen	Value of CANopen object 0x6041 (drive state)
	PROFIDRIVE	Value of drive state word 1 (ZSW1)
	Lightbus	Value of drive signal 23 from the drive (drive state 'DriveState3')
	For further information refer to the drive documentation or the specification of the drive profile used.	
Signal flow	CNC → PLC	
ST path	gpAx[*axis_idx*]^.lr_state.**native_drive_state_r**	
Data type	UINT	
Access	PLC reads	

Read drive data	
Description	Data transferred by the drive to the NC kernel. The content is application-specific This data is also provided in parallel in the state of the control unit, see read/write drive data cyclically
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**uns32_1** gpAx[*axis_idx*]^.lr_state.**uns32_2** gpAx[*axis_idx*]^.lr_state.**uns32_3** gpAx[*axis_idx*]^.lr_state.**uns32_4**
Data type	UDINT
Access	PLC reads

Special features	**Currently, data transfer can only be used for SERCOS drives.**
	Therefore, enable transfer of the value in the axis parameter list to the drive, e.g.:
	
	`# Cyclically read the 4 byte PLC value`
	`# uns32_3 auf S-0-0819`
	`antr_digital.typ.sercos.at[1].ident_nr 0819`
	`antr_digital.typ.sercos.at[1].ident_len 4`
	`antr_digital.typ.sercos.at[1].nc_ref LR_VAR3_IN`

DRIVE_STATE_MODE_0	
Description	Current drive state, i.e. Position controller DRIVE_STATE_MODE_0 0x00000001
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**mode_0_r**
Data type	BOOL
Value range	
Access	PLC reads

DRIVE_STATE_MODE_1	
Description	DRIVE_STATE_MODE_1 0x00000002
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**mode_1_r**
Data type	BOOL
Value range	
Access	PLC reads

DRIVE_STATE_MODE_2	
Description	DRIVE_STATE_MODE_2 0x00000004
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**mode_2_r**
Data type	BOOL
Value range	
Access	PLC reads

DRIVE_STATE_MODE_3	
Description	DRIVE_STATE_MODE_3
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**mode_3_r**
Data type	BOOL
Value range	
Access	PLC reads
Special features	**Is currently not programmed**

DRIVE_STATE_MODE_4	
Description	DRIVE_STATE_MODE_4
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**mode_4_r**
Data type	BOOL
Value range	
Access	PLC reads
Special features	**Is currently not programmed**

DRIVE_STATE_MODE_5	
Description	DRIVE_STATE_MODE_5
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**mode_5_r**
Data type	BOOL
Value range	
Access	PLC reads
Special features	**Is currently not programmed**

DRIVE_STATE_MODE_6	
Description	DRIVE_STATE_MODE_6
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**mode_6_r**
Data type	BOOL
Value range	
Access	PLC reads
Special features	**Is currently not programmed**

2.3.5.2 Compensation of an axis

Compensation of drive drift	
Description	State of activation of drive drift compensation
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**drift_error_comp_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Access	PLC reads
Special features	**Is currently not programmed**

Compensation of spindle leadscrew error	
Description	State of activation of spindle leadscrew error compensation
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**pitch_error_comp_r**
Data type	BOOL
Value range	[TRUE = Spindle leadscrew error compensation active, FALSE]
Access	PLC reads
Special features	**Is currently not programmed**

Compensation of temperature influence	
Description	Activation of drive temperature compensation
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**temperature_error_comp_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Access	PLC reads
Special features	**Is currently not programmed**

Compensation of backlash error	
Description	State of activation of compensation of drive backlash error
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**backlash_error_comp_r**
Data type	BOOL
Value range	[TRUE = Compensation of backlash error active, FALSE]
Access	PLC reads
Special features	**Is currently not programmed**

Feed forward	
Description	State of activation of feedforward control
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**feed_forward_r**
Data type	BOOL
Value range	[TRUE = Feedforward control active, FALSE]
Access	PLC reads
Special features	**Is currently not programmed**

2.3.5.3 Measurement

Measuring probe state	
Description	State of measuring probe
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**measure_equipment_active_r**
Data type	BOOL
Value range	[TRUE = probe active (actuated), FALSE]
Access	PLC is reading
Special features	**Parameter is included for downward compatibility reasons.** Parameter is not supplied in current versions.

Measurement value valid	
Description	A valid measurement value was latched by the drive
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**measure_value_ok_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Access	PLC is reading
Special features	

State of measuring probe	
Description	The state of the probe indicates whether it is actuated or not.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**probe_actuated_r**
Data type	BOOL
Value range	[TRUE = probe actuated, FALSE]
Access	PLC is reading
Note	Parameter available as of V3.3080.04

Measurement run active	
Description	This data item indicates whether a measurement run is active or not.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**measurement_active_r**
Data type	BOOL
Value range	[TRUE = Measurement run active, FALSE]
Access	PLC is reading
Note	Parameter available as of V3.3080.04

2.3.5.4 Link to ADS

Box identification	
Description	Device-dependent data defined when the system is configured (e.g. system manager) can be read out here.
Data type	HLI_TWINCAT_BOX_DATA
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**twincat_box**
Access	PLC reads
Special features	**This element is not available in every PLC runtime environment.**
Elements of the data type	
ST element	**.ads_netid_r**
Description	This defines the AmsNetId of the TwinCAT computer to execute the function. An empty string can also be specified for the local computer. The PLC variable of this type is a string containing the AMS network identifier of the target device for which the ADS command is intended. The string consists of six numerical sections separated by dots. Each numerical section includes a number between 0 and 254. Valid AMS network address include: "1.1.1.2.7.1" or "200.5.7.170.1.7". If an empty string is transferred, the AMS network identifier of the local device is automatically accepted.
ST element	**.ads_port_r**
Description	ADS devices in a TwinCAT network topology are identified by an AMS network address and a port number. The port number of the ADS device is specified by the system manager during configuration.
ST element	**.channel_r**
Description	

Device identification	
Description	The IO device where the function is executed is specified by the device ID. The device IDs are specified by the TwinCAT system manager during hardware configuration.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**twincat_device**
ST element	**.id_r**
Data type	UDINT
Access	PLC reads
Special features	The device ID is currently used for ring global actions (e.g. phase switching or reading). **This element is not available in every PLC runtime environment.**

2.3.5.5 Pre-calculated status information

Precalculated data is valid, axis	
Description	This data element indicates whether precalculated axis data is valid. If the data element is TRUE, the values are valid for the precalculation of position [▶ 47], velocity [▶ 48] and acceleration [▶ 48] at a future point in time. FALSE indicates that no data could be calculated for the future point in time. The point in time is defined by the Index 0 in P-CHAN-00324 or by #CHANNEL SET[ESA_TIME0=...].
Signal flow	CNC →PLC
ST Path	gpAx[*axis_idx*]^.ipo_state.**esa_data_valid**
Data type	BOOL
Value range	TRUE/FALSE
Access	PLC is reading
Special feature	**Available as of Build V3.1.3104.08**

Precalculated position, axis	
Description	Axis position at a future point in time. The point in time is defined by the Index 0 in P-CHAN-00324 or by #CHANNEL SET[ESA_TIME0=...].
Signal flow	CNC →PLC
ST Path	gpAx[*axis_idx*]^.ipo_state.**esa_pos**
Data type	LREAL
Access	PLC is reading
Special feature	**Available as of Build V3.1.3104.08**

Precalculated velocity, axis	
Description	Axis velocity at a future point in time. The point in time is defined by the Index 0 in P-CHAN-00324 or by #CHANNEL SET[ESA_TIME0=...].
Signal flow	CNC →PLC
ST Path	gpAx[*axis_idx*]^.ipo_state.**esa_vel**
Data type	LREAL
Access	PLC is reading
Special feature	**Available as of Build V3.1.3104.08**

Precalculated acceleration, axis	
Description	Axis acceleration at a future point in time. The point in time is defined by the Index 0 in P-CHAN-00324 or by #CHANNEL SET[ESA_TIME0=...].
Signal flow	CNC →PLC
ST Path	gpAx[*axis_idx*]^. ipo_state.**esa_acc**
Data type	LREAL
Access	PLC is reading
Special feature	**Available as of Build V3.1.3104.08**

2.3.6 Axis control commands

Feed rate enable, axis	
Description	Axis-specific feed rate enable The feed rate enable must be set for all axes to be moved. If this is not the case, no path motion occurs.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Special features	**SERCOS drives** With SERCOS drives, the datum is derived from bit 13 of the control word.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**release_feedhold**
Commanded, requested and return value	
ST element	**.command_w** **.request_r**
Data type	BOOL
Value range	[TRUE = Drive enable, Transition TRUE → FALSE: The drive is shut down in compliance with the acceleration parameters. FALSE = Drive STOP]
Return value	
ST element	**.state_r**
Data type	BOOL
Value range	[TRUE = Drive enable, FALSE = Drive STOP]
Redirection	
ST element	**.enable_w**

Feedhold ON/OFF, axis	
Description	Axis-specific feedhold The axis-specific feedhold on an axis acts as follows: • Axis interpolated and moved in path compound: – Stop all axes in path compound immediately • Axis interpolated and stopped in path compound: – Remaining axes in path compound continue to move, • Axis interpolated and stopped in path compound, only moves in the next block: – Remaining axes in path compound move to current block end and then stop. • Axis interpolated as single axis (manual mode, independent axis, SAI axis, spindle): – Only this axis is stopped. Otherwise, the global feedhold of the channel acts on al axes except for spindles. The operation mode of the global and axis-specific feedhold can be parameterised for these axes (P-AXIS-00529, P-AXIS-00540).
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpAx[*axis_idx*]^.ipo_mc_control.**feedhold**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = feedhold on, FALSE = feedhold off]
Redirection	
ST Element	**.enable_w**

Directional feedhold, axis			
Description	Axis-specific directional feedhold A directional feedhold can be set for all moving axes If an axis command is in the block direction, the axis-specific feedhold is set. An axis-specific directional feedhold acts on an axis in exactly the same way as an axis-specific feedhold. • Axis interpolated and moved in path compound: – Stop all axes in path compound immediately • Axis interpolated and stopped in path compound: – Remaining axes in path compound continue to move, • Axis interpolated and stopped in path compound, only moves in the next block: – Remaining axes in path compound move to current block end and then stop. • Axis interpolated as single axis (manual mode, independent axis, SAI axis, spindle): – Only this axis is stopped. The directional and normal axis-specific feedholds can be superimposed. Otherwise, the global feedhold of the channel acts on al axes except for spindles. The operation mode of the global and axis-specific feedhold can be parameterised for these axes (P-AXIS-00529, P-AXIS-00540).		
Data type	MC_CONTROL_UNS08_UNIT, see description of Control Unit [▶ 13]		
Access	PLC reads request_r + state_r and writes command_w + enable_w		
ST path	gpAx[axis_idx]^.ipo_mc_control.**directional_feedhold**		
Commanded, requested and return values			
ST element	**.command_w** **.request_r** **.state_r**		
Data type	USINT		
Value range	**Value**	**PLC constant**	**Meaning**
	0	HLI_DIRECTIONAL_FEED-HOLD_NO_FEEDHOLD	The axis is not blocked in any motion direction.
	1	HLI_DIRECTIONAL_FEED-HOLD_POS	The axis is blocked in the positive motion direction.
	2	HLI_DIRECTIONAL_FEED-HOLD_NEG	The axis is blocked in the negative motion direction.
	3	HLI_DIRECTIONAL_FEED-HOLD_POS_NEG	The axis is blocked in the positive and negative motion directions.
Redirection			
ST element	**.enable_w**		
Special feature	The axis drive is shut down in compliance with the acceleration parameters. This means that the axis moves a short distance in the blocked direction before stopping. The feedhold always acts on the complete current block, This means that a block can be blocked even if the current direction of axis motion is not pointing in the blocked direction. **Available as of V3.1.3079.26**		

	Examples: 1. Example of a circle: A full circle is programmed in the XY plane. A directional feedhold in negative direction is set for the X axis. Although the X axis first moves in the blocked direction after 180°, it does move through the entire circle since both motion directions are contained in the block. 2. Example of polynomial contouring: With polynomial contouring, splitting the contouring movement into several blocks can cause the axis compound to stop before the actual motion in the blocked direction. 3. Special case of a spindle: With a superimposed spindle motion, the direction of spindle motion is determined by set-points. This causes the axis to move a short distance in the blocked direction from stand-still before it stops.

Feedhold watchdog	
Description	This control unit is the interface of a watchdog which stops the CNC interpolator by activating a feedhold if the PLC fails. If the axis for which the watchdog was triggered is a member of an axis group and is currently commanded to be in motion, all other axes moved by this command also stop. If this control unit is enabled, the PLC must set the element **command_w** to TRUE in every PLC cycle. The NC kernel than resets the command back to FALSE. To enable this control unit function, the axis parameter **enable_feed_enable** (P-AXIS-00313) must be set in many versions of the NC kernel. In other version the function of this control unit is still available.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**enable_feed_wdg**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Signals that PLC is invoked cyclically and executes its function, FALSE]
Redirection	
ST element	**.enable_w**

Feed override, axis	
Description	Axis-specific feed override The axis-specific feed override permits weighting the axis path velocity with an additional factor. The axis-specific feed override only acts on an axis if the axis is currently moved in manual mode or as an independent axis and but in a path compound. Otherwise the global override of the channel acts on the axis. The axis-specific feed override also acts on single axes and spindles. With spindles the feed override also acts on the velocity acknowledgement of programmed M3/ M4 or MC_MoveVelocity commands. Velocity acknowledgement occurs when the weighted feed rate is reached and in extreme cases when the feedrate is 0. (see lr_state.rev_erreicht_r [▶ 89])
Data type	MC_CONTROL_UNS16_UNIT, see description of Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**override**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	UINT
Unit	0,1 %
Value range	[0 … P-AXIS-00109] The parameter P-AXIS-00109 is an axis-specific parameter. The value is typically 1000.
Redirection	
ST element	**.enable_w**

Feed override valid, axis	
Description	Axis-specific feed override valid.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**override_valid**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Axis-specific feed override valid, FALSE]
Redirection	
ST element	**.enable_w**

Drive ON	
Description	Drive ON
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Special features	**SERCOS drives** With SERCOS drives, the datum is derived from bit 15 of the control word.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**drive_on**
Commanded and requested values	
ST element	**.command_w** **.request_r**
Data type	BOOL
Value range	[TRUE = Drive ON, Transition TRUE → FALSE: The drive is shut down in the best possible manner. FALSE = Drive OFF]
Return value	
ST element	**.state_r**
Data type	BOOL
Value range	[TRUE = Drive ON, FALSE = Drive OFF]
Redirection	
ST element	**.enable_w**

Enabling of drive controller	
Description	Control unit enable ↔ axis-specific torque connection.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Special features	**SERCOS drives** With SERCOS drives, the datum is derived from bit 14 of the control word.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**torque_permission**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Torque connection, FALSE = Drive is torque-free]
Redirection	
ST element	**.enable_w**

The diagram below shows the relationship between HLI control units and the SERCOS control word or the SERCOS status word for SERCOS drives.

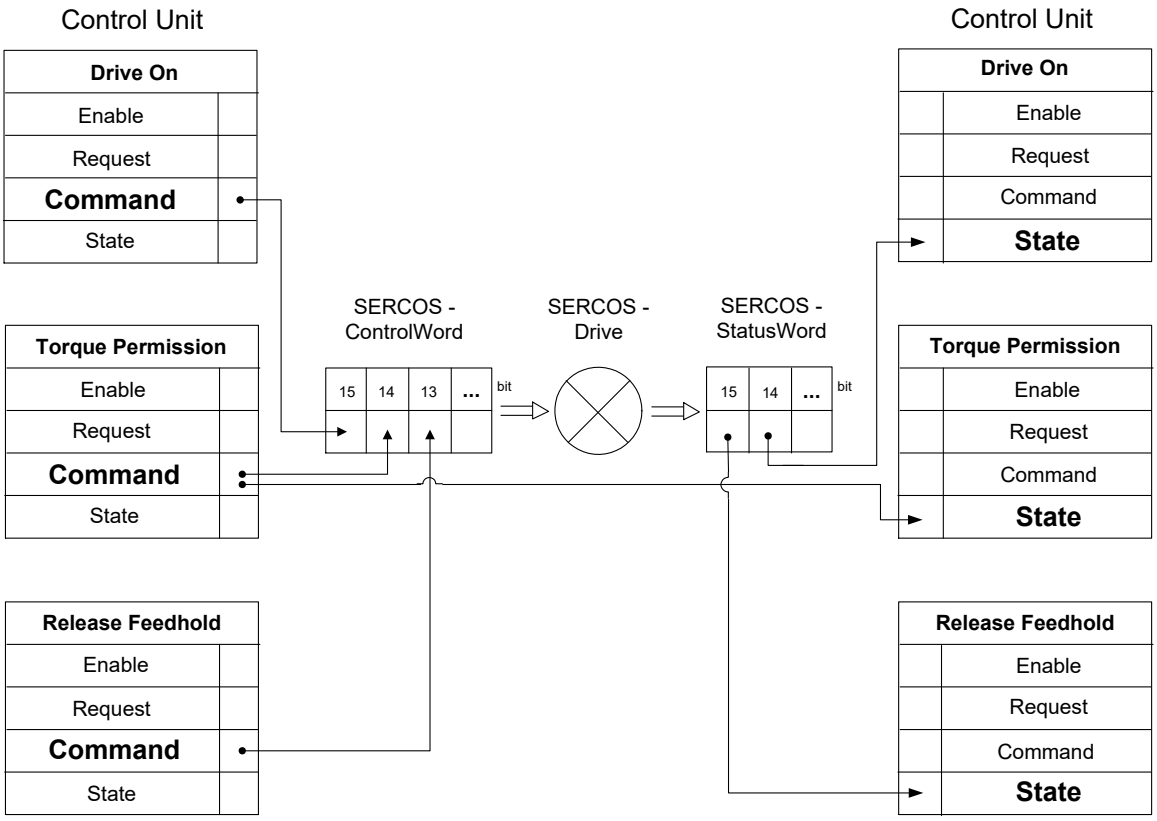


Fig. 10: Interaction of control units and SERCOS control or status words

Reference cam	
Description	Reference cam signal when homing.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Characteristics	When this signal is commanded, it only acts if the parameter lr_hw[i].cam_direct_access = 0 (P-AXIS-00036) is set in the axis machine data record of the associated axis. When the parameter lr_hw[i].cam_level (P-AXIS-00038) is used in the axis machine data record, the effect of this command can be parameterised from high-active to low-active. The default action is described below.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpAx[*axis_idx*]^.lr_mc_control.**reference_cam**
Commanded, requested and return values	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Reference cam switched, FALSE = Reference cam not switched]
Redirection	
ST Element	**.enable_w**

Reduced velocity, axis	
Description	When this signal is set, the axis velocity in G00 and G01 is reduced to the values defined in the axis parameters P-AXIS-00214 or P-AXIS-00155.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Special features	If this axis is in a channel, the limits of the other axes participating in the motion are also considered. The effective value for reduced velocity is determined so that none of the axes participating in the motion overshoots its configured limit. The response is then identical to the command via a channel.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**reduced_speed**
Commanded, requested and return values	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Reduced velocity active, FALSE = Reduced velocity not active]
Redirection	
ST element	**.enable_w**

Reduced velocity in zone 1, axis	
Description	By setting this signal, the axis velocity is limited to the velocity defined in the axis parameter P-AXIS-00030 when the axis is located within the range defined by the parameters P-AXIS-00085 and P-AXIS-00093. If necessary the axis is decelerated after entering the zone.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Special features	If the axis is in a channel, the limits of the other axes participating in the motion and also located within their configured zone are considered.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAX[*axis_idx*]^.ipo_mc_control.**reduced_speed_zone**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Reduced velocity in zone 1 active, FALSE = Reduced velocity in zone 1 not active]
Redirection	
ST element	**.enable_w**

Reduced velocity in zone 2, axis	
Description	By setting this signal, the axis velocity is limited to the velocity defined in the axis parameter P-AXIS-00030 when the axis is located within the range defined by the parameters P-AXIS-00097 and P-AXIS-00105. If necessary the axis is decelerated after entering the zone.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Special features	If the axis is in a channel, the limits of the other axes participating in the motion and also located within their configured zone are considered.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAX[*axis_idx*]^.ipo_mc_control.**reduced_speed_2_zone**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Reduced velocity in zone 2 active, FALSE = Reduced velocity in zone 2 not active]
Redirection	
ST element	**.enable_w**

Suppressing Read-In Enable	
Description	Suppressing Read-In Enable When the read-in enable (setting NoEfg) is cleared, the interpolator does not read any new previously decoded NC motion information, i.e. the motion is stopped at the end of the current commands in the interpolator.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**no_efg**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_**r
Data type	BOOL
Value range	[TRUE = No Read-In Enable, FALSE = Read-In Enable]
Redirection	
ST element	**.enable_w**

Machining simulation, axis	
Description	Activates and deactivates an axis-specific machining simulation. During machining simulation, all axis-specific technology commands of the NC program are no longer output to the PLC but are acknowledged internally.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**machining_simulation**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Machining simulation active, FALSE = Machining simulation inactive]
Redirection	
ST element	**.enable_w**

Ignoring minimum tool velocity	
Description	If a value for minimum tool velocity is configured at tool change, the NC kernel monitors that this limit is not undershot by specifying an override. This control unit can switch this response off and the override acts on the axis as specified.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Special features	Control unit is only effective if the axis is a spindle.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**ignore_vb_min_tool**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Undershooting minimum tool velocity permitted, FALSE]
Redirection	
ST element	**.enable_w**

OTC offset	
Description	When this wear offset is set, wear in the direction of this axis can be compensated. When the SURF_NORM_ORI mode is used (wear in the direction of the surface normal), the offset value must be assigned in the third axis.
Data type	MC_CONTROL_SGN32_UNIT, see description of Control Unit [▶ 13]
Special features	The wear offset is distributed by the CNC over several cycles.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpAx[*axis_idx*]^.ipo_mc_control.**otc_offset**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	DINT
Unit	0.1 µm
Value range	[-P-TOOL-00031, P-TOOL-00031]
Redirection	
ST Element	**.enable_w**

Move back manual mode offset	
Description	If manual mode is active in the channel and if the commanded axis fails to move, the axis is moved by this command so that afterwards manual mode offset is 0.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Characteristics	A rising edge (FALSE → TRUE) at **command_w** triggers the process. The signal is ignored if a manual mode motion is still active or manual mode offset is already 0.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpAx[*axis_idx*]^.ipo_mc_control.**manual_mv_back_to_start**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	rising edge (FALSE → TRUE) triggers backward motion
Redirection	
ST Element	**.enable_w**

Stop the motion “Move back manual mode offset”	
Description	The motion that was started by the Control Unit command “Move back manual mode offset” is stopped by the Control Unit with this command.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Special features	A rising edge (FALSE → TRUE) initiates the command. Up to final axis standstill, the datum Axis-specific interpolator [▶ 27] shows that the stop process is active by the HLI_AX_MAN_MV_BACK_WAIT_STOP [▶ 27] bit.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**manual_mv_back_stop**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = rising edge stops the motion, FALSE]
Redirection	
ST element	**.enable_w**

Tracking operation	
Description	The axis is set to tracking mode, i.e. the commanded setpoint is set identical to the actual value read in. Setting the setpoint and actual values equal is executed until **command_w** = TRUE.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Special features	By setting the setpoint and actual value equal, the current position lag is = 0. This may result in an external force (weight of the axis) slowly changing the axis position (drift).
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**follow_up**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = control loop open, FALSE]
Redirection	
ST element	**.enable_w**

Running out gantry difference	
Description	If the axis is a gantry slave axis and both the master and slave are referenced, the gantry difference is run out.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**gantry_on**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Gantry difference runout permitted, FALSE]
Redirection	
ST element	**.enable_w**

<table>
<tr><th colspan="3">Accepting reference position</th></tr>
<tr><td>Description</td><td colspan="2">Accept the reference position and mark the axis as referenced on a rising edge of this control unit. Depending on the value of the parameter P-AXIS-00278 the actual axis position is set to the following value:</td></tr>
<tr><td></td><td>P-AXIS-00278</td><td>Reference position of the axis</td></tr>
<tr><td></td><td>ABSOLUT</td><td>Value of P-AXIS-00152</td></tr>
<tr><td></td><td>OFFSET</td><td>Encoder position of the drive + P-AXIS-00279</td></tr>
<tr><td></td><td>PLC</td><td>Value in the refpos_position [▶ 63] control unit.</td></tr>
<tr><td></td><td>PLC_OFFSET</td><td>Encoder position of the drive + value in the refpos_position [▶ 63] control unit.</td></tr>
<tr><td></td><td colspan="2">The state_r variable indicates whether the reference position was set manually and the coordinate system was shifted as a result. The manual setting can be cleared by CNC-controlled homing (G74).
The reference position can also be set manually for an axis with absolute measuring system.</td></tr>
<tr><td>Data type</td><td colspan="2">MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]</td></tr>
<tr><td>Characteristics</td><td colspan="2">Edge evaluation: The function is triggered on the rising edge at the command input.</td></tr>
<tr><td>Access</td><td colspan="2">PLC reads request_r + state_r and writes command_w + enable_w</td></tr>
<tr><td>ST Path</td><td colspan="2">gpAx[axis_idx]^.lr_mc_control.set_reference_position</td></tr>
<tr><td colspan="3">Commanded, requested and return values</td></tr>
<tr><td>ST Element</td><td colspan="2">.command_w
.request_r
.state_r</td></tr>
<tr><td>Data type</td><td colspan="2">BOOL</td></tr>
<tr><td>Value range</td><td colspan="2">[TRUE, FALSE]</td></tr>
<tr><td colspan="3">Redirection</td></tr>
<tr><td>ST Element</td><td colspan="2">.enable_w</td></tr>
</table>

Reference position to be set		
Description	If the parameter P-AXIS-00278 in the axis parameter list is set to the value “PLC” or “PLC_OFFSET”, the value of this control unit is used to calculate the setting position when acceptance of the reference position is triggered via the HLI (see also the control unit set_reference_position [▶ 62]). In this case, there are options on how to use the value of this control unit. See below:	
	P-AXIS-00278	**Reference position of the axis**
	PLC	Value in this control unit.
	PLC_OFFSET	Encoder position of the drive + value in this control unit.
Data type	MC_CONTROL_SGN32_UNIT, see description of Control Unit [▶ 13]	
Access	PLC reads request_r + state_r and writes command_w + enable_w	
ST Path	gpAx[*axis_idx*]^.lr_mc_control.**refpos_position**	
Commanded, requested and return values		
ST Element	**.command_w** **.request_r** **.state_r**	
Data type	DINT	
Unit	0.1 µm or 10-4°	
Value range	[MIN_SGN32, MAX_SGN32]	
Redirection		
ST Element	**.enable_w**	

Clearing referencing	
Description	If an axis was homed by setting the reference position or by G74, this state can be cleared by using the actual control unit. If an axis has an absolute measuring system, the axis is subsequently considered as not referenced (it can by referenced again by a G74). This switches off the software limit switches temporarily, for example. Any offset activated by the control unit “Setting the reference position” is not cleared. The variable **state_r** indicates whether the axis is currently not referenced.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Special features	As long as the command (command_w) of the control unit is TRUE, the axis is regarded as not referenced and can also be marked as referenced by triggering acceptance of the reference position (see also the control unit set_reference_position [▶ 49]).
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**clear_reference_position**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Measurement signal	
Description	This control unit can transfer the probing signal. When this control unit is used, set the entry kenngr.measure.signal (P-AXIS-00516) to “PLC” or “PLC_TIMESTAMP” in the parameter list of the associated axis.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Characteristics	Edge evaluation: Use the edge parameterised in the axis parameter list in the entry kenngr.mess_neg_flanke (P-AXIS-00518) to accept the measured value.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpAx[*axis_idx*]^.lr_mc_control.**probing_signal**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST Element	**.enable_w**

Measurement value, axis	
Description	If the axis parameter P-AXIS-00257 or the NC command #MEAS […SIGNAL=PLC] changed over the measurement signal source to the control unit probing_signal [▶ 49], activate this control unit in addition to the to transfer the measurement value via the HLI.
Data type	MC_CONTROL_SGN32_UNIT, see description of Control Unit [▶ 13]
Special features	If this control unit is not activated, the actual position of the axis on the rising edge of command_w of the control unit probing_signal [▶ 49] is used as the measurement value. The value assigned to the element command_w of the probing_position control unit is transferred to the decoder without any changes and is then available for further calculations.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**probing_position**
Commanded, requested and return values	
ST element	**.command_w** **.request_r** **.state_r**
Data type	DINT
Unit	0,1 µm bzw. 10^{-4} °
Value range	[MIN_SGN32, MAX_SGN32]
Redirection	
ST element	**.enable_w**

Deactivating an axis (park)	
Description	Axis-specific deactivation of an axis. With a deactivated axis the following actions are not executed by the NC kernel: Error monitoring: errors signalled by the drive are not indicated by the CNC. HLI control bits are not transferred to the drive. A CNC reset does not trigger a drive reset. If you try to move a parked axis the error message P-ERR-70265 is issued. No actions are executed in the drive.
Data type	MC_CONTROL_BOOL_UNIT, see description of Control Unit [▶ 13]
Special features	**PROFIDRIVE drives:** When this control unit is active, the bit 0x80 in set in the control word 2 (STW2).
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**deactivate_axis**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = axis is deactivated, FALSE = axis is active (default mode)]
Redirection	
ST element	**.enable_w**

Timestamp	
Description	This control unit transfers the timestamp of a digital input terminal to the CNC via the HLI.
Data type	MC_CONTROL_SGN64_UNIT, see description of Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpAx[axis_idx]^.lr_mc_control.**timestamp**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	LINT
Unit	[ns]
Value range	[MIN_SGN64, MAX_SGN64]
Redirection	
ST Element	**.enable_w**
Special features	If the timestamp is used for the function “Measure with distributed clocks timestamp”, the probing_signal control unit [▶ 65] must also be enabled. Available as of CNC Build V3.01.3079.28

2.3.6.1 Commanding axis couplings

Commanding axis couplings	
Description	This control unit defines axis couplings for each axis. The motion of the axis can be influenced additionally or exclusively by the motion of other axes. Further details are contained in the documentation FCT-A9.
Data type	MC_CONTROL_AXIS_COUPLING, see description Control Unit [▶ 13]
Access	PLC reads state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**axis_coupling**
Commanded values	
ST Element	**.command_w**
Signal flow	PLC → CNC
Data type	HLI_COMMAND_AXIS_COUPLING [▶ 68]
Access	PLC writes
Return value	
ST Element	**.state_r**
Signal flow	CNC → PLC
Data type	HLI_STATE_AXIS_COUPLING [▶ 68]
Access	PLC reads
Redirection	
ST Element	**.enable_w**

Axis couplings, state	
Description	Indicates whether the axis couplings are active for this axis and if so, which ones.
Signal flow	CNC → PLC
Data type	HLI_STATE_AXIS_COUPLING
ST path	gpAx[*axis_idx*]^.lr_mc_control.axis_coupling**.state_r**
Access	PLC reads
Elements of the data type	
Element	**.desc[ ]**
Data type	ARRAY[0..HLI_AX_COUPLING_MAXIDX] OF HLI_AXIS_COUPLING_DESC [▶ 72]
Access	PLC reads
Element	**.active_r**
Data type	BOOL
Access	PLC reads

Axis couplings, command	
Description	This entry defines a coupling all coupling specification for the axis. The maximum possible number of coupling specifications is defined by the constant HLI_AX_COUPLING_MAXIDX + 1. Further details on the definition of coupling specifications are contained in the documentation @@[FCT-A9].
Signal flow	PLC → CNC
Data type	HLI_COMMAND_AXIS_COUPLING
ST path	gpAx[*axis_idx*]^.lr_mc_control.axis_coupling**.command_w**
Access	PLC writes
Elements of the data type	
Element	**.desc[ ]**
Data type	ARRAY[0..HLI_AX_COUPLING_MAXIDX] OF HLI_AXIS_COUPLING_DESC [▶ 72]
Access	PLC writes
Element	**.semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC accepts the commanded values when this element has the value TRUE. After acceptance, the CNC sets this value to FALSE. The PLC sets this element to TRUE when the commanded values are released for acceptance by the CNC. The commanded values can only be updated by the PLC if this element has the value FALSE.

Description of an axis coupling specification			
Description	This entry defines a coupling all coupling specification for the axis. The maximum number of definable coupling specifications is defined in the constant HLI_AXIS_COUPLING_COUNT. Further details on the definition of the coupling specifications are contained in the documentation [FCT-A9].		
Signal flow	PLC → CNC		
ST Path	gpAx[*axis_idx*]^.lr_mc_control.axis_coupling.command_w**.desc[***idx***]** gpAx[*axis_idx*]^.lr_mc_control.axis_coupling.state_r**.desc[***idx***]**		
Data type	HLI_AXIS_COUPLING_DESC		
Access	Command for axis coupling: PLC writes Status of axis coupling: PLC reads		
Elements of the data type			
ST Element	**.mode**		
Data type	UINT		
Value range	**Constant**	**Value**	**Description**
	HLI_AXIS_COUPLING_INACTIVE	0	Coupling is not active.
	HLI_AXIS_COUPLING_ZERO	1	Coupling factor is zero, used to deactivate an axis.
	HLI_AXIS_COUPLING_DIRECT	2	Coupling factor is 1.
	HLI_AXIS_COUPLING_MIRROR	3	Coupling factor is -1.
	HLI_AXIS_COUPLING_FRACT	4	Coupling factor is a fraction defined by .desc[idx].FractNumerator / .desc[idx].FractDenominator.
ST Element	**.ax_nr**		
Data type	UINT		
Description	Logical axis number of the source axis (the influencing axis). If the axis is to be moved by the NC program when the coupling is active, a coupling specification must be defined with the logical axis number of the axis and the coupling mode HLI_AXIS_COUPLING_DIRECT.		
ST Element	**.fract_num**		
Data type	INT		
Value range	The permissible value range is [-32768, 32767]. A value of 0 in this element has the same effect as the coupling mode HLI_AXIS_COUPLING_INACTIVE.		
Description	Numerator of the coupling factor if the coupling mode is specified as HLI_AXIS_COUPLING_FRACT. This element is not evaluated for all other coupling modes.		

	The maximum permissible value for the coupling factor desc[*idx*].fract_num / .desc[*idx*].fract_denom is defined by the constant HLI_AXIS_COUPLING_FACT_MAX. If this value is exceeded, the error message P-ERR-70397 is issued.
ST Element	**.fract_denom**
Data type	INT
Value range	The permissible range is [-32768, 32767] excluding the 0. A value of 0 in this element results in output of the error message P-ERR-70396.
Description	Denominator of the coupling factor if the coupling mode is specified as HLI_AXIS_COUPLING_FRACT. This element is not evaluated for all other coupling modes. The maximum permissible value for the coupling factor .desc[*idx*].fract_num / .desc[*idx*].fract_denom is defined by the constant HLI_AXIS_COUPLING_FACT_MAX. If this value is exceeded, the error message P-ERR-70397 is issued.

2.3.6.2 Distance control

Commanding distance control	
Description	This control unit can influence the distance control of the axis. This is dependent on whether the distance control function is selected in the axis parameters (see P-AXIS-00328).
Data type	MC_CONTROL_DISTANCE_CONTROL, see description of Control unit with usage check [▶ 14]
Access	PLC reads state_r and writes command_w + enable_w
ST path	gpAx[axis_idx]^.lr_mc_control.distance_control
Flow control of commanded values	
ST element	**.command_semaphor_rw**
Signal flow	PLC → CNC
Data type	BOOL
Peculiarities	**Consumption data item**
Access	CNC accepts the commanded values when this element has the value TRUE. After successful acceptance the CNC sets this value to FALSE. PLC sets this value to TRUE when the commanded value is enabled for acceptance by the CNC. The commanded values can only be updated by the PLC if this element has the value FALSE.
Commanded values	
ST element	**.command_w**
Signal flow	PLC → CNC
Data type	HLI_DISTANCE_CONTROL_COMMAND [▶ 72]
Access	PLC writes
Distance control state	
ST element	**.state_r**
Signal flow	CNC → PLC
Data type	HLI_DISTANCE_CONTROL_STATE [▶ 72]
Access	PLC is reading
Redirection	
ST element	**.enable_w**

Command for distance control	
Description	This entry commands distance control.
Signal flow	PLC → CNC
Data type	HLI_DISTANCE_CONTROL_COMMAND
ST Path	gpAx[*axis_idx*]^.lr_mc_control.distance_control.**command_w**
Access	PLC writes
Elements of the data type	
ST element	**.transition**
Data type	UDINT
Value range/Description	See Table - Permissible transitions to command distance control [▶ 74]
ST element	**.position**
Data type	DINT
Value range	[DINT_MIN, DINT_MAX]
Description	The meaning depends on the commanded transition: HLI_DIST_CTRL_ON: Command position of the workpiece surface (SET_POS) HLI_DIST_CTRL_REF: Reference position of the workpiece surface (REF_POS)

Distance control state	
Description	This entry reads the distance control state.
Signal flow	PLC → CNC
Data type	HLI_DISTANCE_CONTROL_STATE
ST Path	gpAx[*axis_idx*]^.lr_mc_control.distance_control.**state_r**
Access	PLC is reading
Elements of the data type	
Element	**.actual_state**
Data type	UDINT
Access	PLC is reading
Value range/Description	See table: Distance control [▶ 75]
Element	**.actual_position**
Data type	DINT
Unit	0.1 µm or 0.0001°
Access	PLC reads
Description	This data item indicates the current actual position of the workpiece surface detected by the sensing controller.
Characteristics	This data item is entered only if distance control is activated in the axis parameters (see P-AXIS-00328).
Element	**.actual_offset**
Data type	DINT
Unit	0.1 µm or 0.0001°
Access	PLC reads
Description	This data item indicates the current position offset of distance control by which the axis was moved due to deviations between the actual workpiece surface and the specified position (SET_POS). The following applies in stationary state (constant workpiece surface and position offset completely run out): Position offset = SET_POS – actual_position
Characteristics	This data item is entered only if distance control is activated in the axis parameters (see P-AXIS-00328).

Tables for commanding distance control

The tables below list the permissible values to command distance control and the defined states resulting.

Permissible transitions to command the distance control

Transition	Value	Meaning
HLI_DIST_CTRL_OFF	0	Distance control is deactivated. The program switches to the TURNING OFF state in which the position offset is run out. The state then switches automatically to the INACTIVE state.
HLI_DIST_CTRL_ON	1	Distance control is activated. When activated a command position for the workpiece surface must be transferred in the “position” datum. If no absolute encoder is used, distance control must first be referenced.
HLI_DIST_CTRL_FREEZE	2	The current position offset is frozen. Axis adjustment to the actual workpiece surface is ended.
HLI_DIST_CTRL_REF	3	Referencing distance control if no absolute encoder is used. Referencing is only permitted in the INACTIVE state. With this transition a reference position must additionally transferred in the “position” datum.
HLI_DIST_CTRL_ON_CONST_DIST	4	Activating distance control with continuous distance specification. On activation, a set position must be specified. If the distance sensor supplies no absolute values, distance control must be referenced in advance.
HLI_DIST_CTRL_DRYRUN	5	Activate distance control for pure evaluation of data. No axis tracking in case of changes to the workpiece surface. When activated a command position for the workpiece surface must be transferred in the “position” datum. If no absolute encoder is used, distance control must first be referenced.
HLI_DIST_CTRL_CONST_DIST	6	Activate distance control for pure evaluation of data. No axis tracking in case of changes to the workpiece surface. On activation, a set position must be specified. If the distance sensor supplies no absolute values, distance control must be referenced in advance.

Defined states of distance control

State	Value	Meaning
HLI_DIST_CTRL_STATE_INACTIVE	0	Distance control is deactivated. The specified offset (“actual_offset”) is 0.
HLI_DIST_CTRL_STATE_ACTIVE	1	Distance control is active and adjusts the axis to the workpiece surface.
HLI_DIST_CTRL_STATE_FREEZE	2	Distance control is active. The offset (“actual_offset”) is frozen, i.e. the axis is not adjusted to the workpiece surface.
HLI_DIST_CTRL_STATE_TURNING_OFF	3	Distance control was deactivated. The actual active offset (“actual_offset”) is run out. As soon as the offset is 0, an automatic switch to INACTIVE state takes place.
HLI_DIST_CTRL_STATE_ACTIVE_CONST_DIST	4	Distance control is active and adjusts the axis to the workpiece surface. Continuous specification of the set distance of the tool to the workpiece surface.
HLI_DIST_CTRL_STATE_ERROR	5	Distance control is in error state, e.g. due to an incorrect state transition or due to an error in the position controller. Only a transition to TURNING OFF is possible to exit this state.
HLI_DIST_CTRL_STATE_DRYRUN_CONST_DIST	6	Distance control is active, however axis is not adjusted to workpiece surface. This allows the evaluation of data, for example filter effect without feedback from the control system. Continuous commanding of the set distance of the tool to the workpiece surface.
HLI_DIST_CTRL_STATE_DRYRUN_SETPOS	7	Distance control is active, however axis is not adjusted to workpiece surface. This allows the evaluation of data, for example filter effect without feedback from the control system.

2.3.6.3 Monitoring axis positions

Monitoring lower position limit	
Description	The control unit is used to specify the lower limit of a position range which the axis should not exceed.
Data type	MC_CONTROL_SGN32_UNIT, see description of Control Unit [▶ 13]
Access	PLC writes command_w and reads request_r and state_r
ST Path	gpAx[*axis_idx*]^.ipo_mc_control.**dyn_pos_limit_low**
Special features	**Available as of V3.01.3054.05**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** (currently not supported) **.state_r**
Special features	Commanded and return values both have the same data type but different significances.
Data type	DINT
Unit	**.command_w**: 0.1 µm or 10^{-4} ° **.state_r**: no unit
Value range	**.command_w**: [DINT_MIN, DINT_MAX] This is a position value and describes the lower limit of the position range which the axis should not exceed. **.state_r**: See the Monitoring states of position limit [▶ 78] table with descriptions. This is the state of the monitoring process for the lower limit.
Redirection	
ST element	**.enable_w**

Monitoring upper axis position limit	
Description	The control unit is used to set the upper limit of a position range which the axis should not exceed.
Data type	MC_CONTROL_SGN32_UNIT, see description of Control Unit [▶ 13]
Access	PLC writes command_w and reads request_r and state_r
ST Path	gpAx[*axis_idx*]^.ipo_mc_control.**dyn_pos_limit_high**
Peculiarities	**Available as of V3.01.3054.05**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** (not supported) **.state_r**
Peculiarities	Commanded and return values both have the same data type but different significances.
Data type	DINT
Unit	**.command_w**: 0.1 µm or 10^{-4} ° **.state_r**: no unit
Value range	**.command_w**: [DINT_MIN, DINT_MAX] This is a position value and describes the upper limit of the position range which the axis should not exceed. **.state_r**: See Table 2-1 [▶ 77] with descriptions. This is the state of the monitoring process for the upper limit.
Redirection	
ST Element	**.enable_w**

Values for the monitoring state of a position limit

Global constant	**Value**	**Description**
HLI_DYNPL_STATE_INACTIVE	0	The position limit is not active.
HLI_DYNPL_STATE_ACTIVATION	1	This is the transition state after commanding the control unit until monitoring of axis position to the limit is activated.
HLI_DYNPL_STATE_ACTIVE	2	The position limit is active and the axis position limit is monitored.
HLI_DYNPL_STATE_ACTIVE_BRAKING	3	A braking operation was initiated down to standstill to prevent the axis from exceeding the position limit.
HLI_DYNPL_STATE_ACTIVE_BRAKE	4	Deceleration process to maintain the position limit completed, axis is at standstill.

2.3.7 Control commands of a drive

Reading/writing drive data cyclically	
Description	4 elements are provided for writing by the PLC in each interpolation cycle. They can be transferred to the drives via the subordinate drive protocol. The content and action are application-specific (dependent on the drives)
Data type	MC_CONTROL_UNS32_UNIT

Special features	**Currently, data transfer can only be used for SERCOS drives.**
	Therefore, enable transfer of the value in the axis parameter list to the drive, e.g.:
	
	`# write the 4 byte PLC value cyclically`
	`# uns32_1 auf S-0-0815`
	`antr_digital.typ.sercos.mdt[1].ident_nr 0815`
	`antr_digital.typ.sercos.mdt[1].ident_len 4`
	`antr_digital.typ.sercos.mdt[1].nc_ref LR_VAR1_OUT`
	
	`# Cyclically read the 4 byte PLC value`
	`# uns32_3 auf S-0-0819`
	`antr_digital.typ.sercos.at[1].ident_nr 0819`
	`antr_digital.typ.sercos.at[1].ident_len 4`
	`antr_digital.typ.sercos.at[1].nc_ref LR_VAR3_IN`

Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**uns32_**<*i*> where i = [1, 4]
Commanded and requested value	
ST element	**.command_w** **.request_r**
Data type	UDINT
Return value	
ST path	**.state_r**
Data type	UDINT
Special features	These values are also provided at the position controller interface. Siehe Antriebsdaten zyklisch lesen [▶ 38] (gpAx[*axis_idx*]^.lr_state.**uns32_1**)
Redirection	
ST path	**.enable_w**

Operation mode	
Description	Various drive operation modes can be commanded by the PLC (or operator): **SERCOS drives:** With SERCOS drives, this information is sent to bit 8 and bit 9 of the control word. This switches over between main and ancillary drive operation modes. mode_0 corresponds to the lowest operation mode control bit of the drive. Currently, variables mode_0 and mode_1 are used only for **SERCOS drives**. **PROFIDRIVE drives:** **mode_0:** This control unit activates the drive function ‘parking axis’ by setting bit 7 in control word 2. The control unit state_r element indicates whether the function is active in the drive (value of drive status word 2 bit 7). Tracking mode is activated by the CNC internally for a parked axis. **mode_1:** This control unit activates the drive function ‘parking encoder’ by setting bit 14 of the encoder control word. The control unit state_r element indicates whether the function is active in the drive (value of bit 14 in encoder status word). Tracking mode is activated internally in the CNC for a parked encoder. **mode_2:** Currently not in use.
Data type	MC_CONTROL_BOOL_UNIT
Special features	**The elements mode_3 mode_6 are currently not in use.**
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**mode_**<*i*> where i = [0, 6]
Commanded, requested and return values	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Bit set, FALSE]
Redirection	
ST element	**.enable_w**

2.3.8 External axis commanding

Activating external position or velocity command values, axis	
Description	Specifying velocity or position command values by the SPS effective in addition to the interpolator. No monitoring takes place of transferred values for compliance with the dynamic axis limits. To activate this interface, set the parameter P-AXIS-00732 to 1.
Data type	MC_CONTROL_ADD_CMD_VALUE_UNIT, see description of Control Unit [▶ 13]
Special features	When this interface is used, the axis positions in the interpolator and decoder are permanently offset. A repeated synchronisation of axis positions is executed, e.g. at program start, after homing, after measuring or by the NC command #CHANNEL INIT or #SET DEC LR SOLL (old syntax). The command position value is accepted as the start value in m_add_pos_value on a rising edge at the enable input. There is no axis motion here. All further position command changes lead to an axis motion when the control unit is activated. Therefore, always specify a position command value before activating the control unit. If synchronisation should not take place (the offset remains as static offset), set the parameter P-AXIS-00322 to the value 1. In this case no start value is set on the rising edge of the enable input. **Caution:** With gantry axes, no automatic duplication of the additive command value or the master axis occurs on the slave axes. The same additive command value must be applied simultaneously to the master and slave axes in order to achieve a synchronous movement of a gantry axis configuration.
Access	PLC writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.lr_mc_control.**add_cmd_values**
Commanded values	
ST element	**.command_w**
Data type	HLI_ADD_CMD_VALUE
Activation	
ST element	**.enable_w**

External command value, axis	
Description	Transferring additional position or velocity command values. On activation both values are effective at the same time.
Data type	HLI_ADD_CMD_VALUE
ST path	gpAx[*axis_idx*]^.lr_mc_control.add_cmd_values.**command_w**
Data structure elements	
ST element	**.m_add_pos_value**
Special features	Absolute value
Data type	DINT
Unit	0,1 µm
Access	PLC writes
ST element	**.m_add_speed_value**
Data type	DINT
Unit	1 µm/s
Access	PLC writes

2.3.9 Measuring with external measuring hardware

Interface for external measuring hardware	
Description	The CNC informs the PLC about the initialisation, start or end of a measurement run via the external measuring interface. The PLC prepares the external measuring hardware and activates or deactivates it. In order to use a switchable measuring probe, EXT_PROBE_WITH_DRIVE must be selected, otherwise PLC_EXT_LATCH_CTRL. (see P-AXIS-00516 or [PROG//Extended programming])
Data type	LcControlExtLatchControl
Special features	This interface is only used to control the activation and deactivation of external measuring hardware. • PLC_EXT_LATCH_CTRL: The CNC receives information about the measured value or the detected probing position via the control units probing_signal [▶ 65] or probing_position [▶ 65]. • EXT_PROBE_WITH_DRIVE: Probing position via drive interface
Access	PLC writes please_rw + done_w
ST path	gpAx[*axis_idx*]^.**ext_latch_control**
Command	
ST element	**.please_rw**
Description	When please_rw is set, the CNC signals to the PLC to prepare, activate or deactivate the external measuring hardware since a measurement run was started, aborted or terminated.
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	The CNC refreshes the data of the measuring interface only if this element is FALSE. After a refresh, the CNC sets this element to TRUE after the element done_w was set to FALSE. The PLC reads the data of the measuring interface if this element has the value TRUE. After the data is transferred, the PLC sets the value to FALSE.

Parameter		
ST element	**.ext_latch_order_r**	
Description	The CNC signals to the PLC in this datum the required measurement parameters such as activate/deactivate the measuring function or the number of the measurement input.	
Data type	HLI_EXT_LATCH_ORDER	
Access	PLC is reading	
Acknowledgement		
ST element	**.done_w**	
Description	By setting the element done_w to TRUE, the PLC signals to the CNC that the measuring job was executed. The measuring hardware is then ready, activated or deactivated.	
Data type	BOOL	
Value range	[TRUE, FALSE]	
Special features	**Consumption data item**	
Access	The PLC sets the value to TRUE if the measuring order was executed. The CNC sets the value to a FALSE before a new command.	
Measuring order data		
Description	In this parameter, the CNC signals to the PLC whether the measuring hardware is to be prepared, activated or deactivated. In addition, the CNC informs the PLC of the measurement channel to be used and the relevant measurement edge.	
Data type	HLI_EXT_LATCH_ORDER	
ST path	gpAx[*axis_idx*]^.ext_latch_control.**ext_latch_order_r**	
Access	PLC is reading	
Measuring order identifier		
Description	ID of measuring order to be executed.	
ST element	**.order_id**	
Data type	UDINT	
Value range	**Constant**	**Meaning**
	HLI_EXT_LATCH_PREPARE_PROBE	The measuring hardware must be prepared (only EXT_PROBE_WITH_DRIVE)
	HLI_EXT_LATCH_ENABLE_PROBE	Start of measurement run, the measuring hardware must be activated.
	HLI_EXT_LATCH_DISABLE_PROBE	The measurement run was terminated or aborted by reset. The measuring hardware must be again deactivated/switched off
Access	PLC is reading	

Number of probing input		
Description	In this data item, the CNC signals to the PLC the input to be used for measuring (see P-AXIS-00517)	
ST element	**.input**	
Data type	DINT	
Value range	Probing input 1 – 255 Probing input 1-2 (only EXT_PROBE_WITH_DRIVE)	
Access	PLC is reading	
Relevant measurement edge		
Description	In this data item, the CNC signals to the PLC the relevant measurement edge (rising/falling) to be used for measuring (see P-AXIS-00518)	
ST element	**.edge**	
Data type	UDINT	
Value range	**Constant**	**Meaning**
	HLI_MEAS_SIGNAL_LOW_ACTIVE	The measured value is to be detected on the falling edge.
	HLI_MEAS_SIGNAL_HIGH_ACTIVE	The measured value is to be detected on the rising edge.
Access	PLC is reading	

Signal profile with PLC_EXT_LATCH_CTRL

The block diagram below shows an example of the signal profile of the control units ext_latch_control, probing_signal [▶ 65] and probing_position [▶ 65] during a measurement process with external hardware:

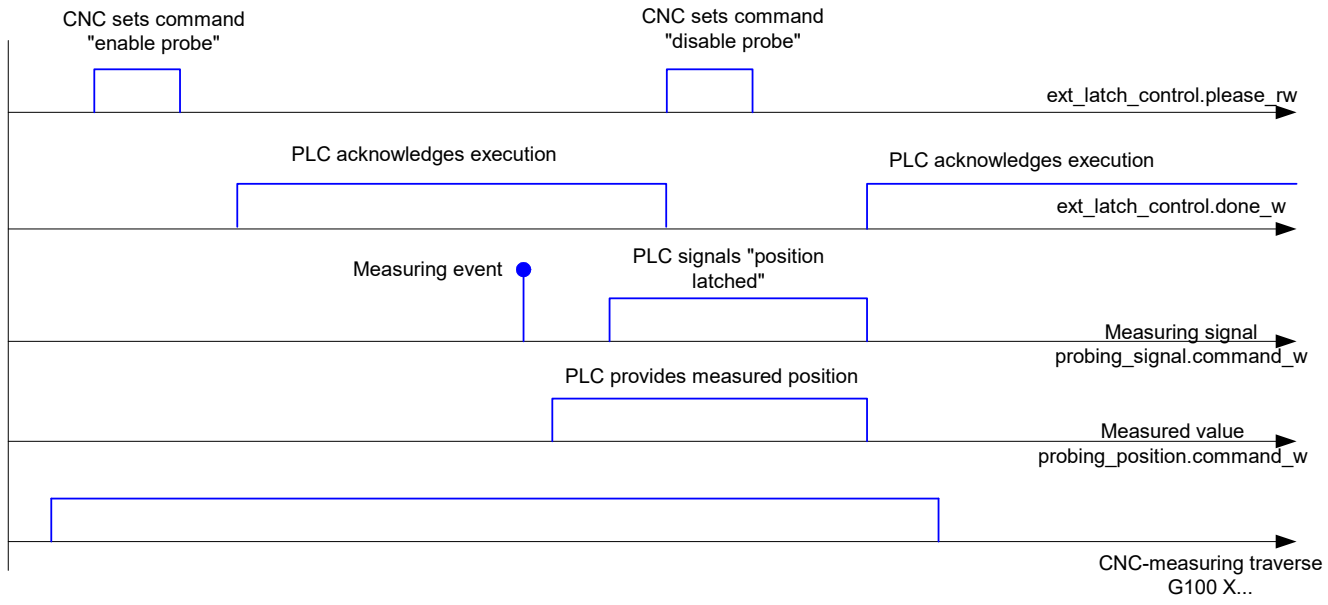


Fig. 11: Example of signal profile of measurement with external hardware

Signal profile with EXT_PROBE_WITH_DRIVE

The block diagram below shows an example of a measurement process using switchable external hardware:

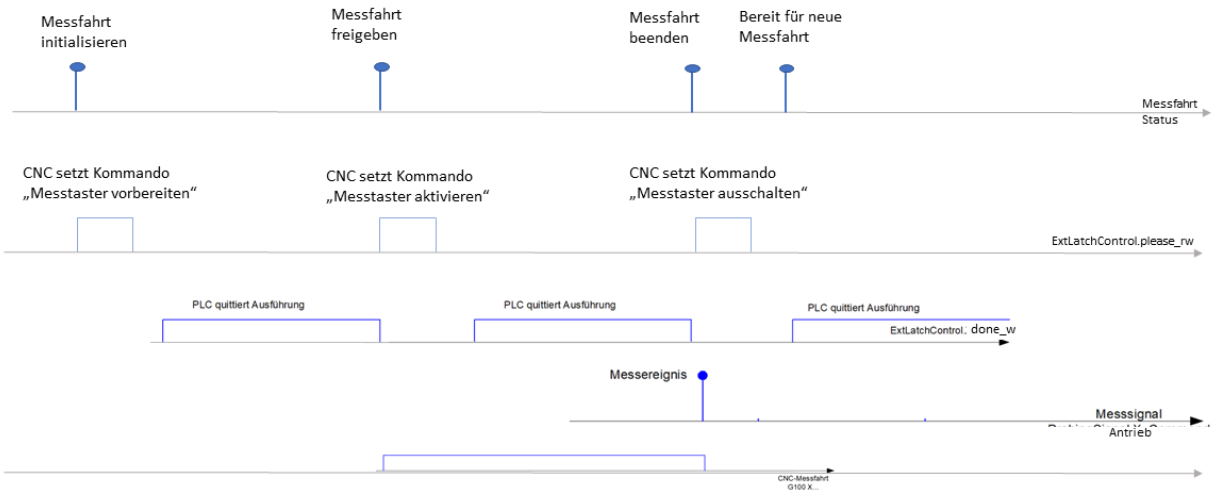


Fig. 12: Signal profile with switchable external hardware

Notice

When the external measuring interface is used, the probing signal of the probing_signal control unit is not dependent on the relevant measurement edge P-AXIS-00518. A positive edge always signals the successful detection of a measured value in the external measuring hardware.

Notice

If the control unit probing_position is not enabled when the measurement event occurs, the CNC uses the actual position of the axis as measurement value at the time stamp of the probing signal occurrence.

3 Spindle

3.1 Introduction

A spindle is an axis with extended properties. At the CNC end, a spindle is mapped by means of a separate motion controller (interpolator).

This means that the spindle axis can be moved not only by the NC program, motion jobs can also be generated at any time by the PLC.

3.2 Description of the spindle-specific interface

3.2.1 Speeds of a spindle

Speed setpoint	
Description	Current speed setpoint of the spindle
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**active_rev_r**
Data type	DINT
Unit	[0.001°/s]
Access	PLC is reading
Special features	

Actual speed	
Description	Current actual speed of the spindle
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state**.current_rev_r**
Data type	DINT
Unit	[0.001°/s]
Access	PLC is reading
Special features	

Speed programmed	
Description	Speed setpoint of the spindle programmed via M03, M04 or S in the NC program
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**cmd_rev_r**
Data type	DINT
Unit	[0.001°/s]
Access	PLC is reading
Special features	

3.2.2 Positions of a spindle

Target position	
Description	Target position when positioning with M19
Signal flow	CNC => PLC
ST path	gpAx[*axis_idx*]^.ipo_state.spindle.**cmd_position_r**
Data type	DINT
Unit	10^{-4} °
Access	PLC reads

Actual position	
Description	Current actual position when positioning with M19
Signal flow	CNC => PLC
ST path	gpAx[*axis_idx*]^.ipo_state.spindle.**act_position_r**
Data type	DINT
Unit	10^{-4} °
Access	PLC reads

3.2.3 Status information of a spindle

Operating state		
Description	Current operating state of the spindle	
Signal flow	CNC → PLC	
ST path	gpAx[*axis_idx*]^.ipo_state.spindle**.mode_r**	
Data type	UDINT	
Value range	**Value**	**Meaning**
	1	M05 active
	2	M03 active
	4	M04 active
	8	M19 active
	16	Superimposed motion active (PLCopen)
	32	RPF active
Access	PLC reads	

Speed monitoring active	
Description	Speed monitoring is activated for the spindle.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**rev_control_aktiv_r**
Data type	BOOL
Value range	[TRUE = Speed monitoring is active, FALSE]
Access	PLC reads
Special features	If the axis is not a spindle, the value is undefined.

Nominal speed reached	
Description	The spindle has reached the programmed speed setpoint.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**rev_erreicht_r**
Data type	BOOL
Value range	[TRUE = Speed setpoint reached, FALSE]
Access	PLC reads
Special features	If the axis is not a spindle, the value is undefined.

Spindle stopped	
Description	The spindle speed is lower than the value of vb_min_null (speed = 0) contained in the axis parameter list.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**rev_null_r**
Data type	BOOL
Value range	[TRUE = Spindle stopped, FALSE]
Access	PLC reads
Special features	If the axis is not a spindle, the value is undefined.

Speed monitoring invalid	
Description	If the speed of rotation of the spindle exceeds the limit speed of the position sensor of the spindle, speed monitoring is deactivated. The limit speed for the position measuring system is defined in the axis parameter list by parameter vb_regelgrenze.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.lr_state.**rev_control_invalid_r**
Data type	BOOL
Value range	[TRUE = Speed monitoring invalid, FALSE]
Access	PLC reads
Special features	If the axis is not a spindle, the value is undefined.

Distance to go	
Description	Distance to go when positioning with M19
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.ipo_state.spindle**.dist_to_go_r**
Data type	DINT
Unit	10^{-4} °
Access	PLC reads

Spindle command error	
Description	A command sent to the spindle was not executed due to error.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.ipo_state.**spindle_order_error_r**
Data type	BOOL
Value range	[TRUE = Command not executed, FALSE]
Access	PLC reads
Special features	Valid only in conjunction with external spindle command by the PLC

3.2.4 Control commands of a spindle

Spindle stop at program end	
Description	If this element is set to TRUE at end of the program, the spindle is stopped.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**spdl_stop_at_prog_end**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Spindle is stopped at end of program, FALSE]
Redirection	
ST element	**.enable_w**

Spindle reset	
Description	Commanding a reset for the spindle.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**spdl_reset**
Commanded and requested Value	
ST element	**.command_w** **.request_r**
Data type	BOOL
Value range	[TRUE Spindle reset commanded, FALSE]
Return value	
ST element	**.state_r**
Data type	BOOL
Value range	[TRUE = Spindle reset is executed, FALSE]
Redirection	
ST element	**.enable_w**

Spindle error	
Description	Sets the spindle to error state
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**spdl_error**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Emergency stop, spindle	
Description	**Commanding this control unit is only effective if the axis is a spindle.** If this element is set to active for emergency stop (TRUE), a speed setpoint of zero is output. The axis is stopped at the emergency stop deceleration which is parameterised. This type of motion influence has maximum priority.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpAx[*axis_idx*]^.ipo_mc_control.**emergency_stop**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Emergency stop active, FALSE]
Redirection	
ST element	**.enable_w**

3.2.5 External spindle command

3.2.5.1 Control unit of external spindle command

External spindle command	
Description	External spindle command. Commands, e.g. spindle stop or spindle positioning, can be issued to the spindle by this control unit over the HLI interface. Enter the other parameters of the command sent to the spindle in the structure HLI_EXT_TO_IPO_DATA.
Data type	MC_CONTROL_EXT_TO_IPO
Commanded values	
ST path	gpAx[*axis_idx*]^.ipo_mc_control.ext_to_ipo.**command_w**
Signal flow	PLC → CNC
Data type	HLI_EXT_TO_IPO_DATA [▶ 94]
Access	PLC writes
Flow control commanded data	
ST path	gpAx[*axis_idx*]^.ipo_mc_control.ext_to_ipo.**command_semaphor_rw**
Signal flow	PLC → CNC
Data type	BOOL
Value range	[TRUE = Data is valid, FALSE = Data was accepted]
Special features	**Consumption data item**
Access	CNC accepts the commanded values when this element has the value TRUE. After acceptance, the CNC sets this value to FALSE. The PLC sets this element to TRUE when the commanded values are released for acceptance by the CNC. The commanded values can only be updated by the PLC if this element has the value FALSE.

3.2.5.2 User data for external spindle commands

The parameters for external spindle commands must be saved in the structure elements described below. Not all structure elements need to be completed dependent on the type of command for the spindle.

Programmed block feed	
Description	Programmed block feed
Signal flow	PLC → CNC
ST path	gpAx[*axis_idx*]^.ipo_mc_control.ext_to_ipo.command_w.**vb_prog**
Data type	DINT
Unit	10^{-3} °/s
Access	PLC writes

G and M functions			
Description	Bit-encoded spindle parameters		
Signal flow	PLC → CNC		
ST path	gpAx[*axis_idx*]^.ipo_mc_control.ext_to_ipo.command_w.**geo_gm_fkt**		
Data type	UDINT		
Value range	Only the following 2 bits are currently used:		
	Constant	**Value**	**Description**
	HLI_OPTIM_RICHTEN	0x00000010L	Dress optimised rotary axis. When positioning the spindle, the target position is approached by the shortest possible path.
	HLI_ABSOLUT	0x00000100L	Position information absolute
Access	PLC writes		

Motion path	
Description	Motion path (relative or absolute) when positioning the spindle with M19.
Signal flow	PLC → CNC
ST path	gpAx[*axis_idx*]^.ipo_mc_control.ext_to_ipo.command_w.**fahrweg**
Data type	DINT
Unit	10^{-4} °
Access	PLC writes

Type of spindle commands			
Description	The type of spindle commands is defined with this element.		
Signal flow	PLC → CNC		
ST path	gpAx[*axis_idx*]^.ipo_mc_control.ext_to_ipo.command_w.**satz_typ**		
Data type	UINT		
Value range	**Value**	**Constant**	**Meaning**
	1	HLI_NC_MOVE_LIN	Linear interpolation
	7	HLI_NC_MOVREF	Homing
	16	HLI_NC_MOVE_ENDLOS	Endless rotation
	20	HLI_NC_GETRIEBE	Spindle gear changes
	29	HLI_NC_SPINDEL_STOP	Stopping the spindle from endless rotation
	30	HLI_NC_SUPER_IMPOSED	accordingly PLCopen MC_SuperImposed
	31	HLI_NC_TABLE_SELECT	Selection of a table according to PLCopen MC_CamTableSelect
	32	HLI_NC_CAM_IN	accordingly PLCopen MC_CamIn
	33	HLI_NC_CAM_OUT	accordingly PLCopen MC_CamOut
	34	HLI_NC_GEAR_IN	accordingly PLCopen MC_GearIn
	35	HLI_NC_GEAR_OUT	accordingly PLCopen MC_GearOut
	36	HLI_NC_PHASING	accordingly PLCopen MC_Phasing
	38	HLI_NC_TOUCH_PROBE	accordingly PLCopen MC_TouchProbe
	39	HLI_NC_ABORT_TRIGGER	accordingly PLCopen MC_AbortTrigger
Access	PLC writes		

<table>
<tr><th colspan="4">Direction of spindle rotation</th></tr>
<tr><td>Description</td><td colspan="3">Defining the direction of spindle rotation</td></tr>
<tr><td>Signal flow</td><td colspan="3">PLC → CNC</td></tr>
<tr><td>ST path</td><td colspan="3">gpAx[axis_idx]^.ipo_mc_control.ext_to_ipo.command_w.dreh_info</td></tr>
<tr><td>Data type</td><td colspan="3">UINT</td></tr>
<tr><td rowspan="4">Value range</td><td colspan="3">The following values apply when the spindle moves:</td></tr>
<tr><td>Value</td><td>Constant</td><td>Meaning</td></tr>
<tr><td>0x0000</td><td>---</td><td>Negative direction of rotation</td></tr>
<tr><td>0x0001</td><td>HLI_SPDL_POS_DREHR</td><td>Positive direction of rotation</td></tr>
<tr><td>Access</td><td colspan="3">PLC writes</td></tr>
</table>

4 Channel

4.1 Introduction

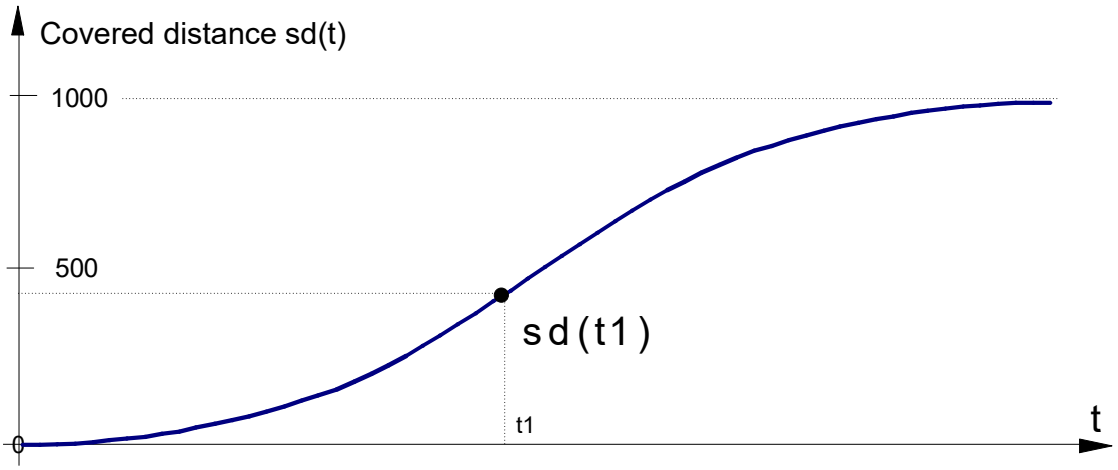
The command variables required for a machine motion are generated within a channel. In this case, input information which may be an instruction from an NC program, a manual motion block or an incremental motion command is converted into command positions for the machine drives. In order to generate a defined relative motion between tool and workpiece, command variables must be generated and coordinated for the machine axes involved. In this context, we refer to axes with a channel relationship. The number of axes controlled in a channel and the type of axes – translatory or rotary – are dependent on the machine concept.

For example, in the case of five-axis milling, three translatory and two rotary axes are controlled in one channel. Today, some machines offer the function of controlled their axes in several channels; an automatic multi-slide lathe is a typical example of this. The specific path motions on these machines are specified in the program for the individual channels and the corresponding command variables are generated independently of each other. However, it is possible to incorporate synchronisation points in each of NC programs to coordinate the channels.

Controls allow these machine concepts so that the generation of command variables is organised in separate channels. Accordingly, a separate NC program is started on each channel. Cross-channel synchronisation is executed either using NC commands or via the PLC.

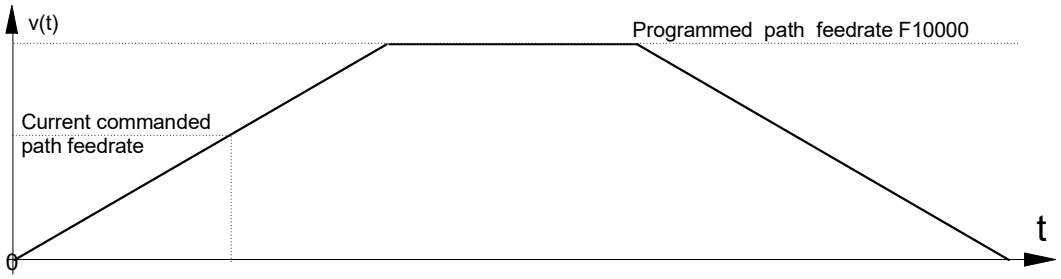
4.2 Description of the channel-specific interface

4.2.1 Status information of a channel

Covered block motion path	
Description	Part of the path motion traversed in the current block in relation to the total path. This status datum contains the current block position referred to the path distance in space in the motion block in per mil sd(t). 
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**covered_distance_r**
Data type	DINT
Unit	0.1 %
Access	PLC is reading
Special features	If a main axis participates in the motion, the covered path motion is in relation to the block path of the first three axes. If no main axis participates in the motion, the covered path motion is the position lag with the longest motion time in relation to the block path.

Currently covered path in the NC program (PCS)	
Description	Reads the current distance covered in the NC program since program start or since the last # DISTANCE PROG START CLEAR NC command. The calculation is based on the current position in the current NC block.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**dist_prog_start**
Data type	UDINT (* LREAL)
Unit	0.1 µm
Access	PLC is reading
Special features	* As of CNC Build V3.1.3104.01 the data element is provided in LREAL forma.

Line counter, NC program	
Description	The datum indicates the NC program line which is the source of the command just processed by the interpolator. The value is derived from the number of NC program lines which the decoder has read since the NC program started. All the lines read the decoder are counted, i.e. repeatedly read lines, empty and comment lines. All commands to the interpolator resulting from decoding a NC program line are assigned to the associated line counter.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.block_count_r
Data type	UDINT
Access	PLC is reading

Programmed path feed	
Description	Path feed that was was programmed by the F word in the NC program.. e.g. using F10000 
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**command_feed_r**
Data type	DINT
Unit	1 µm/s
Access	PLC is reading

Path deed programmed taking into account real-time influences	
Description	Path feed was programmed in the NC program<value> Weighted by the current real-time influences such as override.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**command_feed_active_r**
Data type	DINT
Unit	1 µm/s
Access	PLC reads

Current path feed	
Description	Current path feed during interpolation. This value is displayed with sign when the external path speed (extern_command_speed_valid [▶ 142] control unit) is active
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**active_feed_r**
Data type	DINT
Unit	1 µm/s
Access	PLC is reading

Due to the architecture of the NC kernel in which various components operate asynchronously to each other, certain status information may be provided several times by the various control components.

For example, the program_end_r [▶ 101] signal in the decoder status flag means that the decoder has completed program decoding whereas the actual interpolation by the path interpolator may not have finished. The end of interpolation of the path axes is indicated by the program_end_r [▶ 101] signal in the status data of the path interpolator.

End of program reached	
Description	This status information indicates that the **interpolator** has reached program end or that no NC program is currently under in execution.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**program_end_r**
Data type	BOOL
Value range	[TRUE = End of program reached, FALSE]
Access	PLC is reading

End of program reached	
Description	**Decoder** has reached program end.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.decoder_state.**program_end_r**
Data type	BOOL
Value range	[TRUE = End of program reached, FALSE]
Access	PLC is reading
Special features	**Caution**: The machine cannot move any further since the interpolation signal is relevant for machine motion.

End of program reached	
Description	**Look Ahead function** has reached program end
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bavo_state.**program_end_r**
Data type	BOOL
Value range	[TRUE = End of program reached, FALSE]
Access	PLC is reading
Special features	Required for diagnosis only.

Stop condition	
Description	Displays the condition why the current motion was stopped.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**stop_conditions_r**
Data type	DINT
Value range	See Value range of stop conditions [▶ 103] with explanations.
Access	PLC is reading

Value range of stop conditions

Constant in PLC	Value	Explanation
HLI_SC_FEEDHOLD	0x0001	Path feed stop
HLI_SC_VFG	0x0002	No axis-specific feed enable.
HLI_SC_SINGLE_BLOCK	0x0004	Single step mode active.
HLI_SC_M00_OR_M01	0x0010	M00 (programmed stop), M01 (optional stop) is active.
HLI_SC_PLC_ACKNOWLEDGE	0x0020	Stop occurs due to waiting for an acknowledgement from the SPS. This may occur as a result of the output of M or H technology functions but is not restricted to them alone.
HLI_SC_OVERRIDE_ZERO	0x0040	Override = 0.
HLI_SC_OVERRIDE_RAPID_ZERO	0x0080	Override = 0 with rapid traverse blocks
HLI_SC_DELAY_TIME	0x0200	Dwell time.
HLI_SC_CHANNEL_SYNC	0x0800	Channel synchronisation is active.
HLI_SC_IPO_INPUT_EMPTY	0x1000	Input FIFO of the interpolation is empty.
HLI_SC_IPO_INPUT_DISABLED	0x2000	Input of function blocks (e.g. motion blocks etc.) disabled.
HLI_SC_WAIT_FOR_AXES	0x8000	Stop occurs due to waiting until a commanded axis swap is completed.
HLI_SC_CHANNEL_ERROR	0x00010000	An error occurred in the channel.
HLI_SC_WAIT_TECHNO_ACK	0x00020000	Waiting for acknowledgement of M/H/ST technology functions.
HLI_SC_W_C_AFTER_COLLISION	0x00040000	After a detected collision, waiting for motion resumption.
HLI_SC_SLOPE_SUPPLY_PROBLEM	0x00080000	Block supply problem (only occurs in conjunction with HSC slope).
HLI_SC_BACK_INTERPOLATION	0x00100000	Back interpolation after tracking mode is active.
HLI_SC_STOP_REVERSIBLE	0x00200000	Stop since M00 (programmed stop) is active. However, the NC program can be processed backwards despite M00 (available as of V3.1.3039.01).
HLI_SC_BREAKPOINT_STOP	0x00400000	Stop after a breakpoint (stop point) is reached; available as of V3.1.3039.01.
HLI_SC_M0_STOP	0x02000000	Stop after an M00 function is reached
HLI_SC_M1_STOP	0x04000000	Stop after an M01 function is reached
HLI_SC_INSERT_STOP_AT_DIST	0x08000000	Stop after an M function inserted by the Control Unit “Inserting stop marks” [▶ 164] is reached.
HLI_SC_DEC_SYN_CHAN_EMPTY	0x10000000	Decoder is waiting for synchronisation. NC channel has no jobs.

Error occurred – awaiting clearance	
Description	An internal error has occurred. The interpolator waits for the error to be cleared.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**wait_error_removal_r**
Data type	BOOL
Value range	[TRUE = Error occurred - interpolator waiting, FALSE]
Access	PLC reads

Error occurred– waiting for external input	
Description	In syntax check mode, the decoder waits after an error for an external input (continue, abort)
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.decoder_state.**wait_after_error_r**
Data type	BOOL
Value range	[TRUE = Error occurred – decoder waiting, FALSE]
Access	PLC reads

Interpolator active	

Description	Machine is to be/is moved.
	If a subroutine with path motions is in process, this status is set.
	Example:
	`N10 G01 F1000 X0 Y0`
	`N20 G01 F1000 Y10 Y20 (block is interpolated)`

Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**interpolation_active_r**
Data type	BOOL
Value range	[TRUE = Machine is to be/is moved, FALSE]
Access	PLC is reading

Technology function acknowledgement	
Description	The interpolator waits for the acknowledgement of a technology function from the PLC.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**wait_techno_acknowledge_r**
Data type	BOOL
Value range	[TRUE = Waiting for acknowledgement of technology function, FALSE]
Access	PLC reads

Enable continuation of motion	
Description	The interpolator waits for an Enable to continue the motion after a stop in single-step mode.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**wait_continue_r**
Data type	BOOL
Value range	[TRUE = Interpolator waiting motion to continue, FALSE]
Access	PLC reads

Dwell time active	

Description	The interpolator waits due to a programmed dwell time (G04).
	If a subroutine with a dwell time is in process, this status is set.
	Example:
	`N10 G01 X10 F1000`
	`N20 #TIME 10 ( Dwell time 10 seconds)`

Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**dwell_time_active_r**
Data type	BOOL
Value range	[TRUE = Interpolator waiting, FALSE]
Access	PLC is reading

Violation detected in workspace monitoring	
Description	This datum reads a violation detected by the workspace monitoring system [FCT-C14].
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**area_mon_violation_detected_r**
Data type	BOOL
Value range	[TRUE = error detected in workspace monitoring, FALSE]
Access	PLC is reading

Axis group in position	
Description	All axes in the axis group have reached their programmed end positions.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**axes_in_position_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Access	PLC reads

Waiting for axis group in position	
Description	The interpolator waits for all axes to be in position in single-block mode, a programmed stop (M00), an optional stop (M01) or a exact stop set (G60/G360).
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**wait_axes_in_position_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Access	PLC is reading

Waiting for requested axis	
Description	The look ahead function waits for a programmed axis request (see [PROG//#CALL AX]) top contain the axis.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bavo_state.**wait_for_axis_r**
Data type	BOOL
Value range	[TRUE = Waiting for requested axis, FALSE]
Access	PLC reads

Block search active	
Description	The interpolator works in block search mode. No axis motion occurs. As long the block search process in the interpolator remains in the HLI_BS_ACTIVE or HLI_BS_WAIT_FOR_PLC_OFF state, this value indicates TRUE.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.block_search_active_r
Data type	BOOL
Value range	[TRUE = active - Interpolator works in block search mode., FALSE]
Access	PLC is reading

Block search active	
Description	Look ahead works in block search mode
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bavo_state.**block_search_active_r**
Data type	BOOL
Value range	[TRUE = active – Look ahead works in block search mode, FALSE]
Access	PLC reads

Block search active	
Description	The decoder works in block search mode
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.decoder_state.**block_search_active_r**
Data type	BOOL
Value range	[TRUE = active - Decoder works in block search mode, FALSE]
Access	PLC reads

Block search, distance from continuation position	
Description	If a NC program is started in block search modus, the NC program is processed in simulative mode (without axis motion) until the continuation position is reached. Block search is then in the HLI_BS_WAIT_FOR_PLC_OFF state and calculates the distance between the actual positions of the axis and the continuation position. If block search is in the HLI_BS_RETURNING_TO_CONTOUR state, this value is refreshed cyclically.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.block_search_path_deviation_r
Data type	UDINT
Unit	0.1 µm
Value range	[0, MAX_SGN32]
Access	PLC is reading

Block search, state		
Description	Indicates the current state of the block search mode in the interpolator.	
Signal flow	CNC → PLC	
ST path	gpCh[*channel_idx*]^.bahn_state.**block_search_state_r**	
Data type	INT	
Value range	**Constant**	**Value**
	HLI_BS_INACTIVE	0
	HLI_BS_WAIT_FOR_PLC_ON	1
	HLI_BS_ACTIVE	2
	HLI_BS_WAIT_FOR_PLC_OFF	3
	HLI_BS_WAIT_RETURN_TO_CONTOUR	4
	HLI_BS_RETURNING_TO_CONTOUR	5
	HLI_BS_WAIT_FOR_CONTINUE_CONTOUR	6
Access	PLC is reading	

Rapid traverse velocity, axes move in the channel	
Description	If the value is TRUE, the path axes move on the programmed path when rapid traverse velocity was specified as path velocity. TRUE is only indicated if at least one axis actually moves.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**rapid_mode_r**
Data type	BOOL
Value range	[TRUE = at least one path axis moves and rapid traverse velocity is specified, FALSE]
Access	PLC reads

Collision detected, wait to continue motion	
Description	Indicates that after a collision is detected, the axis waits for a command to continue the motion.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**wait_continue_after_collision_r**
Data type	BOOL
Value range	[TRUE = after a collision is detected, wait to continue the path, FALSE]
Access	PLC reads

Block number, current path motion	
Description	If the N function [PROG//N function] is used in the active NC program to program NC block numbers, this datum indicates the NC block number of the NC block currently processed in the interpolator in this datum.
Signal flow	CNC → PLC
ST path	gpCh[channel_idx]^.bahn_state.block_nr_r
Data type	UDINT
Value range	[0, MAX_UNS32]
Access	PLC is reading

NC program, file name	
Description	Name of the file containing the active NC program. If the active NC program is a global subroutine, the filename containing the global subroutine is indicated.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**file_name_r.zeichen**
Data type	STRING(HLI_NAME_SIZE)
Access	PLC reads

NC program, file offset	
Description	This indicates the current command processed in the interpolator and the offset at which the related NC program line starts in the NC program file.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**active_fileoffset_r**
Data type	DINT
Unit	Byte
Value range	[0, MAX_SGN32]
Access	PLC reads

NC block, inserted	
Description	Indicates whether an additional NC block was inserted by the control unit during interpolation. Additional NC blocks may be created by functions such as polynomial contouring or tool radius compensation.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**block_inserted_r**
Data type	BOOL
Value range	[TRUE = the control unit inserted an NC block, FALSE]
Access	PLC reads

Manual mode is active, without parallel interpolation	
Description	Indicates whether exclusive manual mode is active. No interpolation is active at the same time.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**g200_active_r**
Data type	BOOL
Value range	[TRUE = Manual mode is active without parallel interpolation, FALSE]
Access	PLC reads

Manual mode active, with parallel interpolation	
Description	Indicates whether superimposed manual mode is active. This means that the setpoints for the axes to be moved are calculated by superimposing the defaults for path interpolation and the mode interface of a particular axis.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**g201_active_r**
Data type	BOOL
Value range	[TRUE = Manual mode is active with parallel interpolation, FALSE]
Access	PLC reads

Measurement process, active	
Description	Indicates that a measurement process was commanded by G100 or G108. The rising edge of the datum indicates that the measurement process was started. The value then remains TRUE until a measurement event was triggered for all the axes participating in the measurement run and for which measurement was activated.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**measure_active_r**
Data type	BOOL
Value range	[TRUE = A measurement process is commanded but the measurement results have not yet been received for all measuring axes, FALSE]
Access	PLC reads

Look ahead distance	
Description	Indicates the look ahead distance up to which the motion was calculated by the interpolator. The distance is referred to program starts or to the position where the last NC command #DISTANCE PROG START CLEAR was executed.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**dist_contour_lah_high** gpCh[*channel_idx*]^.bahn_state. **dist_contour_lah _low**
Data type	UDINT
Unit	0,1 µm
Access	PLC reads
Special features	In the NC this is an integer number which occupies 8 bytes in the memory. At the HLI the number is provided in the form of two 4-byte wide values. The value in dist_prog_start_low represents the 4 lower bytes 0 to 3 and the value in dist_prog_start_high the 4 higher bytes 4 to 7 of the 8-byte value present in the NC kernel. **Available as of Build V3.1.3030.0 and higher**

Delete distance to go, state	
Description	If a command is sent via the delete_distance_to_go control unit [▶ 149], this value is TRUE as long as the NC block which exerts a linear motion to the target position of the next motion block is executed (short cut). It also remains TRUE if the active short cut is itself shortened by a new command. The signal is again reset when the active NC block is no longer related to the command of the delete_distance_to_go control unit [▶ 149]. See functional description [FCT-C28].
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**delete_distance_to_go_active_r**
Data type	BOOL
Value range	[TRUE = linear motion is executed to the target position of the next motion block (short cut), FALSE]
Access	PLC is reading

High-resolution velocity setpoint, state	
Description	The value indicates whether the interface for the external input of velocities [▶ 141] continues to process the inputs at high resolution. Processing the input at high resolution is enabled by the NC command #EDM ON and disabled by programming #EDM OFF. This is used in combination with electrical discharge machining.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**edm_active_r**
Data type	BOOL
Value range	[TRUE = high-resolution processing of input value is active, FALSE = input values are not processed at high resolution]
Access	PLC reads

Time to next motion command containing G01, G02	
Description	Is rapid traverse velocity is specified for the current path motion, this datum indicates the time until the next motion block containing G01 or G02 is active. This time is only calculated and displayed if it is enabled in the channel parameter P-CHAN-00650 of the channel or, alternatively, in the start-up parameter P-STUP-00070 by specifying FCT_LOOK_AHEAD_STANDARD | **FCT_CALC_TIME**. (See Interpolation function table.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**time_to_next_feed_block_r**
Data type	UDINT
Value range	[0, MAX_UNS32]
Access	PLC is reading

Stop, value	
Description	When a reversible stop is programmed, an additional user-specific parameter can be specified: #STOP REVERSIBLE[USR_VAL = ...] That value is indicated in this element as soon as a stop is executed at this position. After resuming the motion, the value is deleted. In this context also see the "Reversible stop" control unit (stop_reversible_level) [▶ 152] and the functional description [FCT-C7// Automatic reversing after stop].
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**stop_reversible_usr_val_r**
Data type	UDINT
Value range	[0, MAX_UNS32]
Access	PLC reads

Example

Reversible stop

Output of the waiting condition identifier and evaluation of the maximum waiting time

Required channel parameter:

```
forward_backward.disable_stop_backward      0
forward_backward.disable_stop_2nd_forward   0
forward_backward.disable_stop_1st_forward   0
```

NC program:

```
%stop_reversible
N01 X0 Y0 Z0
N10 X100
N20 Y100
N30 X0
N40 Y0
N45 #STOP REVERSIBLE[ USR_VAL=500]
N50 X0 Y0 Z0
N60 X100
N70 Y100
N80 X0
N90 Y0
N95 #STOP REVERSIBLE[ USR_VAL=2000]
M30
```

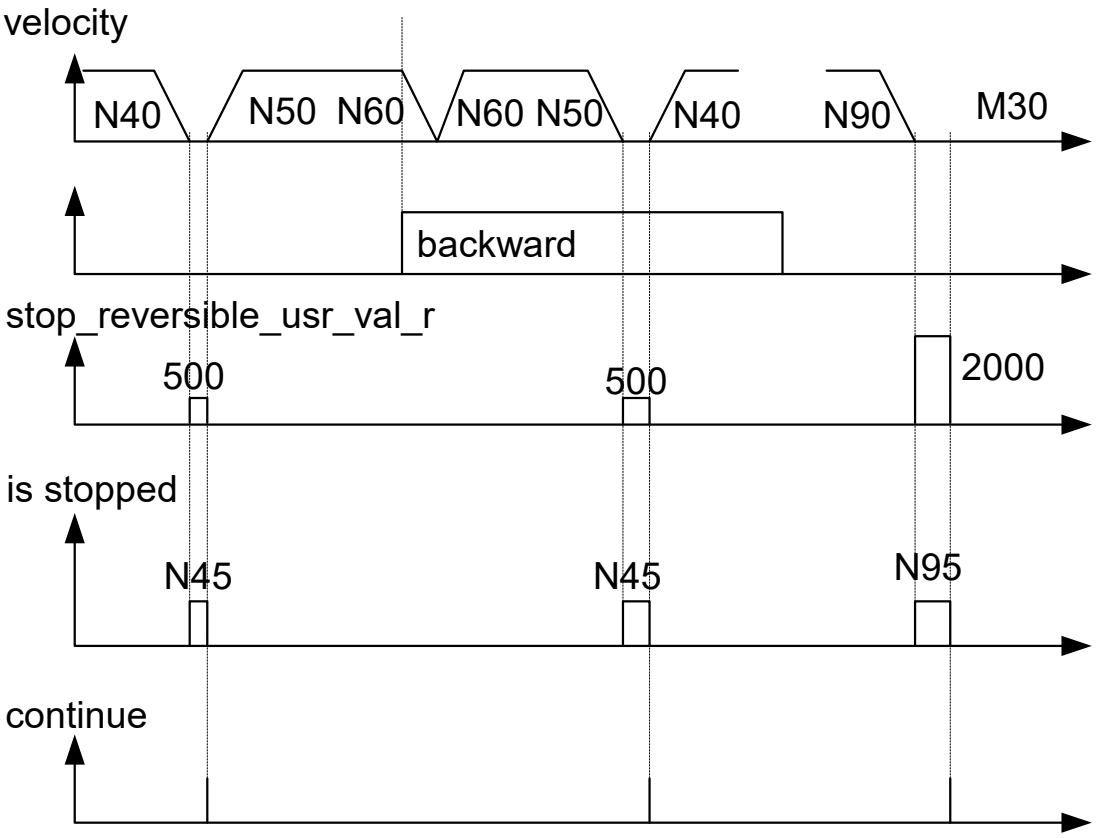


Fig. 13: Stop interaction on backward/forward motion

Distance to edge below limit	
Description	This signals that the distance to the edge defined by the channel parameter P-CHAN-00222 (edge_machining.pre_dist*)*, was undershot. As from this time, motion on the programmed path is effective at the path feed defined by the channel parameter P-CHAN-00223 (edge_machining.pre_feed*)*.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.edge_function_r.**signal_1**
Data type	BOOL
Value range	[TRUE = active – Distance to edge undershoots parametrised limit, FALSE]
Access	PLC reads

Wait time at edge	
Description	The signal indicates that the motion was stopped at the edge and the wait time specified by the channel parameter P-CHAN-00224 (edge_maching.wait_time) is expiring.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.edge_function_r.**signal_2**
Data type	BOOL
Value range	[TRUE = active – wait time active, FALSE]
Access	PLC reads

Distance after edge below limit	
Description	The signal indicates that the tool is moving away from the edge but the distance from the edge is still smaller than the value defined by the channel parameter P-CHAN-00225 (edge_machining.post_dist*)*. The path feed valid for this section is defined by the channel parameter P-CHAN-00226 (edge_machining.post_feed*)*.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.edge_function_r.**signal_3**
Data type	BOOL
Value range	[TRUE = active - distance after edge not reached, FALSE]
Access	PLC reads

Angle between active and next NC block	
Description	Indicates the angle between 2 consecutive NC blocks if both NC blocks are motion blocks containing programmed feed. The indicated value is in the range of [0, 1800000] which corresponds to [0°, 180°].
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.edge_function_r.**angle_end**
Data type	DINT
Unit	10^{-4} °
Value range	[0,10000000]
Access	PLC reads
Special features	If the the next motion block is a rapid traverse block, value 5000000 is output NC block that leads to motion stop, value 6000000 is output. In all other cases the default value 10000000 is indicated.

Velocity at end of current NC block	
Description	Indicates the velocity at transition from currently interpolated to next motion block which results from evaluating the geometrical situation and the velocity of the next block. If the dwell time is programmed or a predictable motion stop is programmed at the transition between the NC blocks caused by the output of a technology function with corresponding synchronisation start in the next block, the value 0 is indicated.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.block_dynamic_r.**vel_end_geo**
Data type	DINT
Unit	1 µm/s
Value range	[0, MAX_SGN32]
Access	PLC reads

Numerator of the real-time loop	
Description	The datum shows the number of passes of the real-time loop in which the current machining is located. If a loop transition is in a forward direction, the datum is incremented by 1 and it is decremented in a backward direction. If more loop passes were processed in the backward direction, a negative numeral is displayed.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**rt_loop_count_r**
Data type	DINT
Access	PLC reads
Special feature	Date available as of CNC Build V3.1.3105.01 and higher

Real-time loop active	
Description	The datum indicates whether current machining is located in a #RT WHILE - #RT ENDWHILE sequence.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**inside_rt_loop_r**
Data type	BOOL
Value range	[TRUE = Machining within #RT WHILE #RT ENDWHILE, FALSE]
Access	PLC is reading
Special feature	Date available as of CNC Build V3.1.3105.01 and higher

Waiting for external command velocity	
Description	The NC channel waits for an external command velocity.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**wait_ext_command_speed_r**
Data type	BOOL
Value range	[TRUE = channel is waiting FALSE = external velocity exists]
Access	PLC is reading

4.2.1.1 Status information on tool orientation

The three vectors

- path tangent vector (tb_vec)
- normal vector to path (tn_vec)
- face normal vector (fn_vec)

form a complete clockwise spatial coordinate system (moving trihedron).

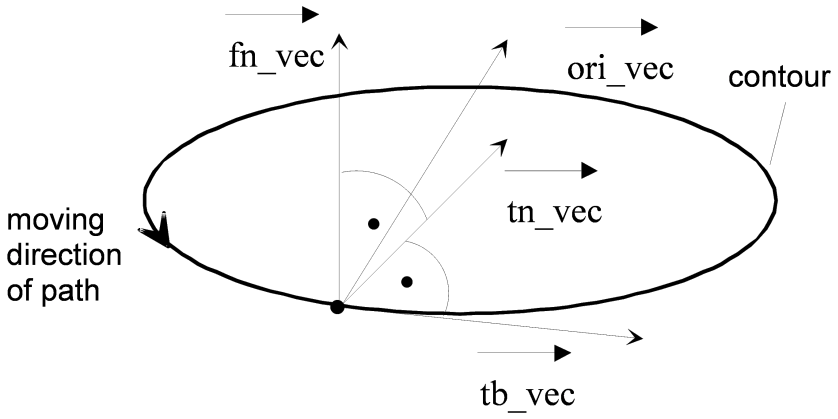


Fig. 14: Vectors of the tool coordinate system

Notice

Configuration of the tool direction vector

When tool geometry compensation is active and when machining with the face of the tool, the result is the tool direction vector ori_vec obtained from the face normal vector fn_vec, the path tangent vector tb_vec and the advance and lateral angle.

The vectors of the tool coordinate system are integrated in the data structure HLI_TOOL_PATH_DISP_DATA. The tables below contain the description of the elements for this data structure:

Tool direction vector	
Description	Components of the tool direction vector and recalculation from face normal vector and path tangent vector (see Figure).
ST Element	gpCh[*channel_idx*]^.bahn_state.tool_path_r**.ori_vec[***vec_idx***]**
Data type	ARRAY [0..HLI_CS_AXES_MAXIDX] OF DINT
Peculiarities	The direction vector is normalised to length 10^6.

Path tangent vector	
Description	Components of the path tangent vector (see Figure).
ST Element	gpCh[*channel_idx*]^.bahn_state.tool_path_r**.tb_vec[***vec_idx***]**
Data type	ARRAY [0..HLI_CS_AXES_MAXIDX] OF DINT
Peculiarities	The direction vector is normalised to length 10^6. The last valid motion direction is retained.

Normal vector to the path tangent	
Description	Components of the resulting vector, cross product of ori_vec × tb_vec, or cross product of fn_vec × tb_vec (see Figure).
ST Element	gpCh[*channel_idx*]^.bahn_state.tool_path_r**.tn_vec[***vec_idx***]**
Data type	ARRAY [0..HLI_CS_AXES_MAXIDX] OF DINT
Peculiarities	The direction vector is normalised to length 10^6.

Auxiliary vector of the complete trihedron, face normal vector	
Description	Components of the resulting vector, cross product of tb_vec × tn_vec, or face normal vector (see Figure).
ST Element	gpCh[*channel_idx*]^.bahn_state.tool_path_r**.fn_vec[***vec_idx***]**
Data type	ARRAY [0..HLI_CS_AXES_MAXIDX] OF DINT
Peculiarities	The direction vector is normalised to length 10^6.

4.2.1.2 Pre-calculated status information

Precalculated data is valid, path	
Description	If a data element in the field is TRUE, path velocity was precalculated for a future point in time and is therefore valid. This is indicated by the same index as precalculated velocity [▶ 119]. FALSE indicates that no value could be calculated for the future point in time. Several points in time were defined by P-CHAN-00324. Times can also be defined by #CHANNEL SET[ESA_TIME<i>=…].
Signal flow	CNC →PLC
ST Path	gpCh[channel_idx]^.bahn_state.**esa_data_valid[ ]**
Data type	ARRAY[0..2] OF BOOL
Value range	TRUE/FALSE
Access	PLC is reading
Special feature	**Available as of Build V3.1.3104.08**

Precalculated velocity, path	
Description	Path velocity at a future point in time. Several points in time were defined by P-CHAN-00324. The index of a configured point in time corresponds with the index of the precalculated path velocity. Times can also be defined by #CHANNEL SET[ESA_TIME<i>=…].
Signal flow	CNC →PLC
ST Path	gpCh[channel_idx]^.bahn_state.**esa_vb[ ]**
Data type	ARRAY[0..2] OF LREAL
Access	PLC is reading
Special feature	**Available as of Build V3.1.3104.08**

4.2.1.3 Status information for velocity limit

Path velocity below limit	
Description	The path velocity undershoots the parametrised limit.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.**speed_limit_detect_r**
Data type	BOOL
Value range	[TRUE = active – Path velocity undershoots the parametrised limit, FALSE]
Access	PLC is reading

4.2.1.3.1 Speed Limit Detect, Look Ahead for velocity limit

General

This function generates a CNC status signal depending on the motion blocks and the current path velocity. It is mainly used for plasma cutting technology to deactivate distance control of the cutting head via the PLC if the speed drops below a certain limit. This occurs, for example

- when decelerating ahead of and
- accelerating after a corner.

Therefore, the critical positions in the motion segment (corner) are defined by a speed limit.

"Speed limit detect" status flag

The “speed limit detect” status flag to the PLC is set if the current path velocity is below the specified limit. The drop in path velocity results from:

- Reduction in velocity at block transition due to a kink in the path contour.
- Reduction of velocity due to override setting.
- Expected M function acknowledgement from PLC at block transition.
- Look-ahead reduces velocity because of inadequate block supply.

Zone

In addition the signal can be set in advance or cleared with a delay at a specified distance (time/distance).

- Advance: The expected velocity at block end undershoots the speed limit, for example due to a geometrical corner. The status flag is set in advance at the specified distance to the expected limit undershoot.
- Delay: The expected velocity at block start already undershoots the speed limit. The status flag is again cleared with a delay at the specified distance to the actual speed overshoot.

Clear

I.e. the status flag is reset when both of the following conditions are met:

- The current path velocity rises above the speed limit.
- The path position is outside the specified time and distance delay.

4.2.1.3.2 Description

Activation

When the function is activated, a CNC status signal is generated according to the control flag P-CHAN-00017 and signals a speed limit undershoot or the detection of a future speed limit undershoot.

Limit value

The speed limit is defined via the percentage weighting (P-CHAN-00089) of the F word in the NC program.

Advance, delay

The expected drop in velocity at block end can be signalled in advance by the parameter P-CHAN-00013. Accordingly, the signal can also be cleared with a delay by the parameter P-CHAN-00012. The parameters P-CHAN-00012 / P-CHAN-00013 can therefore define a type of hysteresis.

Distance, time

The parameter P-CHAN-00018 defines the zone parameter unit as either a distance or time.

The CNC generates the “speed limit detect” status signal if the path position is inside this zone.

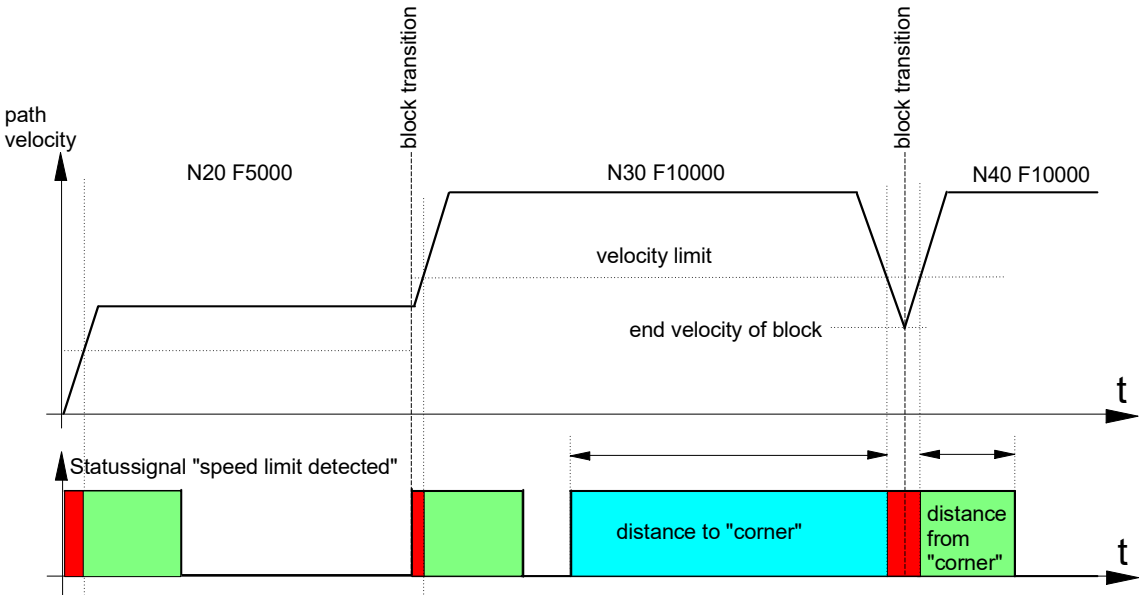


Fig. 15: F Word and status signal “speed limit detected”

Influence of override

The parameter P-CHAN-00155 controls the influence of the speed limit by means of the real-time feed override.

In the default setting (P-CHAN-00155 = 0) the real-time feed override does not influence the speed limit P-CHAN-00089. The “speed limit detect” status signal is also set if the path feedrate weighted by the override drops below the speed limit P-CHAN-00089.

Since override is a user-initiated online influence of speed, the delay/advance of the status signal cannot be evaluated in this case.

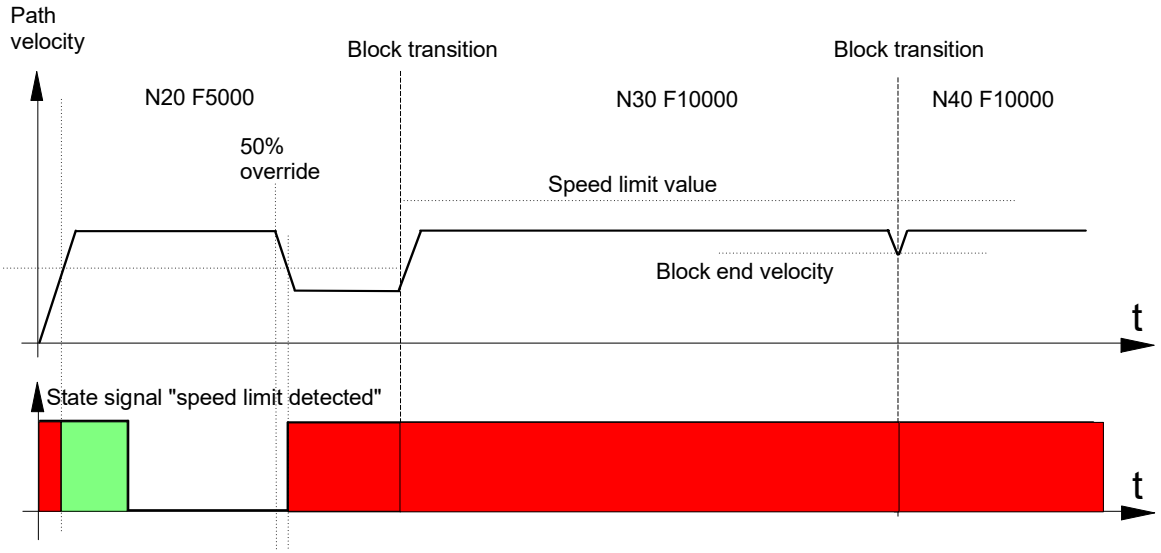


Fig. 16: Timing diagram without override weighting (f_override_weight_v_limit = 0)

When P-CHAN-00155 is set, the parameterised speed limit P-CHAN-00089 is weighted by the override value. This is desirable e.g. for path start-up or run-in.

Note that, with a non-constant programmed feed, the "speed limit detect" signal is activated in each acceleration phase because the speed limit at the start of the block is set to the new value.

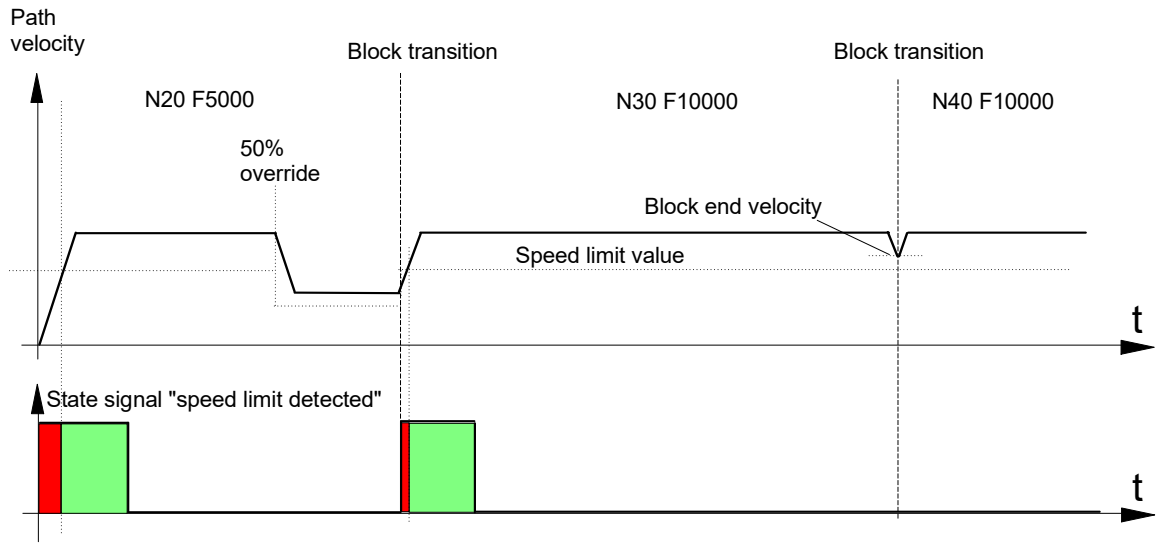


Fig. 17: Timing diagram with override weighting (f_override_weight_v_limit = 1)

Influence of technology functions

The “speed limit detect” status signal is set if the CNC has to stop and wait due to certain types of technology functions or missing PLC synchronisation. In this case, the signal advance or delay related to the set time or distance is also evaluated.

Waiting for PLC acknowledgement

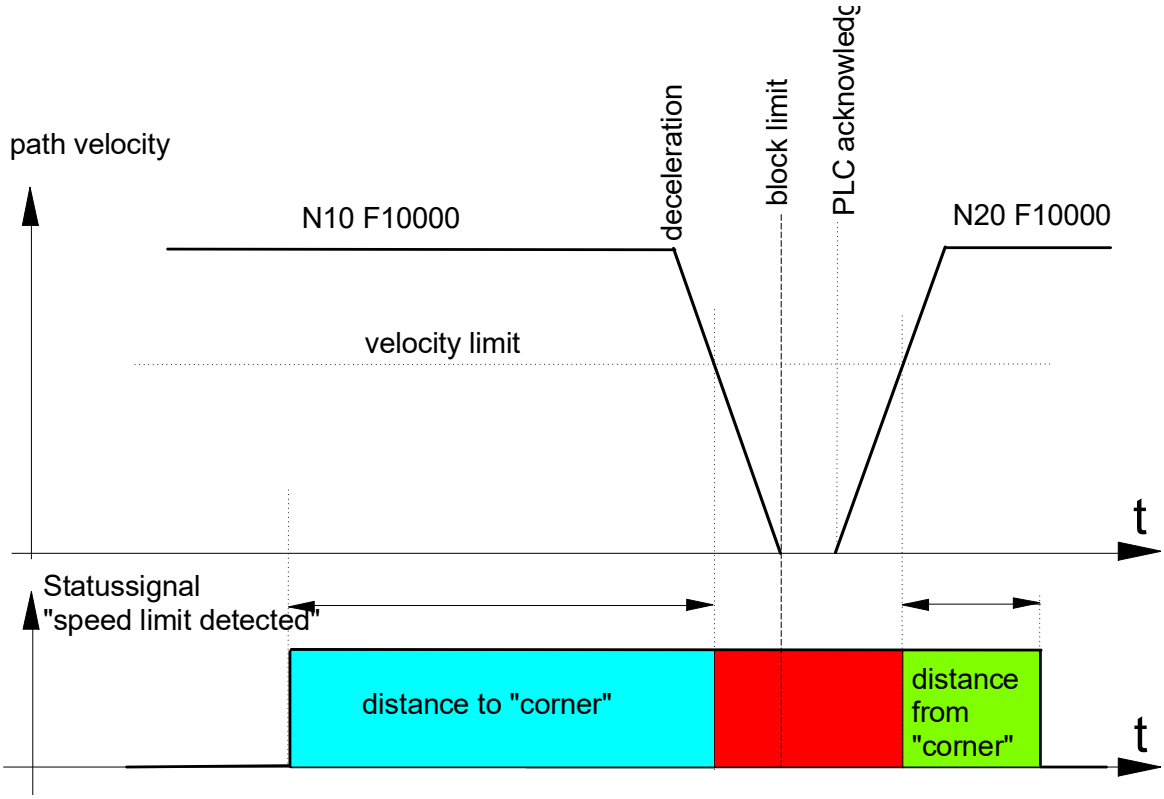


Fig. 18: Missing PLC acknowledgement and "speed limit detected" status signal

With M functions of the MVS_SNS type, later synchronisation or M functions with look-ahead are only stopped if the PLC acknowledgement is missing. If the PLC acknowledgement arrives before the motion, a restart can be executed immediately.

However, the advance signal (advance, distance to “corner”) can still be executed correctly although an actual speed undershoot no longer arrives.

PLC acknowledgement during deceleration

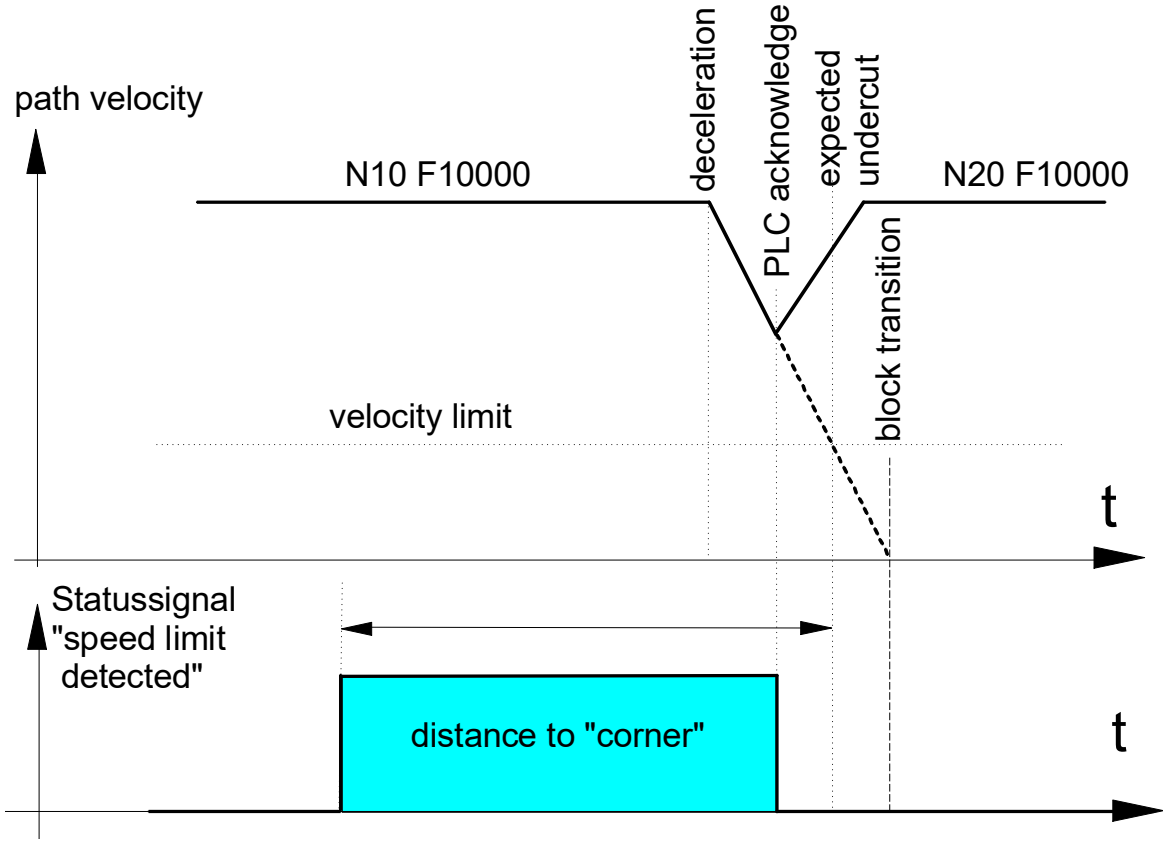


Fig. 19: A restart after PLC acknowledgement resets the “speed limit detect” status signal.

Notice

Each M or H function of the MVS_SVS or MNS_SNS type always results in a motion stop (see also [FCT-C1]).

Interruption of block supply

If the path velocity fluctuates due to short blocks and inadequate block supply, this may cause activation of the “speed limit detect” status signal. In the example below the advance/delay parameters (distance to corner and distance from corner) are disabled for the sake of simplification.

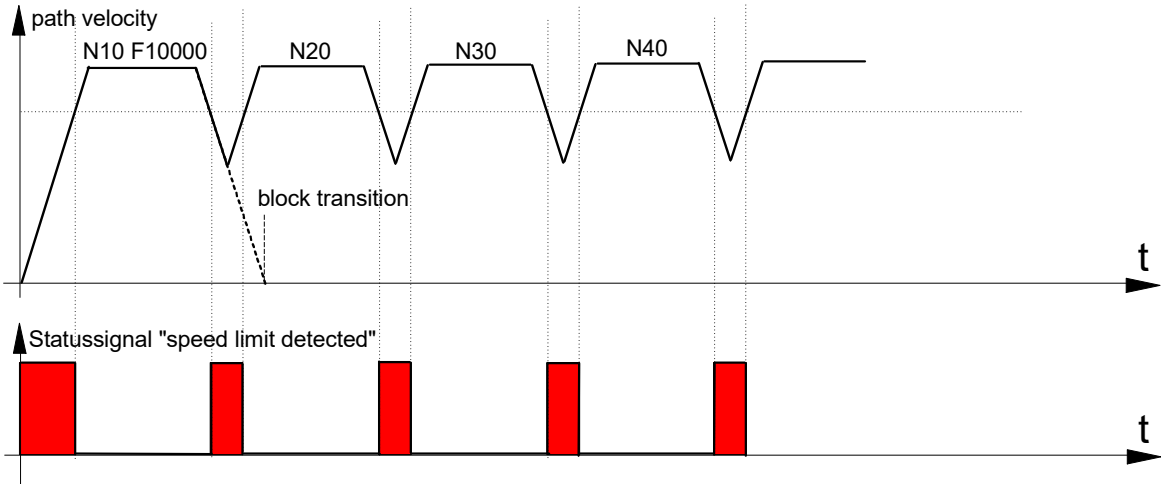


Fig. 20: Inadequate block supply results in the activation of the "speed limit detected" signal.

4.2.1.3.3 Example

The “speed limit detect” status signal is generated depending on the set parameters if the programmed paths cause deceleration along the path and the speed drops below the speed limit, e.g. due to a corner.

Parameter

Excerpt from the channel parameter list [CHAN]:

```
# Speed limit look ahead parameterisation
# =======================================
speed_limit_look_ahead.f_enable                          1
speed_limit_look_ahead.v_limit                         750
speed_limit_look_ahead.f_time                            0
speed_limit_look_ahead.dist_to_corner                10000
speed_limit_look_ahead.dist_from_corner              10000
speed_limit_look_ahead.f_override_weight_v_limit         0
```

Notice

Parameters can also be changed in the NC program by appropriate variables (V.G.SPEED_LIMIT.*) [PROG].

Programing Example

"Speed limit detect” status signal

Speed drop at end of NC block

```
%main
X0 Y0
N10 G01 X50 F5000
N20 X100
N30 X150
N40 X200 (speed drop at end of NC block)
```

```
N50 X250 Y-25
N60 X300 Y-50
M30
```

The parameters listed above and the F word in the NC program result in:

Speed limit = 75% of the programmed velocity

→ v_limit = 3750 mm/min (62500 um/s)

In the NC program example the path velocity drops to 8.562 um/s at block transition N40 -> N50 due to a path kink angle of 30 degrees. This means that the “speed limit detect” status signal is set 1 mm before limit speed undershoot at the block end of N40 and reset 1 mm after limit speed overshoot at the block start of N50.

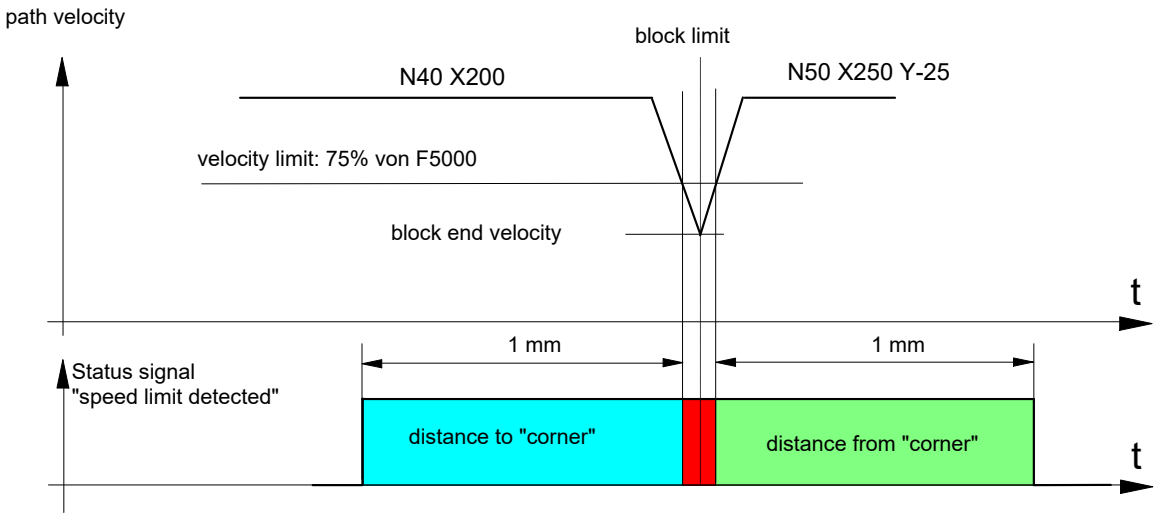


Fig. 21: F word and “speed limit detected” status signal

4.2.2 Control commands of a channel

Skip mode, NC block	
Description	Activates/deactivates skip mode at interpreter level for the NC program. The status of skip mode is only evaluated at the start of the NC program. Switchover during execution of an NC program has no effect.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13] Available as of CNC Build V3.01.3021.1 MC_CONTROL_UNS32_UNIT
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh[*channel_idx*]^.decoder_mc_control.**program_block_ignore**
Commanded, requested and return values	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL or UNS32
Value range	[TRUE = Skip mode NC block ON, FALSE = Skip mode NC block OFF, default: FALSE]
	Available as of CNC Build V3.01.3021.1 for UNS32: 0x0 - Skip mode NC block OFF 0x1 – Skip level 1 0x2 - Skip level 2 0x4 - Skip level 3 0x8 - Skip level 4 0x10 – Skip level 5 0x20 - Skip level 6 0x40 - Skip level 7 0x80 - Skip level 8 0x100 – Skip level 9 0x200 - Skip level 10 Skip levels active simultaneously are enabled by bitwise ORing. Example: Enable all skip levels by setting 0x3FF.
Redirection	
ST element	**.enable_w**

Single-block mode at interpreter level NC block	
Description	Activates/deactivates single-block mode at interpreter level. A restart must be present for each block
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Characteristics	**Not supported**
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh[*channel_idx*]^.decoder_mc_control.**single_block**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = single-block mode ON, FALSE]
Diversion	
ST Element	**.enable_w**

Enable processing of next NC block	
Description	Enable to process the next NC block. This element serves to synchronise the geometry chain with the PLC. For this purpose, the PLC blocks the single-block step enable and thus stops the interpreter.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Special features	**Is currently not supported.**
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.decoder_mc_control.**continue_machining**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Continuation NC program decoding	
Description	Continues decoding an NC program after an error has occurred if syntax check (simulation mode) and interactive step enabling of decoding is activated (decoder parameter, characteristic parameter: syn_chk.interaktiv = 1).
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Special features	**Is currently not supported.**
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.decoder_mc_control.**release_stop**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Stop NC program decoding	
Description	Stops decoding an NC program if interactive enabling of further processing of the NC program was activated in Syntax Check Simulation mode.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Special features	**Is currently not supported.**
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.decoder_mc_control.**dec_stop**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Channel operation mode			
Description	Selects a special channel operation mode, e.g. syntax check or machining time calculation		
Data type	MC_CONTROL_SGN32_UNIT, see description of Control Unit [▶ 13]		
Access	PLC reads request_r + state_r and writes command_w + enable_w		
ST path	gpCh[*channel_idx*]^.decoder_mc_control.**execution_mode**		
Commanded, requested and return values			
ST element	**.command_w** **.request_r** **.state_r**		
Data type	DINT		
Value range	**Value**	**Constant**	**Meaning**
	0x0000	ISG_STANDARD	Normal mode
	0x0001	SV	Block search
	0x0002	SOLLKON	Nominal contour visualisation simulation with output of visualisation data
	0x0802	SOLLKON_SUPRESS_OUT-PUT & SOLLKON	Nominal contour visualisation simulation without output of visualisation data
	0x0004	ON_LINE	Online visualisation simulation
	0x0008	SYNCHK	Syntax check simulation
	0x0010	PROD_TIME	Simulation machining time calculation (in TwinCAT without function)
	0x0020	ONLINE_PROD_TIME	Simulation online machining time calculation
	0x0040	MACHINE_LOCK	Dry run without axis motion
	0x0080	ADD_MDI_BLOCK	Extended manual block mode: the end of a manual block is not evaluated as a program end. It permits the commanding of further manual blocks.
	0x0100	KIN_TRAFO_OFF	Overwrites automatic enable for kinematic transformations by a characteristic parameter defined in the channel parameters (sda_mds*.lis).
	0x1000	BEARB_MODE_SCENE	When SCENE mode is enabled, the output of #SCENE commands is activated on the interface (see also [FCT-C17// Scene contour visualisation]). An additional client is linked to this output via DataFactory / CORBA.
	0x2000	SUPPRESS_TECHNO_OUT-PUT	Without output of technology functions (M/H/T). Set implicitly in connection with syntax check.
	0x10000	SUPPRESS_POSITION_RE-QUEST	Fast program start without position request at program start

	0x20000	SUP-PRESS_PROG_START_INIT	Suppress program start sequence for machining on the belt
Redirection			
ST element	**.enable_w**		

Index of position offset group	
Description	A position offset defines an additional offset. For instance, this takes into consideration different clamping positions of a workpiece in a machine workspace. Position offsets are defined in the position offset data. The position offsets for each axis are specified in a position offset group. A position offset group is selected by the group index. Position offset data is evaluated by the CNC at program start.
Data type	MC_CONTROL_SGN16_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.decoder_mc_control.**clamp_position**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	INT
Value range	[0, 67]
Redirection	
ST element	**.enable_w**

Feedhold ON/OFF type 1	
Description	Channel-specific feedhold. Setting this element to TRUE means immediate ramp-down of the feed rate during interpolation in accordance with the parameterised accelerations to feed rate = 0.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Special features	The channel-specific parameter P-CHAN-00097 can be used to exert additional influence on the parameterised accelerations to be used. Feedhold ON/OFF can also be commanded by the control unit for switching operation modes (see Sec. 8.2 [▶ 190]). A HOLD command results in stopping the channel; a RESUME command cancels the stop command. If the PLC is registered at both control units, pay attention to the following safety note:
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**feedhold**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Feedhold ON, FALSE]
Redirection	
ST element	**.enable_w**

⚠ CAUTION

Stop command to CNC is not executed.

Possible damage to machine.

Similar to other control units, a command from an external application, e.g. an HMI, is indicated in **request_r**. If the PLC operates this control unit and therefore set **enable_w** to TRUE, the command from the external application only becomes effective when **request_r** is written to **command_w**. Here the semaphores must be operated as usual.

This must also be considered if the PLC operates the control unit in order to switch operation modes (see section 8.2 [▶ 190]) and therefore sets the **enable_w** to TRUE A HOLD command sent by this control unit triggers the NC kernel to send a request to activate feedhold. This is again indicated in request_r of the control unit described here and only becomes effective when the PLC copies PLC **request_r** to **command_w**. This also applies to cancelling feedhold.

The figures below depict these functions.

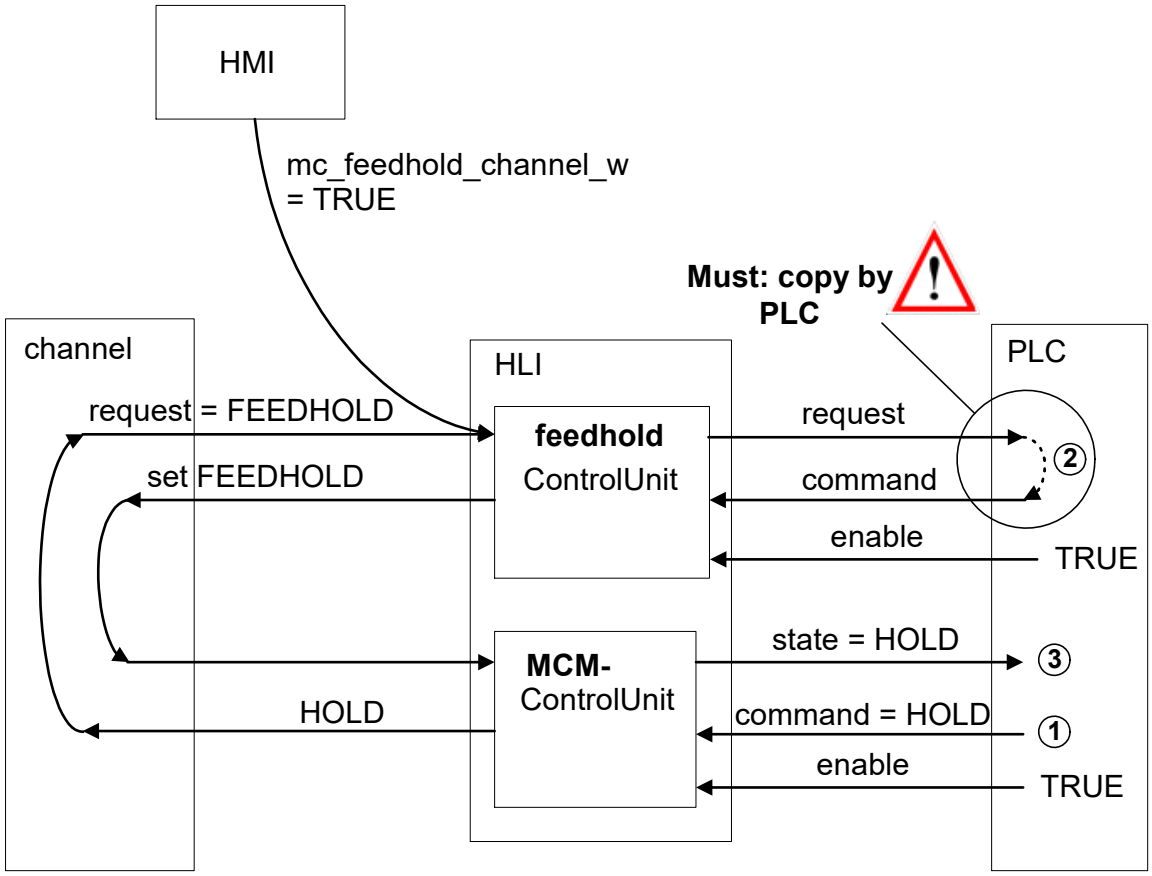


Fig. 22: Interaction between feedhold and NC channel stop

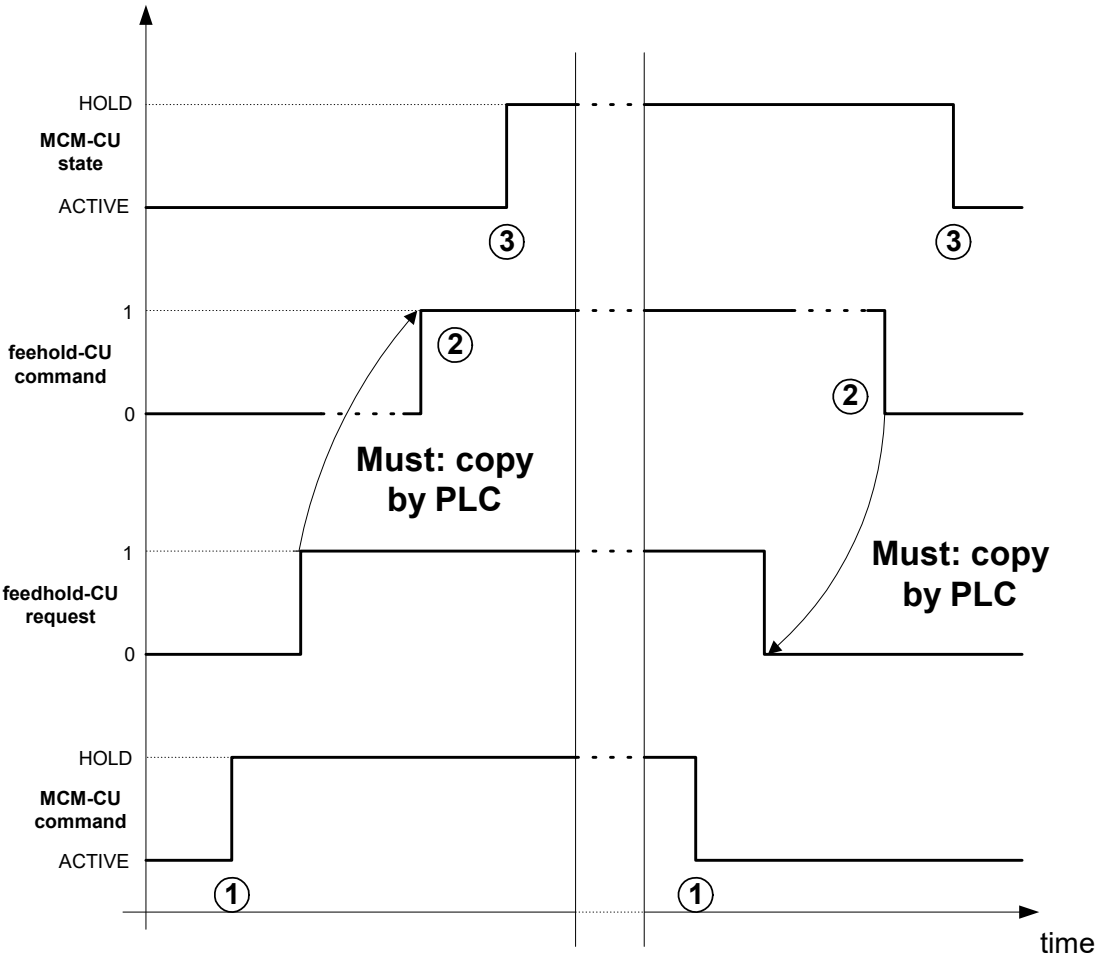


Fig. 23: Time sequence of feedhold and NC channel stop

Feedhold ON/OFF type 2	
Description	When this element is set to TRUE, the feed rate is immediately ramped down to feed rate = 0 during interpolation.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Special features	Irrespective of the channel-specific parameter P-CHAN-00097, the effective deceleration is defined by the values of the parameterised feedhold acceleration which is specified by the axis parameters P-AXIS-00024, P-AXIS-00053 (a_feedh) and P-AXIS-00081 (tr_feedh).
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**e_feedhold**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Emergency stop, feedhold ON, FALSE]
Redirection	
ST element	**.enable_w**

Emergency stop, channel	
Description	If this emergency stop element is set active (TRUE), interpolation is aborted immediately by output of the zero setpoint or by a deceleration as specified in the emergency deceleration. The NC kernel reverts to error state. This type of motion influence has maximum priority.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**emergency_stop**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Emergency stop active, FALSE]
Redirection	
ST element	**.enable_w**

Single-block mode	
Description	Activating/deactivating single-block mode. Single-block mode refers only to motion blocks. As long as single-block mode is activated, the system is decelerated to feed rate = 0 at the end of each motion block. The following blocks can then only be executed by setting the element “continue motion” if all axes are located in the control window.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**single_block**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Optional stop	
Description	Activating/deactivating optional stop. If the function **M01** (optional stop) is programmed in the current block of the NC program, set this element to the value TRUE to stop at block end (ramped-down deceleration complying with the permissible accelerations). The following block can be enabled by activating the element “continue machining” if the NC kernel indicates that all axes are located within the control window by resetting the status flag **wait_axes_in_position_r**.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**m01_stop_enable**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Optional stop active, FALSE]
Redirection	
ST element	**.enable_w**

Resumption of motion	
Description	If program execution is interrupted by selecting "Single block mode" [▶ 127] or"Optional stop" [▶ 127] or by M00, this control unit can resume NC program execution. A falling edge for the command value (command_w) of the control unit "Continue motion", i.e. a transition from TRUE to FALSE, leads to a resumption of NC program execution. The condition for this is that all axes are located in the control window.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Special features	**Falling** edge of the command resumes NC program execution.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**continue_motion**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Machining simulation ON/OFF	
Description	Activating/deactivating machining simulation. During machining simulation, all technology functions of the NC program are not output to the PLC but are acknowledged internally.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**machining_simulation**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Machining simulation active, FALSE = Machining simulation inactive]
Redirection	
ST element	**.enable_w**

Feed override	
Description	Feed override allows the programmed path velocity to be weighted with an additional factor
Data type	MC_CONTROL_UNS16_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**override_feedrate**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	UINT
Unit	0,1 %
Value range	[0, P-CHAN-00056] The parameter P-CHAN-00056 is a channel-specific parameter. The value is typically 1000.
Redirection	
ST element	**.enable_w**

Rapid traversing override	
Description	The rapid traverse override can weight G0 path motions by an additional factor. See also special features.
Data type	MC_CONTROL_UNS16_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**override_rapid_move**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	UINT
Unit	0,1 %
Value range	[0, 1000]
Redirection	
ST element	**.enable_w**
Special features	
Parameterisation/ mode of operation	Rapid traverse override is only active if this function is also activated in the channel parameter list. Otherwise, there is no distinction made between feed and rapid traverse blocks. Setting options in the channel parameter P-CHAN-00181: Rapid traversing override is inactive. Rapid traversing override acts on feed and rapid traversing blocks. Rapid traversing override is active. Feed override acts on feed blocks and the minimum of feed and rapid traversing override acts on rapid traversing blocks. Rapid traversing override is active. Feed override acts on feed blocks and only rapid traversing override acts on rapid traversing blocks.

Interrupt output of command values to real axes	
Description	This interrupts the output of NC channel command values to physical axes. The NC channel is stopped and assignment to the real axes is disabled. Physical axes can then be requested and moved by another channel. A different logical axis can be linked to a physical axis here. After clearing this interruption, the axes can be requested again and the original channel continues its motion.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh[channel_idx]^.bahn_mc_control.suspend_axis_output
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

External path velocity specified	
Description	External path velocity specified. The path velocity setting is activated by the control unit ext_command_speed_valid [▶ 127]. If the velocity specified in negative, the tool moves backwards along the path.(See FCT-C7// Forward/backward motion on the path)
Data type	MC_CONTROL_UNS32_UNIT, see description of Control Unit [▶ 13]
Special features	The path velocity transferred by this interface is automatically restricted to the limits defined in the axis parameters.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh[*channel_idx*]^.bahn_mc_control.**ext_command_speed**
Commanded, requested value	
ST Element	**.command_w** **.request_r**
Data type	UDINT
Unit	1 µm/s
Return value	
ST Element	**.state_r**
Data type	UDINT
Unit	1 µm/s
Special features	The state_r element indicates the path velocity actually used in the interpolator, including any influence by override. By default the externally specified velocity only affects machin8ing motions (G01, G02, G03). The channel parameter P-CHAN-00102 (plc_command_rapid_feed) can set whether the externally specified velocity also acts on rapid traverse motions (G00).
Redirection	
ST Element	**.enable_w**

Activation of external path velocity	
Description	Activate the velocity commanded in the ext_command_speed [▶ 127] control unit. To reach the commanded velocity, all axes involved in the motion are accelerated or decelerated. If this value is TRUE, the sign is considered in the current path feed (active_feed_r [▶ 101] control unit).
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh[*channel_idx*]^.bahn_mc_control.**ext_command_speed_valid**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST Element	**.enable_w**

Input disable, interpolator	
Description	When the control unit is activated, the interpolator stops after it has processed the input command blocks since this control unit can disable the input of further command blocks. The channel parameter P-CHAN-00267 can define the event at which an activated input disable becomes effective, e.g. effective as of the next rapid traverse block.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Special features	See safety not below.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**input_disable**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

⚠ CAUTION

Stop command to CNC is not executed.

Possible damage to machine.

Similar to other control units, a command from an external application, e.g. an HMI, is indicated in **request_r**. If the PLC operates this control unit and therefore set **enable_w** to TRUE, the command from the external application only becomes effective when **request_r** is written to **command_w**. Here the semaphores must be operated as usual.

This must also be considered if the PLC operates the control unit in order to switch operation modes (see section 8.2 [▶ 190]) and therefore sets the **enable_w** to TRUE A HOLD command given by this control unit triggers the NC kernel to request input disable. This is again indicated in request_r of the control unit described here and only becomes effective when the PLC copies PLC **request_r** to **command_w**. The same also applies to cancelling input disable.

The figures below depict these functions.

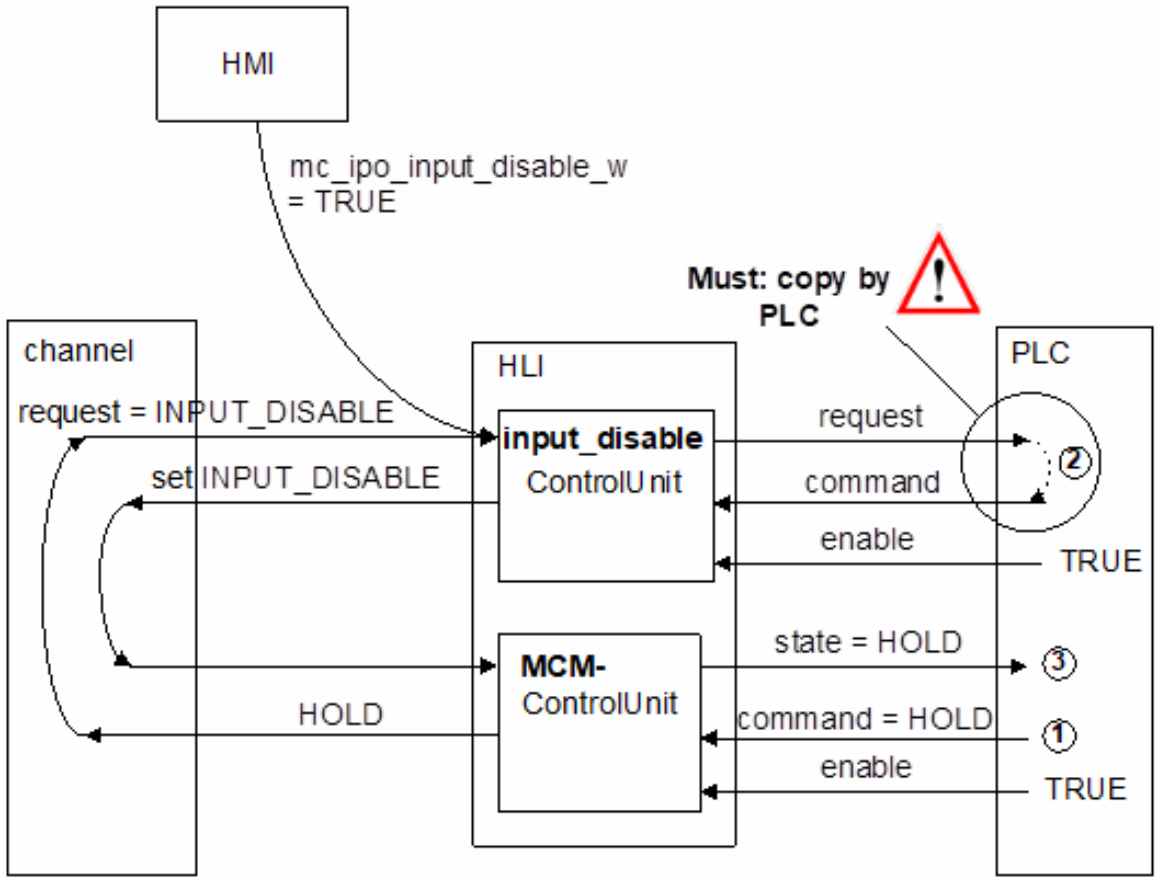


Fig. 24: Interaction of input disable and NC channel stop

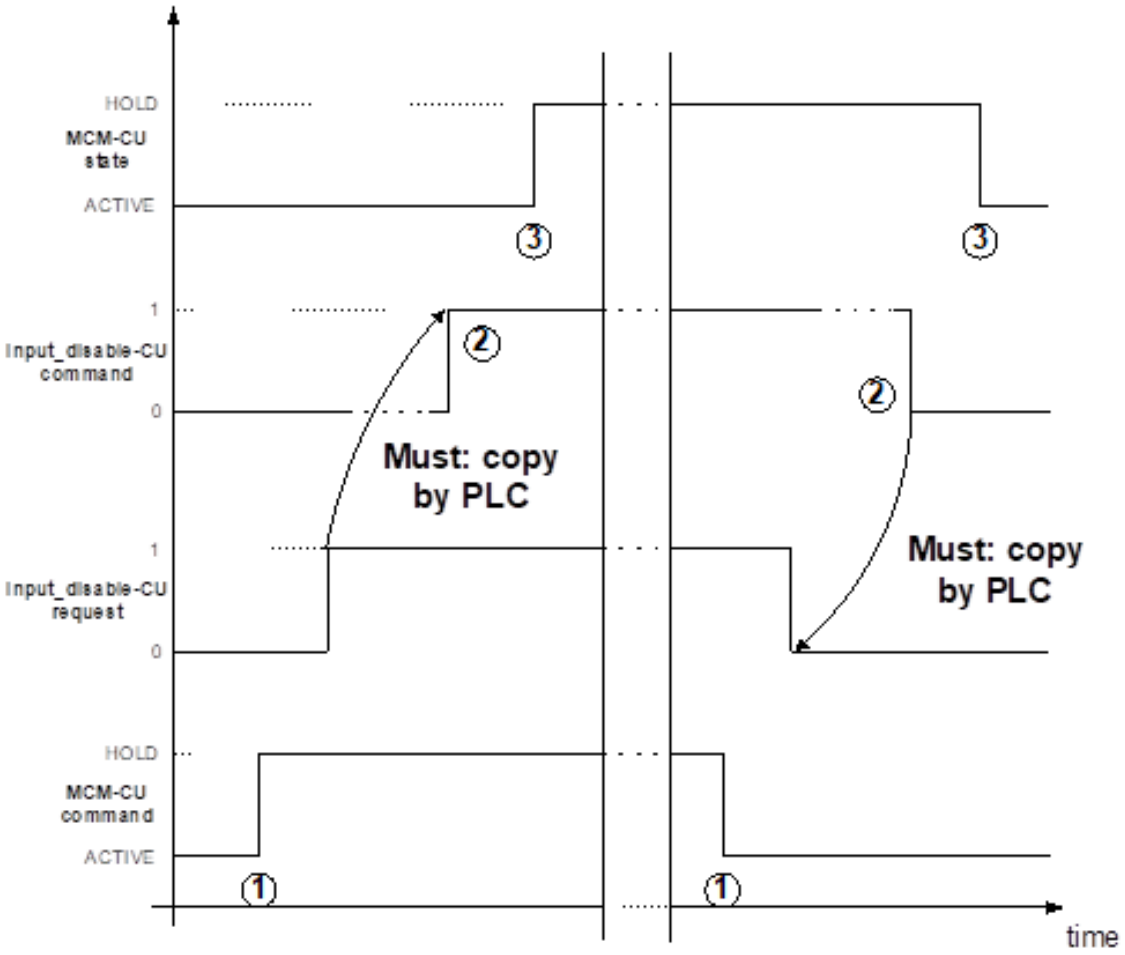


Fig. 25: Time sequence of input disable and NC channel stop

Reduced velocity, channel	
Description	When this signal is set, the path velocity is reduced to the values defined in the axis parameters P-AXIS-00214 and P-AXIS-00155.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Characteristics	The limits of the axes participating in the motion are taken into consideration. The effective value for reduced velocity is determined so that none of the axes participating in the motion overshoots its configured limit.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh[*channel_idx*]^.bahn_mc_control.**reduced_speed**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Reduced velocity active, FALSE = Reduced velocity not active]
Redirection	
ST Element	**.enable_w**

Reduced velocity in zone 1, channel	
Description	When this signal is set, the path velocity is limited to the velocity defined in the axis parameterP-AXIS-00030 if the axis is located within the area defined by the parameters P-AXIS-00085 and P-AXIS-00093. If necessary the axis is decelerated after entering the zone.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**reduced_speed_zone**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Reduced velocity in zone 1 active, FALSE = Reduced velocity in zone 1 not active]
Redirection	
ST element	**.enable_w**

Reduced velocity in zone 2, channel	
Description	When this signal is set, the path velocity is limited to the velocity defined in the axis parameterP-AXIS-00030 if the axis is located within the area defined by the parameters P-AXIS-00097 and P-AXIS-00105. If necessary the axis is decelerated after entering the zone.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**reduced_speed_2_zone**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Reduced velocity in zone 2 active, FALSE = Reduced velocity in zone 2 not active]
Redirection	
ST element	**.enable_w**

Path acceleration reduction	
Description	This control unit is used to reduce path acceleration while the program is running. The change in acceleration is time-delayed and is therefore only suitable for slow changes. The state of the control unit returns the commanded value as soon as the value is accepted the first time. If a value outside the value range is commanded, the value is not accepted and the state 0 is displayed.
Data type	MC_CONTROL_UNS16_UNIT, see description of Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[channel_idx]^.bahn_mc_control.**reduction_acceleration**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	UDINT
Unit	[0.1%]
Value range	[0, 999]
Redirection	
ST element	**.enable_w**

Activation of reduced manual mode acceleration	
Description	This control unit activates the parameterised reduced maximum axis acceleration (P-AXIS-00545) in manual mode without any parallel interpolation.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Access	PLC reads requested + feedback values and writes commanded value + redirection.
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**reduced_acceleration**
Commanded, requested and return values	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Time override valid	
Description	Time override is activated
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**command_t_ovrd_valid**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Time override is activated, time override is not activated]
Redirection	
ST element	**.enable_w**

Time override	
Description	Time override can influence the internal CNC time base for motions. Its effect is similar to slow motion. Time override affects path velocity and acceleration differently. Example: If time override is 50% (command_w = 500) the velocity is reduced by a factor of 2 and acceleration by a factor of 4.
Data type	MC_CONTROL_UNS16_UNIT, see description Control Unit [▶ 13]
Special features	The channel parameter P-CHAN-00111 can be used to affect the effect of time override on the dwell time function. See safety not below.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**command_t_ovrd**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	UINT
Unit	0,1 %
Value range	[100, 1000]
Redirection	
ST element	**.enable_w**

Attention

Time override affects real-time and safety functions.

Delays in the use of these functions may lead to longer reaction times for safety functions under certain circumstances.

Time override affects path feed and acceleration independent of real-time functions such as feedhold or safety functions such as reduced speed. The user must consider this when using this function.

Disabling the detection of tool life data	
Description	Detecting tool life data is suppressed when this control unit is activated.
Data type	MC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**tool_life_suppress_capture**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = Tool life data is not detected, FALSE = Tool life data is detected]
Redirection	
ST element	**.enable_w**

Delete distance to go, command	
Description	The rising edge of the commanded value has the effect that the CNC channel is decelerated to feed velocity 0. Then a linear motion is executed to the target position of the next motion block (short cut). The command only affects motion blocks. The functional description [FCT-C28] deals with the topic of “Delete distance to go” in detail.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Characteristics	See state data delete_distance_to_go_active_r [▶ 99]. P-STUP-00033 must be configured for the channel.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh[*channel_idx*]^.bahn_mc_control.**delete_distance_to_go**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r** (TRUE indicates that the command was detected by the CNC)
Data type	BOOL
Value range	[TRUE = distance to go to be deleted, FALSE = no impacts on motion blocks]
Redirection	
ST Element	**.enable_w**

Delete distance to go, end marker	
Description	This control unit sets the end marker defined in the NC program online to valid. A bit mask is defined for this in the NC program. If at least one single bit in the end marker is set on the NC interface, this marker is valid as a jump target. Example: `N10 #DEL DIST2GO [END = '16#0014']` The end marker can be set to valid by bit3 (hexadecimal 4) or by bit5 (hexadecimal 0x10).
Data type	MC_CONTROL_UNS32_UNIT, see description of Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh[*channel_idx*]^.bahn_mc_control.**delete_distance_to_go_activation**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	UDINT
Value range	32-bit
Redirection	
ST Element	**.enable_w**

Online tool compensation (OTC)	
Description	The tool radius is compensated by specifying a wear value. The wear value is specified as an offset from the planned cutting edge. It is only possible to used the control unit in the RADIUS or TOOL_DIR modes and in the DISC or AUTO mode. See NC command #OTC. See also functional description [FCT-C20].
Data type	MC_CONTROL_SGN32_UNIT, see description of Control Unit [▶ 13]
Special features	If the wear value changes, this change is distributed over several interpolation cycles to avoid sudden changes.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh[*channel_idx*]^.bahn_mc_control.**otc_radius_offset**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	DINT
Unit	0.1 µm
Value range	[-P-TOOL-00031, P-TOOL-00031]
Redirection	
ST Element	**.enable_w**

Reversible stop	
Description	Select/deselect reversible stop. If the #STOP REVERSIBLE [LEVEL=<bitmask>] is programmed in the current block in the NC program,<*bitmask*>the program stops at block end (ramped deceleration based on the permitted deceleration values) if the same value assigned to the LEVEL option in the NC program is already sent by this control unit to the MOTION Controller thus activating the STOP instruction. The next block is enabled by a control unit command "Continue motion" [▶ 127] (continue_motion) if the NC kernel displays that all axes are located in the control window by resetting the status display "Wait for axis group in position" [▶ 99] (wait_axes_in_position_r).
Data type	MC_CONTROL_UNS32_UNIT, see description Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**stop_reversible_level**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	UDINT
Value range	[0, MAX_UNS32]
Redirection	
ST element	**.enable_w**

Backward motion	
Description	Select/deselect backward motion on the path In basic setting, M/H functions are executed without synchronisation (MOS) in this mode.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Access	PLC reads requested + feedback values and writes commanded value + redirection.
ST path	gpCh[*channel_idx*]^.bahn_mc_control.**backward_motion**
Commanded, requested and return values	
ST element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Simulated motion	
Description	Select/deselect simulated forward motion on the path In basic setting, M/H functions are executed without synchronisation (MOS) in this mode. Sections in the NC program can be skipped during program runtime in combination with the NC command #OPTIONAL EXECUTION.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Access	PLC reads requested + feedback values and writes commanded value + redirection.
ST Path	gpCh[*channel_idx*]^.bahn_mc_control.**simulate_motion**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Reset backward motion memory	
Description	Deselects backward motion memory No further NC block is saved in the memory. The memory is deleted. The backward motion memory can only be cleared if no NC program is active.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Access	PLC reads requested + feedback values and writes commanded value + redirection.
ST Path	gpCh[channel_idx]^.bahn_mc_control.backward_storage_off
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST element	**.enable_w**

Suppress error output from working space monitoring in manual mode	
Description	This control unit suppresses the output of error messages from the working space monitoring system [FCT-C14] in manual mode.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Access	PLC reads requested + feedback values and writes commanded value + redirection.
ST Path	gpCh[*channel_idx*]^.bahn_mc_control.**suppress_area_mon_manual_mode**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = suppress error output; FALSE]
Redirection	
ST Element	**.enable_w**

Simulate motion mask	
Description	This control unit specifies a mask. Sections in the NC program can be skipped during program runtime in combination with the “simulate motion” control unit [▶ 153] and the NC command #OPTIONAL EXECUTION. When the “simulate motion” control unit [▶ 153] is on a positive edge, all sections flagged by #OPTIONAL EXECUTION [SIMULATE MASK=<mask>] containing a bit of the mask are skipped.
Data type	MC_CONTROL_UNS64_UNIT, see description of Control Unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w.
ST Path	gpCh[*channel_idx*]^.bahn_mc_control.**simulate_motion_mask**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	ULINT
Value range	0 – MAX(UNS64)
Redirection	
ST element	**.enable_w**
Special feature	**Available as of CNC Build V3.1.3107.12**

Switch on velocity limiting	
Description	By setting this command, the maximum path velocity is limited according to the kinematic and velocity settings.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh[channel_idx]^.bahn_mc_control.**tcp_velocity_limit**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Unit	
Value range	[TRUE = limits active, FALSE = limits not active]
Redirection	
ST Element	**.enable_w**
Special feature	**Available as of CNC Build V3.1.3079.26**

4.2.2.1 Dynamic coordinate system

Release Note

This function is available as of CNC Build V3.1.3054

4.2.2.1.1 Control unit

Dynamic CS		
Description	Control unit to switch over dynamic CS tracking.	
Data type	MC_CONTROL_DYN_CS_UNIT [▶ 156]	
ST path	gpCh[*channel_idx*]^.channel_mc_control.**dyn_cs**	
Commanded, requested data		
ST element	**.command_w** **.request_r**	
Data type	HLI_COORDINATE_SYSTEM_INT translation: ARRAY [0..HLI_CS_AXES_MAXIDX] OF DINT;	
		X/Y/Z translation in [0.1 µm]
	rotation : ARRAY [0..HLI_CS_AXES_MAXIDX] OF DINT;	
		A/B/C rotation in [0.0001 degree]
Access	PLC writes command and reads request	
Return data		
ST element	**.state_r**	
Data type	HLI_DYN_CS_STATE actual_state : DINT;	HLI_DYN_CS_INACTIVE = 0
		HLI_DYN_CS_ACTIVATING = 1,
		HLI_DYN_CS_ACTIVE = 2,
		HLI_DYN_CS_DEACTIVATING = 3,
		HLI_DYN_CS_ERROR = -1
Access	PLC is reading	
Flow control of commanded value		
ST element	**.command_semaphor_rw**	
Data type	BOOL	
Value range	[TRUE, FALSE]	
Special features	**Consumption data item**	
Access	CNC accepts the commanded data if this element has the value TRUE and sets this element to the value FALSE after complete acceptance of the data. PLC can write data for commanding if this element has the value FALSE. The PLC sets this element to the value TRUE if all data to be commanded is written.	
Flow control of requested data		
ST element	**.request_semaphor_rw**	
Data type	BOOL	
Value range	[TRUE, FALSE]	
Special features	**Consumption data item**	

Access	CNC writes the data requested by the GUI if this element is FALSE and then sets this element to TRUE. PLC reads the data requested by the GUI if this value is TRUE. After the PLC fully accepts the data, the PLC sets this element to FALSE.
Redirection	
ST path	gpCh[*channel_idx*]^.channel_mc_control.dyn_cs.**enable_w**

Transition			
ST path	gpCh[*channel_idx*]^.channel_mc_control.dyn_cs.**transition_w**		
Data type	HLI_DYN_CS_TRANSITION		
	command	: DINT;	(* -1:DEACTIVATE, 1:ACTIVATE *)
	filter_max_ticks	: UDINT;	(* filter for turning ON/OFF, compare #TRACK CS ON [ID=<i> ...FILTER... *)
	option	: UDINT;	(* additional option, compare #TRACK CS ON [ID=<i> ...OPTION... *)
	f_wait	: BOOL;	(* #TRACK CS ON [...WAIT... *)
	f_set_zero	: BOOL;	(*#TRACK CS ON [...SET_ZERO... *)
	f_kin_base	: BOOL;	(* #TRACK CS ON [...KIN_BASE... *)
	f_rot_trans	: BOOL;	(* #TRACK CS ON [...ROT_TRANS... *)
	kinematic_base_cs	: HLI_COORDINATE_SYSTEM_INT;	
		(* add. shift between error and kinematic base, #TRACK CS ON [ID=<i> X=. Y=. *)	
Access	PLC writes the transition in analogy to the NC command #TRACK CS [ID=0 …] and CNC reads the transition. Correct NC/PLC handshake: First assign all parameters and then set command to +/-1.		

Programing Example

Control unit

```
TYPE HLI_COORDINATE_SYSTEM_INT :
STRUCT
  translation : ARRAY [0..HLI_CS_AXES_MAXIDX] OF DINT;
  fill_up_2 : DINT;
  rotation       : ARRAY [0..HLI_CS_AXES_MAXIDX] OF DINT;
  fill_up_1 : DINT;
END_STRUCT
END_TYPE

TYPE HLI_DYN_CS_STATE :
STRUCT
  actual_state : UDINT;
  fill_up_1    : DINT;
END_STRUCT
END_TYPE

TYPE HLI_DYN_CS_TRANSITION :
STRUCT
  command           : DINT;
  filter_max_ticks  : UDINT;
  option            : UDINT;
  f_wait            : BOOL;
  f_set_zero        : BOOL;
  f_kin_base        : BOOL;
  f_rot_trans       : BOOL;
  kinematic_base_cs : HLI_COORDINATE_SYSTEM_INT;
END_STRUCT
END_TYPE

TYPE MC_CONTROL_DYN_CS_UNIT :
STRUCT
  enable_w            : BOOL;  (* MC <-- PLC takes care *)
  request_semaphor_rw : BOOL;  (* Valid semaphore *)
  command_semaphor_rw : BOOL;  (* Valid semaphore *)
  fill_up_1           : BOOL;
  fill_up_2           : DINT;
  request_r           : HLI_COORDINATE_SYSTEM_INT;
  command_w           : HLI_COORDINATE_SYSTEM_INT;
  transition_w        : HLI_DYN_CS_TRANSITION;
  state_r             : HLI_DYN_CS_STATE;
END_STRUCT
END_TYPE
```

4.2.2.2 Contour look-ahead

Release Note

This function is available as of CNC Build V3.1.3070.

4.2.2.2.1 Control unit – Contour look-ahead

Contour look-ahead	
Description	The PLC can use this control unit to request and read out information on contour elements from the CNC. The PLC can use this information to plan the look-ahead process.
Data type	HLI_CONTOUR_LOOKAHEAD
ST path	gpCh[*channel_idx*]^.**contour_lookahead**
Command data	
ST Element	**.request**
Data type	HLI_CONTOUR_LOOKAHEAD_REQUEST [▶ 161]
Access	PLC writes request
Return data	
ST element	**.response**
Data type	HLI_CONTOUR_LOOKAHEAD_RESPONSE [▶ 162]
Access	PLC reads
Flow control of commanded value	
ST element	**.semaphor_rw**
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	TRUE : PLC triggers on new request FALSE : CNC has read new request
Status value	
ST Element	**.state**
Data type	HLI_CONTOUR_LOOKAHEAD_STATE [▶ 161]

4.2.2.2.2 User data

Contour look-ahead request data	
Description	Data to request the contour elements from the CNC
ST path	gpCh[*channel_idx*]^.*contour_lookahead.request*
ST name	HLI_CONTOUR_LOOKAHEAD_REQUEST
ST element	**.start_position**
Data type	LREAL
Description/special features	The meaning of the element is dependent on the mode of the contour look-ahead function used. Mode 1: Index of the contour element in the marked area Mode 2: Start distance from which the distance of the “length” parameter applies
ST element	**.length**
Data type	LREAL
Description/special features	A distinction is made whether mode 1 or 2 should be used depending on the entry of this element. Value = 0 : Request contour element by index, mode 1 Value != 0: Request contour element by distance, mode 2 This mode indicates the distance up to which contour element is to be requested

Contour look-ahead status data	
Description	Status data of the contour look-ahead control unit
ST Path	gpCh[*channel_idx*]^.contour_lookahead**.state**
ST name	HLI_INSERT_CMD_STATE
ST element	**.max_dist_prog_start**
Data type	LREAL
Description	Maximum look-ahead distance from program start in [0.1 µm].
ST element	**.count_logged_blocks**
Data type	UDINT
Description	Number of available contour elements that can be read out.
ST element	**.f_log_is_active**
Data type	BOOL
Description	This flag signals whether logging is still active for the marked area and all contour elements were read in. TRUE : not all contour elements were read in

Contour look-ahead return data	
Description	Contour information supplied by the CNC
ST Path	gpCh[*channel_idx*]^.contour_lookahead**. response**
ST name	HLI_CONTOUR_LOOKAHEAD_RESPONSE
ST Element	**.block[i]**
Data type	HLI_CONTOUR_LOOKAHEAD_BLOCK [▶ 163]
Description	

Structure HLI_CONTOUR_LOOKAHEAD_BLOCK	
Description	Information on a contour element that was supplied to the PLC by the CNC
ST Element	**.block_type**
Description	Block type 0 – no element exists 1 - linear contour element 2 - circular contour element 3 - #CONTOUR LOOKAHEAD LOG ON 4 - #CONTOUR LOOKAHEAD LOG [PARAM] 5 - #CONTOUR LOOKAHEAD LOG OFF 6 – Polynomial contour element
ST Element	**.block_number**
Description	Programmed NC block number
ST Element	**.block_count_r**
Description	Unique NC block ID number Corresponds to the displayed number of block_count_r [▶ 100] in the Status information of a channel [▶ 99].
ST Element	**.start_position**
Description	Start distance of the contour element from program start
ST Element	**.length**
Description	Length of the contour element of the programmed value of the PARAM.
ST Element	**.programmed_radius**
Description	Programmed radius of the circular element
ST Element	**.compensated_radius**
Description	Compensated radius of the circular element
ST Element	**.circle_angle**
Description	Swept angle of the circular element. • Clockwise [-2*pi,0) • Counter-clockwise (0,2*pi].
ST Element	**.length_min**
Description	With die sinking, the length of the contour element is displayed on the scaled contour (radius = R_MAX_SCALE).
ST Element	**.tangent_variation**
Description	End angle to the previous motion block. (0,pi)

4.2.2.3 Insert stop marks

Release Note

This function is available as of CNC Build V3.1.3105.01.

4.2.2.3.1 Control unit – Insert stop marks (insert command)

Insert stop marks	
Description	During runtime, the PLC can use this control unit to set stop points in the future NC program. The control unit must then be enabled by enable_w = TRUE. The PLC signals a new assignment of command_2 by setting **command_semaphor_rw** to TRUE. The CNC sets the command_semaphor_rw to FALSE after the command_w data is read.
Data type	MC_CONTROL_INSERT_CMD_UNIT
ST path	gpCh[*channel_idx*]^. bahn_mc_control.**insert_cmd**
Command data	
ST element	**.command_w**
Data type	HLI_INSERT_CMD_COMMAND [▶ 165]
Access	PLC writes command_w
Return values	
ST element	**.state_r**
Data type	HLI_INSERT_CMD_STATE [▶ 166]
Access	PLC reads
Activation	
ST element	**.enable_w**
Data type	BOOL
Access	PLC writes
Value range	[TRUE/FALSE] ; TRUE: CU enabled in PLC
Flow control of commanded value	
ST element	**.semaphor_rw**
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	TRUE : PLC triggers on new request FALSE : CNC has read new request

4.2.2.3.2 User data

Command data – Insert command	
Description	Control data for inserted STOP command
ST path	gpCh[*channel_idx*]^.bahn_mc_control.insert_cmd.**command_w**
ST name	HLI_INSERT_CMD_COMMAND
ST element	**.dist_or_pos**
Data type	LREAL
Description/special features	Relative/absolute distance or axis position at which the stop is to be inserted. [0.1 µm] The default setting is DISTANCE mode, the POSITION mode is active at a value of ".axis_nr" > 0
ST element	**.rel_abs_mode**
Data type	UINT
Description/special features	Value = 0 for relative distance with automatic insertion of a new stop when the current stop is reached Value = 1 for one-off insertion at the absolute distance Value = 2 for one-off insertion at the relative distance
ST element	**.axis_nr**
Data type	UINT
Description/special features	Value = 0 for DISTANCE mode Value > 0 for POSITION mode: Stop is inserted at the commanded axis position of the axis with the logical axis number = ".axis_nr".
ST element	**.m_function_nr**
Data type	UINT
Description/special features	Number of the inserted M function • Value = 0 for M00 programmed stop • Value = 1 for M01 optional stop (enabled/disabled with m01_stop_enable CU) • Value > 1 for M<m_function_nr> with MVS_SVS synchronisation
ST element	**.add_nr**
Data type	DINT
Description/special features	Optional additional value of the M function; it is specified as a negative or positive integer.

Status data – Insert command	
Description	Status data of the insert command control unit
ST path	gpCh[*channel_idx*]^.bahn_mc_control.insert_cmd.**state_r**
ST name	HLI_INSERT_CMD_STATE
ST element	**.distance_of_next_stop**
Data type	LREAL
Description	Absolute distance (dist_prog_start) of the feed axes (#FGROUP) at the next stop [0.1 µm] Value >= 0 : Stop found, axis positions correct Value = -1 : Stop still not found in the NC program
ST element	**.position_at_next_stop[*idx*]**
Data type	ARRAY[0..HLI_CS_AXIS_MAXIDX] OF DINT
Description	When • command_w.axis_nr = 0, the axis positions X, Y, Z are assigned accordingly at the next stop in the PCS [0.1 µm] • command_w.axis_nr > 0; .position_at_next_stop[0] contains the axis position of the commanded axis at the next stop position_at_next_stop[1]=0 and position_at_next_stop[2]=0
ST element	**.state**
Data type	DINT
Description	Status if the current command_w. • Value =0: no stop commanded • Value=1: Stop commanded but not yet reached

5 PLC

5.1 Control commands to PLC

5.1.1 Reset

PLC reset, axis	
Description	The PLC can be requested to perform a reset via this axis-specific interface. Here the PLC must indicated that it wants to be notified about reset requests by the NC kernel by setting the element enable_w. See also State graph of operation modes [▶ 187]
Data type	LC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Peculiarities	**Consumption data item**
Access	CNC sets **command_r** to TRUE to command a reset for the PLC. The CNC sets **command_r** to FALSE after the PLC acknowledges the execution of the reset via the **state_w** element.
ST Path	gpAx[*axis_idx*]^.ipo_lc_control.**plc_reset**
Commanded value	
ST Element	**.command_r**
Signal flow	CNC → PLC
Data type	BOOL
Value range	[TRUE = Reset request from CNC to PLC, FALSE]
Return value	
ST Element	**.state_w**
Signal flow	PLC → CNC
Data type	BOOL
Value range	[TRUE = PLC executed reset, FALSE]
Request	
ST element	**.enable_w**
Signal flow	PLC → CNC
Data type	BOOL
Value range	[TRUE = PLC wants to be notified about requests by CNC, FALSE]

PLC reset, channel	
Description	The PLC can be requested to execute a reset via this channel-specific interface. Here the PLC must indicated that it wants to be notified about reset requests by the NC kernel by setting the element enable_w. See also State graph of operation modes [▶ 187]
Data type	LC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Peculiarities	**Consumption data item**
Access	CNC sets **command_r** to TRUE to command a reset for the PLC. The CNC sets **command_r** to FALSE after the PLC acknowledges the execution of the reset via the **state_w** element.
ST Path	gpCh[*channel_idx*]^.bahn_lc_control.**plc_reset**
Commanded value	
ST Element	**.command_r**
Signal flow	CNC → PLC
Data type	BOOL
Value range	[TRUE = Reset request from CNC to PLC, FALSE]
Return value	
ST Element	**.state_w**
Signal flow	PLC → CNC
Data type	BOOL
Value range	[TRUE = PLC executed reset, FALSE]
Request	
ST element	**.enable_w**
Signal flow	PLC → CNC
Data type	BOOL
Value range	[TRUE = PLC wants to be notified about requests by CNC, FALSE]

Notice

Before a new command and after detecting reset acknowledgement, the state_w state is cleared by the CNC.

5.1.2 Block search

Block search on/off, handshake with PLC	
Description	At every block search on/off request, the CNC initiates a handshake with the PLC; As long as the request element has the value TRUE, the PLC is notified of every change in block search mode.
Data type	LC_CONTROL_BOOL_UNIT, see description Control Unit [▶ 13]
Special features	**Consumption data item**
Access	If an NC program starts with the block search mode activated, the CNC set the CNC signal to TRUE and waits for the PLC signal indicating that it is ready for block search. When the PLC has executed the necessary actions to prepare for block search, it notifies this to the CNC by setting the PLC signal to TRUE. After this signal, the NC program can be processed in block search mode. This takes place by the “Continue motion” or “Program start” commands. If the re-engagement position is reached while the NC program is processed, the CNC signals this by setting the CNC signal to FALSE. The PLC detects this, completes its preparations for operation with real axis motions and then sets the PLC signal to FALSE.
ST path	gpCh[*channel_idx*]^.bahn_lc_control.**block_search**
CNC signal	
ST element	**.command_r**
Signal flow	CNC → PLC
Data type	BOOL
Value range	[TRUE = NC program was started in block search mode, FLASE = Block search is disabled]
PLC signal	
ST element	**.state_w**
Signal flow	PLC → CNC
Data type	BOOL
Value range	[TRUE = PLC acknowledged the notification about block search activation, FALSE = PLC acknowledged the notification of block search deactivation]
Requirement	
ST element	**.enable_w**
Data type	BOOL
Value range	[TRUE = PLC wants to be notified about activation of block search, FALSE]

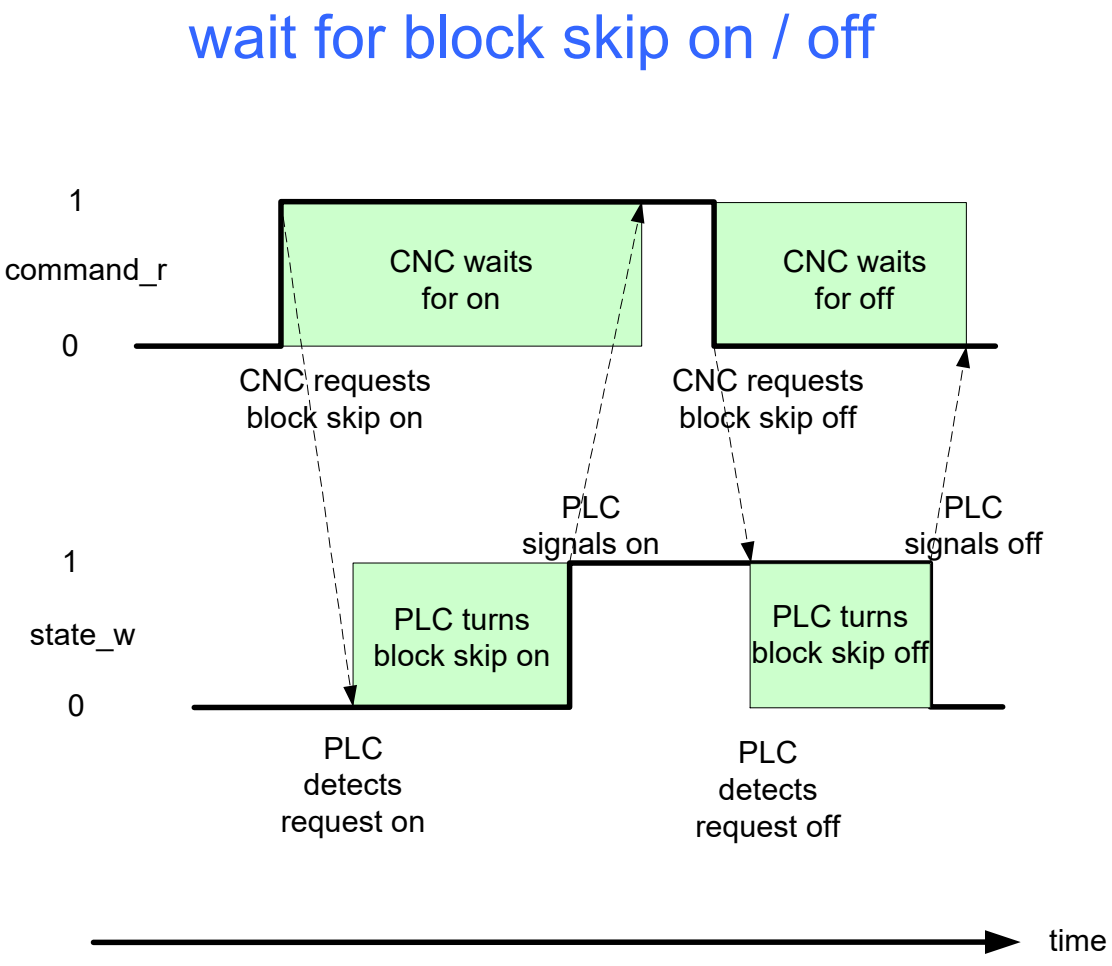


Fig. 26: Interaction between BOOLEAN-LMC control unit and PLC

Notice

When the CNC is reset, the signals command_r and state_w are reset by the CNC.

6 Technology processes

6.1 Introduction

Technology functions can be defined for each channel and each axis.

Channel-specific technology functions are defined in the channel parameters; axis-specific technology functions are defined in the axis parameters of each axis.

This definition also includes the synchronisation mechanism of the technology function. A distinction is made between two basic types of synchronisation:

- Block-by-block synchronisation (default synchronisation),
- Cross-block synchronisation.

M functions not defined are indicated as unknown M functions by an error message after NC program start and decoding is aborted.

6.2 Management of technology functions

A basic distinction is made between two types of technology functions:

technology functions synchronised block-by-block and cross-block synchronised technology functions. This subdivision is also reflected on the High-Level Interface.

All technology functions to be synchronised block-by-block must be acknowledged at the latest at the end of the NC block in which they were programmed. They are therefore saved in consecutive sequence in the corresponding management field.

This is not the case with technology functions with cross-block synchronisation. In this case, the non-acknowledged technology functions yet to be executed may be distributed over the entire field (with gaps). In addition, it may occur that several identical technology functions are present in this field since, owing to cross-block synchronisation, the individual technology commands do not need to be acknowledged at the end of the NC block in which they were programmed. This must be allowed for at the PLC end to generate the acknowledgement of technology functions.

The number of technology functions programmed in an NC block is made available on the HLI for the array of the technology functions synchronised block-by-block. The number of non-acknowledged technology functions is specified for the array of cross-block synchronised technology functions.

6.3 Elements to manage axis-specific technology control units

6.3.1 Block-by-block synchronisation (default synchronisation)

Array of technology functions with block-by-block synchronisation	
Description	Array of M/H/S/T technology functions with block-by-block synchronisation. Technology functions are saved contiguously in this array.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.**techno_unit_std_sync[***tech_unit_idx***]**
Data type	ARRAY [0..HLI_TU_AX_STD_SYNC_MAXIDX] OF TECHNO_UNIT_AX

Number of technology functions with block-by-block synchronisation	
Description	Number of valid entries in the array techno_unit_std_sync (= number of technology functions to be acknowledged in this block)
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.**used_units_std_sync_r**
Data type	UINT
Value range	[0, HLI_TU_AX_STD_SYNC_MAXIDX + 1]

6.3.2 Cross-block synchronisation

Technology functions with cross-block synchronisation	
Description	Array of M/H/S/T technology functions with cross-block synchronisation. There may be entries of already acknowledged technology functions among the entries for non-acknowledged M functions.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.**techno_unit_late_sync[***tech_unit_idx***]**
Data type	ARRAY [0..HLI_TU_AX_LATE_SYNC_MAXIDX] OF TECHNO_UNIT_AX

Number of technology functions with cross-block synchronisation	
Description	Number of non-acknowledged technology functions in the array techno_unit_late_sync
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.**used_units_late_sync_r**
Data type	UINT
Value range	[0, HLI_TU_AX_LATE_SYNC_MAXIDX + 1]

6.4 Elements for managing channel-specific technology control units

6.4.1 Block-by-block synchronisation (default synchronisation)

Array of technology functions with block-by-block synchronisation	
Description	Array of M/H/S/T technology functions with block-by-block synchronisation. Technology functions are saved contiguously in this array.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.**techno_unit_std_sync[***tech_unit_idx***]**
Data type	ARRAY [0..HLI_TU_CH_STD_SYNC_MAXIDX] OF TECHNO_UNIT_CH
Access	PLC reads

Number of technology functions with block-by-block synchronisation	
Description	Number of entries in the array ATechnoUnitChannel_Std (= number of technology functions to be acknowledged in this block)
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.**used_units_std_sync_r**
Data type	UINT
Value range	[0, HLI_TU_CH_STD_SYNC_MAXIDX + 1]

6.4.2 Cross-block synchronisation

Technology functions with cross-block synchronisation	
Description	Array of M/H/S/T technology functions with cross-block synchronisation. There may be entries of already acknowledged technology functions among the entries for non-acknowledged M functions.
Signal flow	CNC → PLC
Unit	0,1 µm
ST path	gpCh[*channel_idx*]^.**techno_unit_late_sync[***tech_unit_idx***]**
Data type	ARRAY [0..HLI_TU_CH_LATE_SYNC_MAXIDX] OF TECHNO_UNIT_CH

Number of technology functions with cross-block synchronisation	
Description	Number of entries in the array ATechnoUnitChannel_Late.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.**used_units_late_sync_r**
Data type	UINT
Value range	[0, HLI_TU_CH_LATE_SYNC_MAXIDX+ 1]

6.5 Data of a technology control unit

6.5.1 Data of an axis-specific technology control unit

Data of a technology function, axis			
Description	A technology control unit contains elements for commanding, acknowledging and transferring any required parameters.		
Data type	TECHNO_UNIT_AX		
ST path	**Default synchronisation:** gpAx[*axis_idx*]^.techno_unit_std_sync[*tech_unit_idx*] **Cross-block synchronisation:** gpAx[*axis_idx*]^.techno_unit_late_sync[*tech_unit_idx*]		
Job			
ST element	**.please_rw**		
Description	By setting please_rw, the CNC signals to the PLC that the technology control unit is to be executed.		
Data type	BOOL		
Value range	[TRUE, FALSE]		
Special features	**Consumption data item**		
Access	The CNC refreshes the data of the technology function only if this element is FALSE. After updating, the CNC sets this element to TRUE and so element **done_w** is set to FALSE. The PLC reads the data of the technology function if this element has the value TRUE. After the data is transferred, the PLC sets the value to FALSE.		
Function type			
ST element	**.fkt_ctrl_r**		
Description	The type of technology function is transferred in fkt_ctrl_r.		
Data type	UINT		
Wertebereich2 (value range 2)	Value	Constant	Meaning
	1	HLI_INTF_M_FKT	M function
	2	HLI_INTF_H_FKT	H function
	3	HLI_INTF_SPINDEL	S function
Access	PLC reads		
Parameter			
ST element	**.attribut_r.data**		
Description	Depending on the content of the element **fkt_ctrl_r** this element contains the parameters of an M function/H function [▶ 178] if the technology function type is HLI_INTF_M_FKT or HLI_INTF_H_FKT S function [▶ 180] (spindle) if the technology function type is HLI_INTF_SPINDEL		
Data type	ARRAY [0.. HLI_TECH_UNIT_AX_MAXIDX] OF BYTE		
Access	PLC reads		

Acknowledgement	
ST element	**.done_w**
Description	By setting element done_w to TRUE, the PLC signals to the CNC that the technology information was executed.
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	PLC sets the value to TRUE when the technology function was executed. The CNC sets the value to a FALSE before a new command.

6.5.2 Data of a channel-specific technology control unit

Data of a technology function, channel			
Description	A technology control unit contains elements for commanding, acknowledging and transferring any required parameters.		
Data type	TECHNO_UNIT_CH		
ST path	**Default synchronisation:** gpCh[*channel_idx*]^.techno_unit_std_sync[*tech_unit_idx*] **Cross-block synchronisation:** gpCh[*channel_idx*]^.techno_unit_late_sync[*tech_unit_idx*]		
Job			
ST element	**.please_rw**		
Description	By setting please_rw, the CNC signals to the PLC that the technology control unit is to be executed.		
Data type	BOOL		
Value range	[TRUE, FALSE]		
Special features	**Consumption data item**		
Access	The CNC refreshes the data of the technology function only if this element is FALSE. After updating, the CNC sets this element to TRUE and so element **done_w** is set to FALSE. The PLC reads the data of the technology function if this element has the value TRUE. After the data is transferred, the PLC sets the value to FALSE.		
Function type			
ST element	**.fkt_ctrl_r**		
Description	The type of technology function is transferred in fkt_ctrl_r.		
Data type	UINT		
Value range	**Value**	**Constant**	**Function**
	1	HLI_INTF_M_FKT	M function
	2	HLI_INTF_H_FKT	H function
	3	HLI_INTF_SPINDEL	S function
	4	HLI_INTF_TOOL	T function
Access	PLC reads		

Parameter	
ST element	**.attribut_r.data**
Description	Depending on the content of the element **fkt_ctrl_r** this element contains the parameters of an M function/H function [▶ 178] if the technology function type is HLI_INTF_M_FKT or HLI_INTF_H_FKT S function [▶ 180] (spindle) if the technology function type is HLI_INTF_SPINDEL T function [▶ 183] if the technology function type is HLI_INTF_TOOL
Data type	ARRAY [0.. HLI_TECH_UNIT_CH_MAXIDX] OF BYTE
Access	PLC reads
Acknowledgement	
ST element	**.done_w**
Description	By setting element done_w to TRUE, the PLC signals to the CNC that the technology information was executed.
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	PLC sets the value to TRUE when the technology function was executed. The CNC sets the value to a FALSE before a new command.

6.6 Data of the technology functions

6.6.1 Data of the M/H function

Data of the M/H function	
Description	The function number and the execution time are transferred as additional parameters of an M or H function.
Data type	HLI_M_H_PROZESS
ST Path	pMHProcess: POINTER TO HLI_M_H_PROZESS; **Channel-specific, default synchronisation:** **pMHProcess** := ADR(gpCh[*channel_idx*]^.techno_unit_std_sync[*tech_unit_idx*].attribut_r.data[0]); **Channel-specific, cross-block synchronisation:** **pMHProcess** := ADR(gpCh[*channel_idx*]^.techno_unit_late_sync[*tech_unit_idx*].attribut_r.data[0]);
Access	PLC is reading

Number of the M function/H function	
Description	Number of the M or H function. This corresponds to the number for an M or H function programmed in the NC program.

	Example: For instance, if M4711 was programmed.
ST Element	**.nr**
Data type	UDINT
Access	FktNr: UDINT; FktNr := pMHProcess^.**nr**

Probable execution time M/H function	
Description	Contains values configured for the channel parameters in the entry P-CHAN-00040 or P-CHAN-00026 (m_prozess_zeit[]). These values can be used to implement timeout monitoring for technology functions in the PLC application. or If synchronisation type MOS_TS is configured, the sample time offset of the M/H technology function is indicated in this element. It is calculated and output by the NC kernel.
ST Element	**.time**
Data type	DINT
Unit	µs
Access	Time : DINT; Time := pMHProcess^.**time**
Characteristics	Re. 1.): For spindle-specific M functions the corresponding parameter is called m*X_prozess_zeit where [*X = 3, 4, 5, 19]. Example: The parameter for the M3 function of a spindle is: spindel[*index*].m3_prozess_zeit.

Block number of the M/H function	
Description	Block number from the NC program to specify the program line in the NC program in which the M or H function was programmed.
ST Element	**.satz_nr**
Data type	UDINT
Access	BlockNr : UDINT; BlockNr := pMHProcess^.**satz_nr**

Program line number of the M/H function	
Description	Number of the NC program line in which the M/H function was programmed.
ST Element	**.prg_zeilen_nr**
Data type	UDINT
Access	PrgLineNr: UDINT; PrgLineNr := pMHProcess^.**prg_zeilen_nr**

Number as additional information	
Description	A number assigned to an M or H function in the NC program and transferred when the M/H function is output to the interface. See [PROG//M/H- functions with additional information].

ST Element	**.add_nr_r**
Data type	DINT
Access	AddNr : DINT; AddNr := pMHProcess^.**add_nr_r**

Number of cross-block synchronised technology functions not acknowledged by the PLC	
Description	Number of technology functions not yet acknowledged by the PLC but with cross-block synchronisation and waiting at the interface. The number contains all types of technology functions.
ST Element	**.nr_late_sync**
Data type	UINT
Access	NrLateSync : UINT; NrLateSync := pMHProcess^.**nr_late_sync**

Synchronisation type of the M/H function	
Description	Synchronisation type of an M or H function. This value corresponds to the values defined in the channel parameters P-CHAN-00041 and P-CHAN-00027 for a particular technology function.
ST Element	**.synch_ctrl**
Data type	DWORD
Access	SynchCtrl: DWORD; SynchCtrl := pMHProcess^.**synch_ctrl**
Characteristics	**This datum is not available in all PLC environments.**

6.6.2 Data of the S function

When M functions (M03, M04, M05, M19) are programmed in the NC program that refers to a spindle, the technology function data is saved as an S function in the associated **axis-specific** HLI area.

Data of the S function	
Description	The parameters of an S function are contained in the SProzess structure.
Data type	HLI_S_PROZESS
ST Path	pSProcess : POINTER TO HLI_S_PROZESS; **Axis-specific, default synchronisation:** **pSProcess** := ADR(gpAx[*axis_idx*]^.techno_unit_std_sync[*tech_unit_idx*].attribut_r.data[0]); **Axis-specific, cross-block synchronisation:** **pSProcess** := ADR(gpAx[*axis_idx*]^.techno_unit_late_sync[*tech_unit_idx*].attribut_r.data[0]);
Access	PLC is reading

Command position of spindle with M19	
Description	Command position for spindle position with M19
ST Element	**.pos**
Data type	DINT
Unit	10^{-4} °
Access	ActivePosition: DINT; ActivePosition := pSProcess^.**pos**;

Programmed spindle speed	
Description	Programmed spindle speed
ST Element	**.rev**
Data type	DINT
Unit	10^{-3}°/s
Access	PrgRevolution: DINT; PrgRevolution: = pSProcess^.**rev**;

Probable execution time	
Description	Time probably required to process an S function.
ST Element	assigns the definition of the main spindle.**time**
Data type	UDINT
Unit	1 µs
Access	ExpectedTime : UDINT; ExpectedTime:= pSProcess^.**time**;

Number of the M function of the spindle indexing function		
Description	Number of the spindle indexing function (M03, M04, M05)	
ST Element	**.move_cmd**	
Data type	UINT	
Value range	**Value**	**M function**
	3	M03
	4	M04
	5	M05
Access	MoveCmdNum : UINT; MoveCmdNum := pSProcess^.**move_cmd**;	

Number of the M function of the spindle positioning function	
Description	Number of the spindle positioning function (M19)
ST Element	**.pos_cmd**

Data type	UINT
Value range	0 no M function specified 19 stands for M19
Access	PosCmdNum: UINT; PosCmdNum := pSProcess^.**pos_cmd**;

Axis number	
Description	Unique system-wide number of a logical axis/spindle
ST Element	**.log_achs_nr**
Data type	UINT
Value range	[1, MAX_UNS16] In TwinCAT normally [1, gNrAX]
Access	LogAxisNum: UINT; LogAxisNum := pSProcess^.**log_achs_nr**;

Number of cross-block synchronised technology functions not acknowledged by the PLC	
Description	Number of technology functions not yet acknowledged by the PLC but with cross-block synchronisation and waiting at the interface. The number contains all types of technology functions.
ST Element	**.nr_late_sync**
Data type	UINT
Access	NrLateSync : UINT; NrLateSync := pSProcess^.**nr_late_sync**

6.6.3 Data of the T function

Data of the T function	
Description	All data for a tool change is combined in a T function.
Data type	HLI_T_PROZESS
ST path	pTProcess : POINTER TO HLI_T_Prozess; **Channel-specific, default synchronisation:** **pTProcess** := ADR(gpCh[*channel_idx*]^.techno_unit_std_sync[*tech_unit_idx*].attribut_r.data[0]); **Channel-specific, cross-block synchronisation:** **pTProcess** := ADR(gpCh[*channel_idx*]^.techno_unit_late_sync[*tech_unit_idx*].attribut_r.data[0]);
Access	PLC reads
Tool identification	
Description	The structure contains the identification number of the tool. There may also be identification numbers of identical or similar tools. A description of the structure format is contained in User data of tool identification [▶ 184].
ST element	**.id**
Data type	HLI_TOOL_ID [▶ 183]
Access	HLIToolId : HLI_TOOL_ID; HLIToolId := pTProcess^.**id**;
Information on the tool unit	
Description	Information supplied to the PLC referring to a tool unit.
ST element	**.add_info_r[ ]**
Data type	ARRAY [0..HLI_ADD_INFO_MAXIDX] OF UDINT
Access	AddInfo : UDINT; AddInfo := pTProcess^.**add_info_r[X]** where *X* = [0, HLI_ADD_INFO_MAXIDX]
Number of cross-block synchronised technology functions not acknowledged by the PLC	
Description	Number of technology functions not yet acknowledged by the PLC but with cross-block synchronisation and waiting at the interface. The number contains all types of technology functions.
ST Element	**.nr_late_sync**
Data type	UINT
Access	NrLateSync : UINT; NrLateSync := pTProcess^.**nr_late_sync**

6.6.3.1 User data of tool identification

With the definition of pTProcess [▶ 183] from the previous section and the definition pHLIToolId: POINTER TO HLI_TOOL_ID gilt:

pHLIToolId := ADR(pTProcess^.id);

and is used as shown in the table below.

Number of the tool to be replaced	
Description	Number of the tool to be replaced
ST element	**.basic**
Data type	DINT
Access	ToChangeToolNum : DINT; ToChangeToolNum := pHLIToolId^.**basic**;
Number of a sister tool	
Description	Number of an identical sister tool
ST element	**.sister**
Data type	DINT
Access	SisterToolNum : DINT; SisterToolNum := pHLIToolId^.**sister**;
Number of a variant tool	
Description	Number of a similar variant tool
ST element	**.variant**
Data type	DINT
Access	VariantToolNum: DINT; VariantToolNum:= pHLIToolId^.**variant**;
Sister tool valid	
Description	Validity identifier for the sister tool.
ST element	**.sister_valid**
Data type	BOOL
Access	SisterToolValid : BOOL; SisterToolValid := pHLIToolId^.**sister_valid**;
Variant tool valid	
Description	Validity identifier for the variant tool.
ST element	**.variant_valid**
Data type	BOOL
Access	VariantToolValid : BOOL; SisterToolValid := pHLIToolId^.**variant_valid**;

7 External variables / V.E variables

Data can be exchanged between the NC program and the PLC via the HLI using external variables. Each channel has a separate data area for external variables known only in the channel. In addition, there is a cross-channel global data area which all channels can access.

At the PLC end, the data areas for the external variables are represented as an ARRAY OF UDINT. The index of the individual array elements starts with the value 0.

Each external variable always occupies one memory block of HLI_EXT_VAR_STR_MAXIDX + 1 (24) bytes, irrespective of its data type. If an array of external variables was defined, the individual variables are saved packed in the memory area (several variables per memory block). Here, several consecutive memory blocks may be used depending on the array size.

When the PLC accesses external variables, the index of the variables in the memory area of the external variables must first be determined:

Example for calculating an index:

4. Variable (VarNr = 4):

Offset = (VarNr –1)* (HLI_EXT_VAR_STR_MAXIDX + 1) / 4 + 1

A variable with the index = 3 results in an offset in the memory of 13.

The memory area must then be accessed depending on the actual data type of the external variables. All information required for this is contained in the configuration list of the external variables.

Please refer to the documentation [EXTV] for further details on external variables.

In the TwinCAT runtime environment, the number of external variables can be configured in the System Manager.

An example program to access external variables is available under the name or HLI-Ve1.pro.

External variable	
Description	Memory area for data exchange between NC program and PLC
Data type	**CNC global** POINTER TO HLI_GLOB_V_E_DATA**:** *channel-specific:* ARRAY[0..HLI_SYS_CH_MAXIDX] OF POINTER TO HLI_CHAN_V_E_DATA
ST path	**CNC global:** gpVEGlobal **channel-specific:** gpVECh[*channel_idx*]
Access	pLRealVal : POINTER TO LREAL; LRealVal : LREAL; VeOffset : DINT; VeOffset := (VarNr –1) * (HLI_EXT_VAR_STR_MAXIDX + 1) / 4 + 1; pLRealVal := gpVEGlobal^.ext_var32[VeOffset]); LRealVal := pLRealVal^;
Special features	Access depends on the data type of the external variables.

8 Processing types

The CNC distinguishes between 5 operation modes. Switchover between these operation modes is performed using the operator interface and/or PLC interface. However, only **one operation mode may be active at any one time**.

The following operation modes are defined:

Operation mode	ST constant	Value	Explanation
Standby	HLI_IMCM_STANDBY_MODE	1	No operation mode is selected. Default after controller start-up.
Automatic mode	HLI_IMCM_AUTOMATIC_MODE	2	The controller can process a complete NC program automatically. Program execution can be interrupted and resumed.
Manual block	HLI_IMCM_MDI_MODE	3	Motions are commanded by the operator computer via a single NC block. The NC block is transferred to the controller in a string and executed via a START command. A motion can be interrupted and resumed.
Manual operation mode	HLI_IMCM_MANUAL_MODE	4	Motions are commanded by peripherals connected directly to the controller (buttons, handwheels).
Homing	HLI_IMCM_REFERENCE_MODE	5	Axes can be referenced. An NC program with the name rpf.nc then starts.

An operation mode may have different states. The states of individual operation modes and their significance for the operation mode are discussed in the sections below.

8.1 State graph of operation modes

Note: There is no state graph for the “Standby“ operation mode.

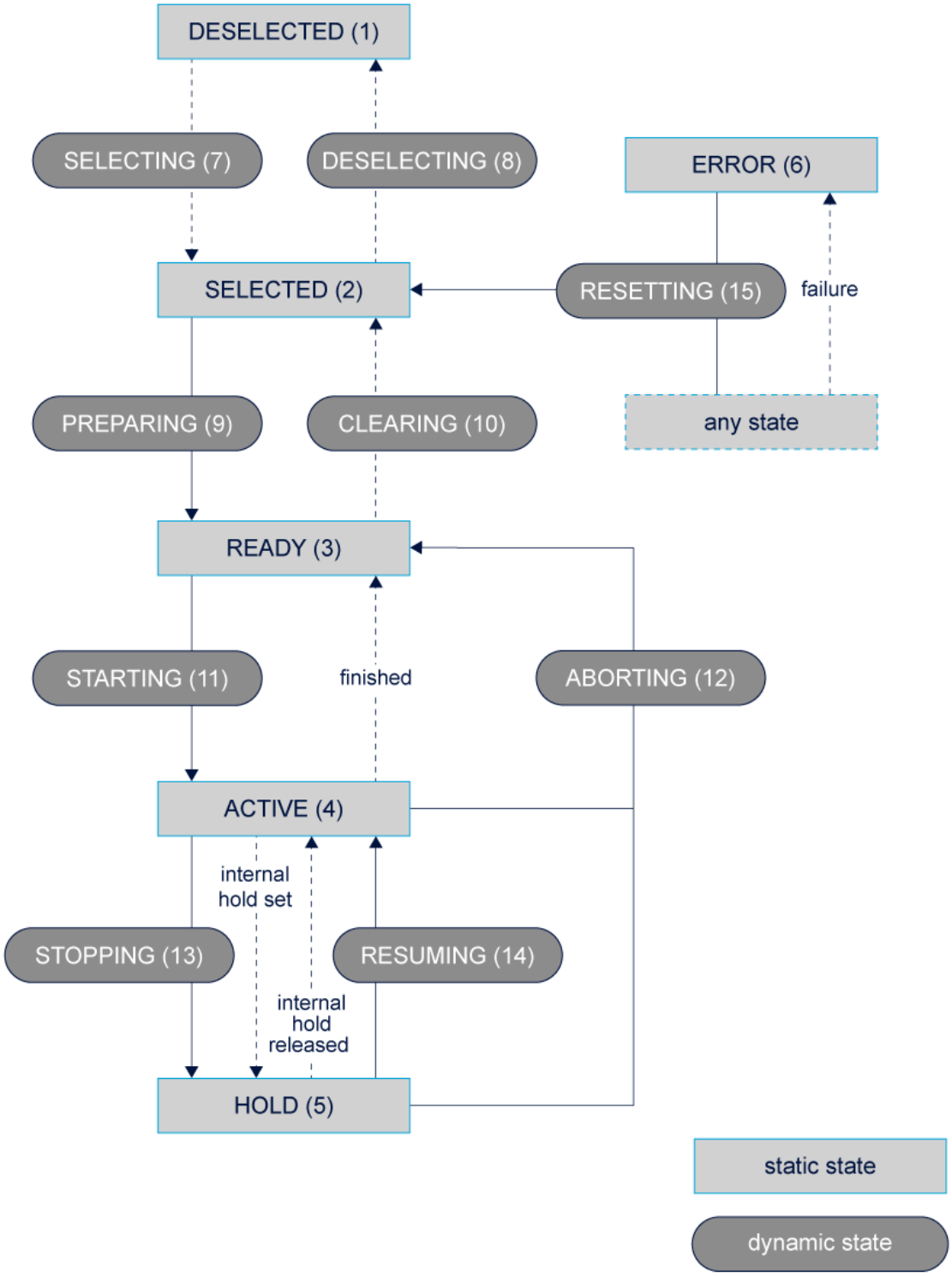


Fig. 27: State graph of an operation mode

Notice

A CNC reset is automatically executed when the target operation mode and target state are commanded if the state change specifies an abort (see “abort”).

The error state is exited by an automatically executed reset if an operation mode was commanded.

In addition, a CNC reset can be forced by the following commands:

a) Select operation mode = STANDBY and state = SELECTED

b) Explicit specification of transition = RESET in the currently active operation mode

c) Specify the target state = RESETTING (see below)

8.1.1 States of operation mode: Automatic mode

State	ST constant	Value	Description
DESELECTED	HLI_IMCM_PROCESS_DESELECTED	1	Operation mode is deselected
SELECTED	HLI_IMCM_PROCESS_SELECTED	2	Automatic mode is selected
READY	HLI_IMCM_PROCESS_READY	3	NC program is selected
ACTIVE	HLI_IMCM_PROCESS_ACTIVE	4	NC program is running
HOLD	HLI_IMCM_PROCESS_HOLD	5	NC program is interrupted (see also Feedhold [▶ 127]).
ERROR	HLI_IMCM_PROCESS_ERROR	6	An error occurred while the NC program is executed.

The NC program name must be transferred where there is a transition from SELECTED to READY.

8.1.2 States of operation mode: Manual block

State	ST constant	Value	Description
DESELECTED	HLI_IMCM_PROCESS_DESELECTED	1	Manual block mode is deselected.
SELECTED	HLI_IMCM_PROCESS_SELECTED	2	Manual block mode is selected. An NC block (NC blocks) can be programmed.
READY	HLI_IMCM_PROCESS_READY	3	MDI block(s) is/are selected. CNC accepted the programmed NC block(s).
ACTIVE	HLI_IMCM_PROCESS_ACTIVE	4	The NC block(s) is/are running.
HOLD	HLI_IMCM_PROCESS_HOLD	5	NC block(s) is/are stopped (see also feedhold). Feedhold [▶ 127]).
ERROR	HLI_IMCM_PROCESS_ERROR	6	Error state

The manual block (string) must be transferred when there is a transition from SELECTED to READY.

8.1.3 States of operation mode: Manual operation mode

State	ST constant	Value	Description
DESELECTED	HLI_IMCM_PROCESS_DESELECTED	1	Operation mode is deselected.
SELECTED	HLI_IMCM_PROCESS_SELECTED	2	Operation mode is selected. (Basic state).
READY	HLI_IMCM_PROCESS_READY	3	Manual mode is programmed.
ACTIVE	HLI_IMCM_PROCESS_ACTIVE	4	Manual mode is running.
HOLD	HLI_IMCM_PROCESS_HOLD	5	Manual mode is stopped (see also feedhold). Feedhold [▶ 127]).
ERROR	HLI_IMCM_PROCESS_ERROR	6	Error state

8.1.4 States of operation mode: Homing

State	ST constant	Value	Description
DESELECTED	HLI_IMCM_PROCESS_DESELECTED	1	Operation mode is deselected.
SELECTED	HLI_IMCM_PROCESS_SELECTED	2	Operation mode is selected. (Basic state).
READY	HLI_IMCM_PROCESS_READY	3	Homing is programmed (e.g. with respect to sequence).
ACTIVE	HLI_IMCM_PROCESS_ACTIVE	4	Homing is running.
HOLD	HLI_IMCM_PROCESS_HOLD	5	Homing is stopped (see also feedhold). Feedhold [▶ 127]).
ERROR	HLI_IMCM_PROCESS_ERROR	6	Error state

Axes to be referenced must be transferred during the transition from SELECTED to READY. If no string is transferred, the default homing program rpf.nc is started.

8.2 Control commands/status information for operation modes

The control unit described below contains data to command a change of operation mode and to poll the current state of operation mode management.

8.2.1 Control Unit

Operation mode	
Description	Control unit to switch over the operation mode and poll the current state of operation mode management, including flow control of user data.
Data type	MC_CONTROL_MCM_MODE_STATE_UNIT, see description Control Unit with usage check [▶ 14]
ST path	gpCh[*channel_idx*]^.channel_mc_control.**mode_and_state**
Commanded, requested data	
ST element	**.command_w** **.request_r**
Data type	HLI_PROC_TRANS_TO_MODE_STATE [▶ 192]
Access	PLC writes command_w and reads request_r
Return data	
ST element	**.state_r**
Data type	HLI_IMCM_MODE_STATE [▶ 195]
Access	PLC reads
Flow control of commanded value	
ST element	**.command_semaphor_rw**
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	CNC accepts the commanded data if this element has the value TRUE and sets this element to the value FALSE after complete acceptance of the data. PLC can write data for commanding if this element has the value FALSE. The PLC sets this element to the value TRUE if all data to be commanded is written.
Flow control of requested data	
ST element	**.request_semaphor_rw**
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	CNC writes the data requested by the GUI if this element is FALSE and then sets this element to TRUE. PLC reads the data requested by the GUI if this value is TRUE. After the PLC fully accepts the data, the PLC sets this element to FALSE.

8.2.2 User data

8.2.2.1 Requested and commanded user data

Initial operation mode		
Description	Operation mode from which the operation mode is to be changed.	
ST path	**Commanded value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.command_w **Requested value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.request_r	
Commanded, requested value		
ST element	**.from_mode**	
Data type	UDINT	
Value range	**Value**	**Constant**
	1	HLI_IMCM_STANDBY_MODE
	2	HLI_IMCM_AUTOMATIC_MODE
	3	HLI_IMCM_MDI_MODE
	4	HLI_IMCM_MANUAL_MODE
	5	HLI_IMCM_REFERENCE_MODE
Special features	This element does not need to be programmed when the operation mode is changed. However, if a value is specified, a check is conducted when the operation mode is changed to establish whether the CNC is actually in the specified operation mode. A warning is issued if this is not the case.	

Initial state of operation mode when the operation mode is changed		
Description	State within the operation mode from which the state switchover is to occur.	
ST path	**Commanded value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.command_w **Requested value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.request_r	
Commanded, requested value		
ST element	**.from_state**	
Data type	UDINT	
Value range	Value	Significance / constant
	0	no significance
	1	HLI_IMCM_PROCESS_DESELECTED
	2	HLI_IMCM_PROCESS_SELECTED
	3	HLI_IMCM_PROCESS_READY
	4	HLI_IMCM_PROCESS_ACTIVE
	5	HLI_IMCM_PROCESS_HOLD
	6	HLI_IMCM_PROCESS_ERROR
Special features	This element does not need to be programmed when the operation mode is changed. However, if a value is specified, a check is conducted when the operation mode is changed to establish whether the CNC is actually in the specified operation mode. A warning is issued if this is not the case.	

Target operation mode when the operation mode is switched over	
Description	Operation mode to which the system is to switch.
ST Path	**Commanded value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.command_w **Requested value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.request_r
Commanded, requested value	
ST Element	**.to_mode**
Data type	UDINT
Value range	See “Operation mode actual value on operation mode change” → Value range [▶ 193]

Target state when operation mode is changed	
Description	Target state within the target operation mode.
ST path	**Commanded value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.command_w **Requested value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.request_r
Commanded, requested value	
ST element	**.to_state**
Data type	UDINT
Value range	See "Actual value of operation mode state for operation mode change" → value range [▶ 192]

Release Note

If the target state = RESETTING = 15 is specified, a CNC reset is executed explicitly.

This function is available as of the following Build:

V2.10.1033.01 and higher

V2.10.1507.02 and higher

V2.10.1800.04 and higher

Parameters for operation mode change	
Description	Parameters for operation mode change. It may be necessary to specify parameters when commanding an operation mode change to ensure the successful change to a specific state of an operation mode. These parameters are saved in this element.
ST path	**Commanded value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.command_w **Requested value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.request_r.
Commanded, requested value	
ST element	**.parameter**
Data type	STRING(HLI_IMCM_MODE_STATE_PARAM_STRL)
Special features	The string for the AUTOMATIC operation mode is limited to a maximum of 83 characters. It is advisable to use the parameters of file paths with longer strings. See P-CHAN-00401 to P-CHAN-00404.

If an operation mode change is commanded, it may be necessary to specify a parameter so that commanding can be executed successfully. The table below lists the cases where this is necessary and what type of parameter needs to be transferred.

If one of the listed state transitions is involved in the operation mode and state change, the corresponding parameter must be specified. Use the "State graph of other operation modes" [▶ 187] to determine whether one of the state transitions listed below is executed in the case of a commanded operation mode change.

Meaning of the parameter string in conjunction with the operation mode

The NC kernel only adopts the content of the “parameter” structure element if a state transition is executed from SELECTED to one of the states READY, ACTIVE or HOLD.

Operation mode setpoint	Parameter
Automatic mode	The NC program name as character string.
Manual block	NC block (blocks)
Manual mode	• no parameter: all axes are activated (G200) • explicit activation of specific axes with G200[Axis_1, ...] • General NC block sequence comprising several lines, e.g. activate a kinematic #KIN ID[1] RN G200 RN: String denoting line break in IEC61131-3
Homing	• no parameter: NC program **rpf.nc** is started • explicit selection of axes by manual block (e.g.: **G74 X1 Z2**)

Channel number	
Description	Number of the channel whose operation mode is to be switched over.
ST path	**Commanded value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.command_w **Requested value** gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.request_r.
Commanded, requested value	
ST element	**.channel_number**
Data type	UDINT
Special features	**Not used (only for compatibility with the HÜMNOS standard).**

8.2.2.2 Status information

Operation mode actual value	
Description	Actual value of the operation mode.
ST path	gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.state_r.**mode**
Data type	UDINT
Value range	See “Operation mode actual value on of operation mode change” → Value range [▶ 192]

Actual value of the operation mode state	
Description	Actual value of the operation mode state.
ST path	gpCh[*channel_idx*]^.channel_mc_control.mode_and_state.state_r.**state**
Data type	UDINT
Value range	See “Actual value of operation mode state for operation mode change” → value range [▶ 192]

9 Manual operation mode

The manual mode function permits the external control of individual axes with physical elements of manual mode (handwheel, continuous jog buttons) via the HLI.

The following three options are available for axis motion:

- **Handwheel function:** : any path at any velocity by specifying handwheel increments.
- **Cont. jog mode:** any path at defined velocity when a button is operated.
- **Incr. jog mode:** defined path at defined velocity when a button is operated.

It is possible to assign the manual mode elements to one single or multiple axis and to change the parametrisation (e.g. the incremental jogging distance) during operation. The control element is assigned to the logical axes via the logical axis number. The diagram below shows as an example of an assignment of manual mode elements to CNC axes.

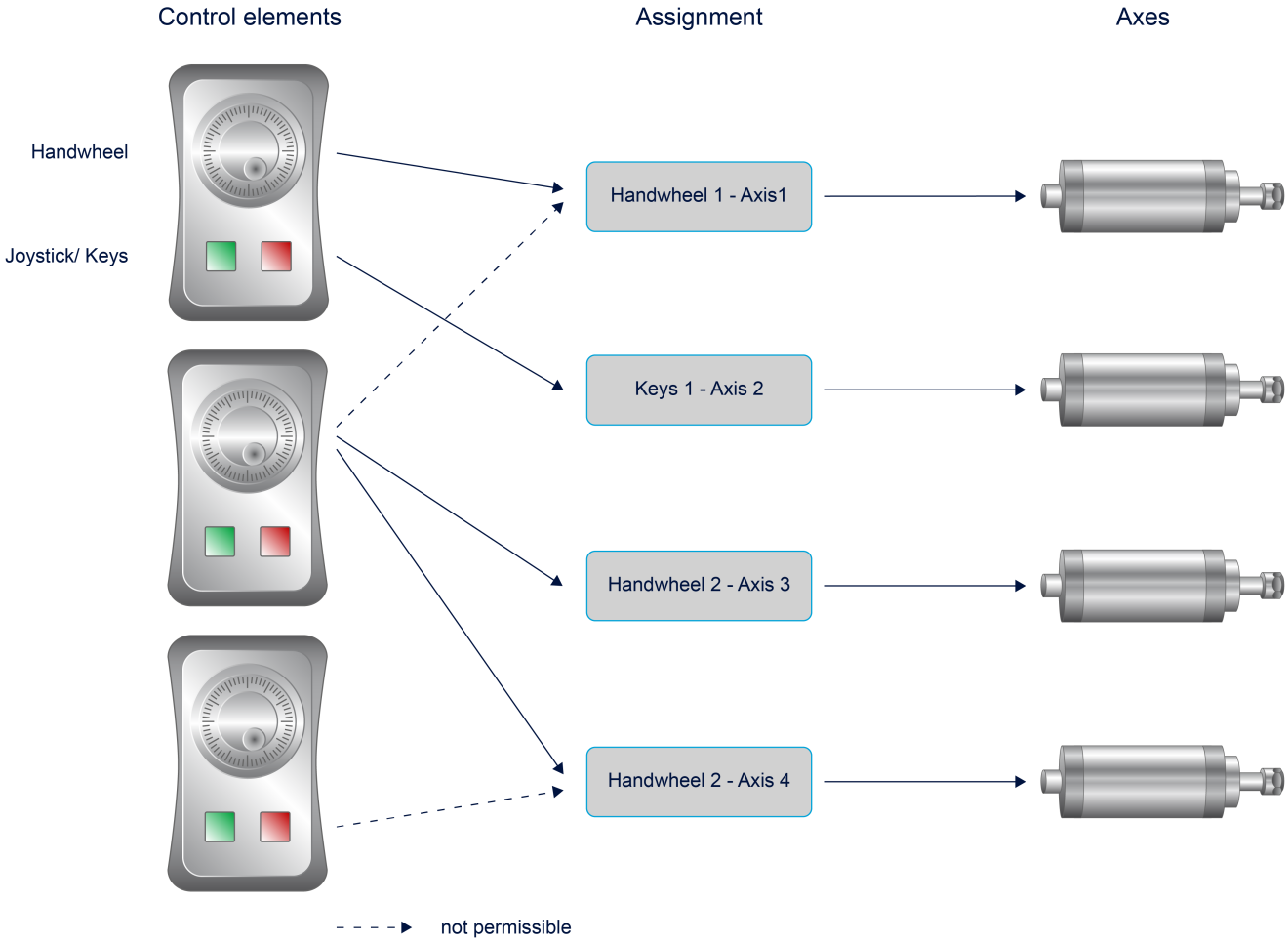


Fig. 28: Control elements and assignment

In manual mode, each axis can assume three different states:

- IDLE
- READY
- ENABLED

The following steps are required to used manual mode for an axis:

- Activating the READY state for axes

- Parametrising the manual mode type
- Activating manual mode (transition to ACTIVE state)
- Transferring control element actions (button presses, handwheel counter state) to the CNC.

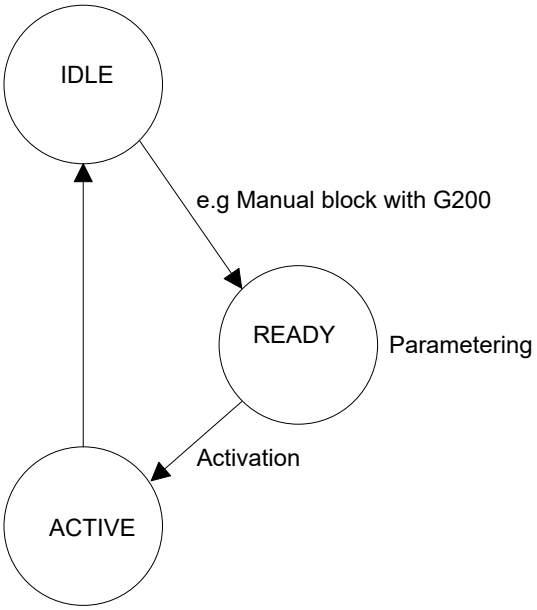


Fig. 29: Manual mode state transitions

Transition to READY state

The transition to READY state can be executed by the following actions:

- Explicit switchover of operation mode via HLI or GUI
- Programming of G200/G201 in the NC program or manual block

The desired manual mode type can then be parameterised in the READY state. See also the section Parameterising manual mode

Transition to READY state

The axis transits to the ACTIVE state as soon as a control element was assigned. Details are contained in the section Activating control elements [▶ 201].

Transferring control element actions

In the ACTIVE state, control element actions can now be transferred to the CNC to move the axis.

Deselecting manual mode

The ACTIVE state of an axis is deselected when the axis is assigned to the control element 0 or If a reset was executed.

Information on the axis state regarding manual mode is contained in the structure HLI_HB_AXIS_DISPLAY_DATA (see Section Status information of manual mode [▶ 199]).

9.1 Status information of manual mode

State of manual mode		
Description	The manual operation mode is in one of the states described below.	
Signal flow	CNC → PLC	
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].hb_display_r.**state**	
Data type	UINT	
Value range	**Value**	**Meaning**
	0	Operation mode deactivated
	1	Operation mode enabled in NC program via G200, G201 but not assigned to a control element.
	2	Operation mode enabled in NC program via G200, G201, assigned to control element.
Access	PLC reads	

Operation mode of manual mode		
Description	Look ahead waits to receive request from an axis.	
Signal flow	CNC → PLC	
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].hb_display_r.**operation_mode**	
Data type	UINT	
Value range	**Value**	**Operation mode**
	0	no operation mode selected
	1	Handwheel mode
	2	Continuous jog mode
	3	jog mode
Access	PLC reads	

Control element number	
Description	Logical number of the control element currently linked to the axis in question.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].hb_display_r.**control_element**
Data type	UINT
Access	PLC reads

Handwheel resolution	
Description	Resolution of the handwheel which is linked as control element to the axis in question.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].hb_display_r.**hr_aufloesung**
Data type	DINT
Unit	Increments per handwheel pulse
Access	PLC reads

Path velocity in continuous jog mode	
Description	Path velocity of the axis in question when moved in continuous jog mode.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].hb_display_r.**tipp_geschw**
Data type	DINT
Unit	1 µm/s
Access	PLC reads

Path velocity in incremental jog mode	
Description	Path velocity of the axis in question when moved in continuous jog mode.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].hb_display_r.**jog_geschw**
Data type	DINT
Unit	1 µm/s
Access	PLC reads

Motion path in incremental jog mode	
Description	Motion path of the axis in question for each button press when moved in incremental jog mode.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.bahn_state.coord_r[*axis_idx*].hb_display_r.**jog_weg**
Data type	DINT
Unit	0,1 µm
Access	PLC reads

9.2 Control commands of manual mode

The individual manual operation modes are controlled by specific control units. These control units contain data on the flow control of user data and the user data itself. User data generally comprises elements of a structure.

9.2.1 Activating control elements for manual mode

After an axis was prepared for manual mode by G200/G201, this axis can be assigned a logical control element (button/handwheel) by a parameter set. This takes place during activation of an axis.

The default value settings are defined in the axis-specific parameter lists:

```
# 1 handwheel, 2 continuous, 3 incremental
handbetrieb.default.operation_mode   2 # 2 continuous,
handbetrieb.default.control_element  1 # logical handwheel/key
```

The default value used is the 0th parameter set (index = 0) of the manual mode parameters.

Every time the manual mode (see Operation mode types) or G200/G201 is reselected, the last known setting of the axes (operation mode, link to control element and parameter set) is restored.

Activating control elements with manual mode	
Description	Control unit to manage data to activate a control element and assign it to an axis in manual mode, including flow control of user data.
Data type	MC_CONTROL_HB_ACTIVATION_UNIT, see description Control Unit mit Verbrauchskontrolle [▶ 14]
Special features	This control unit can only be used if the CNC is in manual mode or if the axes are enabled for manual mode by an explicit G200/G201. Otherwise, activation is refused with an error message (e.g. 150048 -> “Impermissible selection of operation mode in continuous jog mode”).
ST path	gpCh[*channel_idx*]^.hb_mc_control.**activation**
Access	PLC reads **request_r** and writes **command_w** + **enable_w**
Commanded, requested data	
ST element	**.command_w** **.request_r**
Data type	HLI_HB_ACTIVATION [▶ 203], for description, see Section User data on activation [▶ 203]
Flow control commanded data	
ST element	**.command_semaphor_rw**
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	CNC accepts the commanded data if this element has the value TRUE and sets this element to the value FALSE after complete acceptance of the data. PLC can write data for commanding if this element has the value FALSE. The PLC sets this element to the value TRUE if all data to be commanded is written.
Flow control of requested data	
ST element	**.request_semaphor_rw**
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	CNC writes the data requested by the GUI if this element is FALSE and then sets this element to TRUE. PLC reads the data requested by the GUI if this value is TRUE. After the PLC fully accepts the data, the PLC sets this element to FALSE.
Redirection	
ST path	**.enable_w**

9.2.1.1 User data on activation

Axis number	
Description	Unique system-wide number of a logical axis. A control element is assigned to the specified logical axis with which the axis is to be moved in manual mode.
ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.activation.command_w.**log_achs_nr** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.activation.request_r.**log_achs_nr**
Data type	UINT
Value range	[1, MAX_UNS16] In TwinCAT normally [1, gNrAX]

Control element number	
Description	Number of the logical control element to be assigned to the logical axis.
ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.activation.command_w.**control_element** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.activation.request_r.**control_element**
Data type	UINT
Value range	**When continuous and incremental jog mode is activated:** one of the values which are defined as logical button pair numbers in the configuration list hand_mds.lis for the characteristics tasten_data[*X*].log_tasten_nr. **When handwheel mode is activated:** one of the values which are defined as logical handwheel numbers in the configuration list hand_mds.lis for the characteristics hr_data[0].log_hr_nr.
Peculiarities	If 0 is specified as the logical number, the current operation mode of an axis is deselected.

Manual operation mode		
Description	Manual operation mode to be assigned to the logical axis.	
ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.activation.command_w.**operation_mode** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.activation.request_r.**operation_mode**	
Data type	UINT	
Value range	**Value**	**Operation mode**
	0	no operation mode, current operation mode selected
	1	Handwheel mode
	2	Continuous jog mode
	3	jog mode

Manual operation mode parameter set	
Description	Specifies the index of the parameter set to be used for the manual mode.
ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.activation.command_w.**parameter_index** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.activation.request_r.**parameter_index**
Data type	UINT
Value range	[0; 2]
Peculiarities	The first value set in the parameter table (index = 0) is overwritten by the PLC interface when individual parameters are specified. The remaining parameter sets are not changed. This means, they correspond to the values specified in the axis-specific parameter lists.

Note:

The currently active axis can be deactivated by deselecting the operation mode or by assigning the control element 0.

9.2.2 Parametrising of manual mode

Basically, every manual mode can be specifically parametrised. For example, the motion velocities or incremental step can be defined per axis. The default values here are pre-assigned in the axis-specific parameter lists.

```
handbetrieb.hr.aufl[0]                  10
handbetrieb.hr.aufl[1]                  20
handbetrieb.hr.aufl[2]                  30
#
handbetrieb.tipp.geschw[0]              30000
handbetrieb.tipp.geschw[1]              30000
handbetrieb.tipp.geschw[2]              40000
handbetrieb.tipp.vb_eilgang             40000
#
handbetrieb.jog.weg[0]                  1000
handbetrieb.jog.weg[1]                  2000
handbetrieb.jog.weg[2]                  3000
handbetrieb.jog.geschw[0]               30000
handbetrieb.jog.geschw[1]               30000
handbetrieb.jog.geschw[2]               60000
```

In addition, an individual parameter value can be pre-assigned via the PLC interface. This parameter value is the first element (index 0) stored in the table of default parameter. It can be selected when an axis is activated.

Parameters can be changed at any time but they only become effective when an axis is activated (see above). When an axis is activated, the number (index) of the desired parameter set is specified in addition to the operation mode and the control element.

9.2.2.1 Continuous jog mode (by button press)

9.2.2.1.1 Control Unit

Parameterising continuous jog mode in manual mode	
Description	Control unit to manage data for parameterising continuous jog mode in manual mode, including flow control of user data.
Data type	MC_CONTROL_HB_TIP_PARAM_UNIT, see description Control Unit with check usage [▶ 14]
Special features	The manual mode parameter can be written at any time and it is stored internally in a table under the index 0. This value is only becomes effective in the corresponding operation mode when the axis is activated.
Access	PLC reads **request_r** and writes **command_w** + **enable_w**
ST path	gpCh[*channel_idx*]^.hb_mc_control.**tip_parameter**
Commanded, requested data	
ST element	**.command_w** **.request_r**
Data type	MC_CONTROL_HB_TIP_PARAM_UNIT [▶ 207], for description see the section on User Data [▶ 207]
Flow control commanded data	
ST element	**.command_semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC accepts the commanded data if this element has the value TRUE and sets this element to the value FALSE after complete acceptance of the data. PLC can write data for commanding if this element has the value FALSE. The PLC sets this element to the value TRUE if all data to be commanded is written.
Flow control of requested data	
ST element	**.request_semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC writes the data requested by the GUI if this element is FALSE and then sets this element to TRUE. PLC reads the data requested by the GUI if this value is TRUE. After the PLC fully accepts the data, the PLC sets this element to FALSE.
Redirection	
ST path	**.enable_w**

9.2.2.1.2 User data

Axis number	
Description	Unique system-wide number of a logical axis. The specified logical axis is assigned the velocity at which it will be moved in manual mode in continuous jog mode.
ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.tip_parameter.command_w.**log_achs_nr** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.tip_parameter.request_r.**log_achs_nr**
Data type	UINT
Value range	[1, MAX_UNS16] In TwinCAT normally [1, gNrAX]

Continuous joy velocity	
Description	Velocity assigned to the logical axis in continuous jog mode.
ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.tip_parameter.command_w.**speed** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.tip_parameter.request_r.**speed**
Data type	UDINT
Unit	1 µm/s

9.2.2.2 Incremental jog mode (by button press)

9.2.2.2.1 Control Unit

Parameterising incremental jog mode in manual mode	
Description	Control unit to manage data for parameterising incremental jog mode in manual manual, including flow control of user data.
Data type	MC_CONTROL_HB_JOG_PARAM_UNIT, see description Control Unit with usage check [▶ 14]
Special features	The manual mode parameter can be written at any time and it is stored internally in a table under the index 0. This value is only becomes effective in the corresponding operation mode when the axis is activated.
ST path	gpCh[*channel_idx*]^.hb_mc_control.**jog_parameter**
Access	PLC reads **request_r** and writes **command_w** + **enable_w**
Commanded, requested data	
ST element	**.command_w** **.request_r**
Data type	MC_CONTROL_HB_JOG_PARAM_UNIT [▶ 209], for description see the section on User Data [▶ 209]
Flow control commanded data	
ST element	**.command_semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC accepts the commanded data if this element has the value TRUE and sets this element to the value FALSE after complete acceptance of the data. PLC can write data for commanding if this element has the value FALSE. The PLC sets this element to the value TRUE if all data to be commanded is written.
Flow control of requested data	
ST element	**.request_semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC writes the data requested by the GUI if this element is FALSE and then sets this element to TRUE. PLC reads the data requested by the GUI if this value is TRUE. After the PLC fully accepts the data, the PLC sets this element to FALSE.
Redirection	
ST element	**.enable_w**

9.2.2.2.2 User data

Axis number	
Description	Unique system-wide number of a logical axis. The specified logical axis is assigned the velocity and incremental step width for each button press to move the axis in manual mode in incremental jog mode.
ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.jog_parameter.command_w.**log_achs_nr** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.jog_parameter.request_r.**log_achs_nr**
Data type	UINT
Value range	[1, MAX_UNS16] In TwinCAT normally [1, gNrAX]

Incremental job velocity	
Description	Velocity to be assigned to the logical axis in incremental jog mode.
ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.jog_parameter.command_w.**speed** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.jog_parameter.request_r.**speed**
Data type	UDINT
Unit	1 µm/s

Incremental jog path	
Description	Path traversed by the logical axis in incremental jog mode each time the incremental jog button is pressed.
ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.jog_parameter.command_w.**distance** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.jog_parameter.request_r.**distance**
Data type	UDINT
Unit	0.1 µm

9.2.2.3 Handwheel mode

9.2.2.3.1 Control Unit

Parameterising handwheel mode in manual mode	
Description	Control unit to manage data for parameterising handwheel mode in manual mode, including flow control of user data.
Data type	MC_CONTROL_HB_HR_PARAM_UNIT, dee description Control Unit with usage check [▶ 14]
Special features	The manual mode parameter can be written at any time and it is stored internally in a table under the index 0. This value is only becomes effective in the corresponding operation mode when the axis is activated.
ST path	gpCh[*channel_idx*]^.hb_mc_control.**hr_parameter**
Access	PLC reads **request_r** and writes **command_w** + **enable_w**
Commanded, requested data	
ST element	**.command_w** **.request_r**
Data type	MC_CONTROL_HB_HR_PARAM_UNIT [▶ 211], for description see the section on User Data [▶ 211]
Flow control commanded data	
ST element	**.command_semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC accepts the commanded data if this element has the value TRUE and sets this element to the value FALSE after complete acceptance of the data. PLC can write data for commanding if this element has the value FALSE. The PLC sets this element to the value TRUE if all data to be commanded is written.
Flow control of requested data	
ST element	**.request_semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC writes the data requested by the GUI if this element is FALSE and then sets this element to TRUE. PLC reads the data requested by the GUI if this value is TRUE. After the PLC fully accepts the data, the PLC sets this element to FALSE.
Redirection	
ST element	**.enable_w**

9.2.2.3.2 User data

Axis number	
Description	Unique system-wide number of a logical axis. The specified logical axis is assigned the handwheel resolution which is the basis for moving the axis in manual mode handwheel mode.
ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.hr_parameter.command_w.**log_achs_nr** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.hr_parameter.request_r.**log_achs_nr**
Data type	UINT
Value range	[1, MAX_UNS16] In TwinCAT normally [1, gNrAX]

Handwheel resolution	

Description	Resolution of axis motion path for one handwheel revolution.
	The internally used total resolution of the axis in 0.1 µm per applied handwheel increment results from the current handwheel resolution in 0.1 µm/increment divided by the physical handwheel resolution in increment/revolution of the handwheel specified.
	
	Manual parameter list:
	`hr_data[0].hr_aufl_z  1000  # incr./rev. - denominator`
	`hr_data[0].hr_aufl_n  14  # incr./rev.. - denominator`
	Programming command (specified here in mm/revolution):
	#HANDWHEEL [AX=X RES1=0.1 RES2=90.2 RES3=0.5] or
	#SET HR [0.1, 90.2, 0.5] X (old syntax)

ST Path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.hr_parameter.command_w.**resolution** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.hr_parameter.request_r.**resolution**
Data type	DINT
Unit	0,1 µm / handwheel revolution

9.2.3 Control elements for manual mode

9.2.3.1 Enforcing a button press

The HLI has an array of similar control units which command button presses in parallel.

Transfer of GUI requests:

To transfer a request received from the GUI to the PLC, the activated control units are checked in the order of their indices whether the request semaphore is free and the first free request element is used to transfer the button press request to the PLC. The PLC must therefore process the request semaphore request_semaphor_rw for all control units for which the PLC set enable_w to TRUE.

9.2.3.1.1 Control Unit

Enforcing a button press in manual mode	
Description	Control unit to manage data to enforce a button press in manual mode, including flow control of user data.
Data type	ARRAY[0..HLI_KEY_MAXIDX] OF MC_CONTROL_HB_KEY_UNIT, see description of Control unit usage check [▶ 14]
Special features	A button press can be signalled at any time but it remains without any effect if the assigned axis is not in continuous/incremental jog mode.
ST path	gpCh[*channel_idx*]^.hb_mc_control.**key[***key_idx***]**
Access	PLC reads **request_r** and writes **command_w** + **enable_w**
Commanded, requested data	
ST element	**.command_w** **.request_r**
Data type	HLI_HB_KEY [▶ 214], for description see the section User data [▶ 214]
Flow control commanded data	
ST element	**.command_semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC accepts the commanded data if this element has the value TRUE and sets this element to the value FALSE after complete acceptance of the data. PLC can write data for commanding if this element has the value FALSE. The PLC sets this element to the value TRUE if all data to be commanded is written.
Flow control of requested data	
ST element	**.request_semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC writes the data requested by the GUI if this element is FALSE and then sets this element to TRUE. PLC reads the data requested by the GUI if this value is TRUE. After the PLC fully accepts the data, the PLC sets this element to FALSE.
Redirection	
ST element	**.enable_w**

9.2.3.1.2 User data

Button number	
Description	Logical button number from which the command comes.
ST path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.key[*key_idx*].command_w.**log_key_nr** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.key[*key_idx*].request_r.**log_key_nr**
Data type	UINT
Value range	One of the values which are defined as logical jog button pair numbers in the configuration list hand_mds.lis for the characteristics tasten_data[*X*].log_tasten_nr.

Start/end of button press		
Description	Start/end of button press event and motion direction of buttons in manual mode	
ST path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.key[*key_idx*].command_w.**direction** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.key[*key_idx*].request_r.**direction**	
Data type	INT	
Value range	**Value**	**Meaning**
	-1	Start of button press, negative motion direction
	0	End of button press
	1	Start of button press, positive motion direction
Special features	**Consumption data item** Since the enforcement of a button press is a message-orientated transfer, both the “start of button press” event and the “end of button press” event must be generated by the PLC.	

Lifetime of the button signal	
Description	This is a time period defined by number of interpolator cycles. If this element has a value unequal to 0, the CNC independently generates the “End of button press” event after receiving a “Start of button press” event after the time period expires and which was defined by the number of specified interpolator cycles.
ST path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.key[*key_idx*].command_w.**life_time** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.key[*key_idx*].request_r.**life_time**
Data type	UDINT
Unit	Number of interpolator cycles
Special features	See Note [▶ 214]

Retriggering “start of button press” event	
Description	Retriggering the “start of button press” event. If the element “Life time of a button signal” [▶ 214] has a value unequal to 0, the “start of button press” event is retriggered if the “Lifetime of the button signal” has not yet expired.
ST path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.key[*key_idx*].command_w.**f_refresh** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.key[*key_idx*].request_r.**f_refresh**
Data type	BOOL
Value range	[TRUE = Retriggering the “start of button press” event, FALSE]
Special features	See Note [▶ 214]

Note: The “Lifetime of the button signal“ and “Retriggering the ‘start of button press’ event“ elements represent a sort of watchdog function. Their use is indicated in particular if the time response of the PLC is not ensured (deterministic, e.g. soft PLC as Windows task).

9.2.3.2 Rapid traverse velocity during path motion

During continuous jog mode it is possible to switch between the normal velocity and rapid traverse velocity. Rapid traverse velocity is defined in the axis-specific parameter list.

handbetrieb.tipp.vb_eilgang (P-AXIS-00210) 4000

Here the rapid traverse is a button-specific feature and only becomes effective when the corresponding button is pushed and linked to an axis.

9.2.3.2.1 Control Unit

Rapid traverse velocity during path motion in manual mode	
Description	Control unit to activate/deactivate rapid traverse mode by a normal button press in manual mode.
Data type	MC_CONTROL_HB_RAPID_KEY_UNIT, see description Control Unit with usage check [▶ 14]
Special features	Button-specific signalling of rapid traverse mode is possible at any time and is saved internally for each button. But this only has an effect when the corresponding button is pressed.
ST path	gpCh[*channel_idx*]^.hb_mc_control.**rapid_key**
Access	PLC reads **request_r** and writes **command_w** + **enable_w**
Commanded, requested data	
ST element	**.command_w** **.request_r**
Data type	MC_CONTROL_HB_RAPID_KEY_UNIT [▶ 217], for description see the section on User Data [▶ 216]
Flow control commanded data	
ST element	**.command_semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC accepts the commanded data if this element has the value TRUE and sets this element to the value FALSE after complete acceptance of the data. PLC can write data for commanding if this element has the value FALSE. The PLC sets this element to the value TRUE if all data to be commanded is written.
Flow control of requested data	
ST element	**.request_semaphor_rw**
Data type	BOOL
Special features	**Consumption data item**
Access	CNC writes the data requested by the GUI if this element is FALSE and then sets this element to TRUE. PLC reads the data requested by the GUI if this value is TRUE. After the PLC fully accepts the data, the PLC sets this element to FALSE.
Redirection	
ST element	**.enable_w**

9.2.3.2.2 User data

Button number	
Description	Logical button number for which the rapid traverse mode should be selected/deselected.
ST path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.rapid_key.command_w.**log_key_nr** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.rapid_key.request_r.**log_key_nr**
Data type	UINT
Value range	One of the values defined as logical jog button pair numbers in the configuration list hand_mds.lis for the characteristics tasten_data[*X*].log_tasten_nr

Rapid traverse velocity during path motion	
Description	Rapid traverse mode of the button on/off.
ST path	**Commanded value** gpCh[*channel_idx*]^.hb_mc_control.rapid_key.command_w.**key_pressed** **Requested value** gpCh[*channel_idx*]^.hb_mc_control.rapid_key.request_r.**key_pressed**
Data type	UINT
Value range	TRUE = Button for rapid traverse mode is active. The parameterised rapid traverse path motion is used for continuous jog mode. FALSE = Button not active in rapid traverse mode. The parameterised normal path velocity is used in continuous joy mode.]

9.2.3.3 Handwheel increments

Handwheel increments, counter state	
Description	Array of control units to manage the counts of handwheel increments for all handwheels, including flow control of user data.
Data type	ARRAY[0..HLI_HW_MAXIDX] OF MC_CONTROL_SGN32_UNIT, see description Control unit [▶ 13]
Special features	Handwheel counts can be changed at any time but they only have an effect if there is an assigned axis in handwheel mode. Relative changes to a handwheel count are only evaluated after manual mode is activated.
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST path	gpCh[*channel_idx*]^.hb_mc_control.**handwheel_incs[***idx***]**
Commanded, requested and return value	
ST element	**.command_w** **.request_r** **.state_r**
Data type	MC_CONTROL_SGN32_UNIT
Redirection	
ST path	**.enable_w**

10 Safety system

10.1 Channel-specific interface

10.1.1 Watchdog mechanism

Watchdog, CNC monitors PLC	
Description	The CNC uses the data of this unit to monitor whether the PLC is still ready for operation.

Notice

When the PLC is being monitored by the CNC, the PLC task in which the PLC life sign alive_rw is set must be executed with a higher priority than the two CNC tasks SDA and COM.
In order to prevent problems with CNC block supply in large PLC projects, it is also recommended to relocate the watchdog functionality to an extra PLC task and to only assign this task with a higher priority.

Data type	MC_WATCHDOG_UNIT
ST Path	gpCh[*channel_idx*]^.head.**mc_watchdog**
PLC operates functionality	
Description	By setting this element to the value TRUE, the PLC signals to the CNC that it supplies the data for the watchdog functionality.
ST Element	**.enable_w**
Data type	BOOL
Value range	[TRUE = PLC supplies data for a watchdog monitor, FALSE]
Access	PLC writes
PLC signal	
Description	CNC uses this element to determine whether the PLC is invoked cyclically. The CNC only checks this element if the element “PLC operates functionality” and “PLC present” [▶ 223] have the value TRUE.
ST element	**.alive_rw**
Data type	BOOL
Value range	[TRUE = PLC signals that it was invoked, FALSE]
Access	CNC detects whether this element changes from FALSE to TRUE within the time specified by the element "Watchdog cycle time". The PLC application must therefore ensure that this element is set cyclically to TRUE within the "Watchdog cycle time". The CNC sets the value to FALSE after detecting this value.
Signal “PLC is invoked cyclically”	
Description	After the first rising edge at the element "Signal from PLC", the CNC sets this signal to TRUE. The signal remains TRUE until the CNC detects that PLC is no longer invoked cyclically. If this happens, the CNC sets this signal to FALSE. In addition, the CNC outputs the error P-ERR-270053. This means the controller must be restarted.
ST element	**.alive_state_r**
Data type	BOOL
Value range	[TRUE = PLC is invoked cyclically, FALSE = PLC has not reported anything within the watchdog time **or** no rising edge was as yet detected on the "PLC signal"]
Access	CNC writes
Watchdog cycle time	
Description	Cycle time of the watchdog. The values for the watchdog cycle time must be greater than the PLC cycle time in order to allow practical monitoring of the PLC by the CNC.
ST element	**.cycle_time_w**
Data type	UDINT
Unit	1 µs
Access	PLC writes

Watchdog, PLC monitors CNC	
Description	The PLC uses the data of this unit to monitor whether the CNC is still ready for operation.

Notice

When the CNC is monitored by the PLC, the PLC task which checks for the CNC life sign alive_rw must be executed at a higher priority than the two CNC tasks SDA and COM.
In order to prevent problems with CNC block supply in large PLC projects, it is also recommended to relocate the watchdog functionality to an extra PLC task and to only assign this task with a higher priority.

Data type	LC_WATCHDOG_UNIT
ST Path	gpCh[*channel_idx*]^.head.**lc_watchdog**
CNC signal	
Description	The CNC writes TRUE in this element in every interpolation cycle to confirm that the CNC was invoked.
ST Path	**.alive_rw**
Data type	BOOL
Access	The CNC writes the value TRUE in every CNC cycle to confirm its readiness for operation. PLC detects whether this element changes from FALSE to TRUE within the time specified by the element “Watchdog cycle time” [▶ 219]. The PLC sets the value to FALSE after successful detection.
Signal “CNC is not invoked cyclically”	
Description	If the PLC detects that the CNC is no longer ready for operation, it sets this element to TRUE.
ST Path	**.alive_state_w**
Data type	BOOL
Value range	[TRUE = CNC is not invoked cyclically, FALSE = CNC is invoked cyclically]
Access	PLC writes
Watchdog cycle time	
Description	The NC kernel writes the interpolator cycle time in this element. The values for the watchdog cycle time must be greater than the PLC cycle time to permit practical monitoring of the CNC by the PLC.
ST Path	**.cycle_time_r**
Data type	UDINT
Unit	1 µs
Access	PLC reads

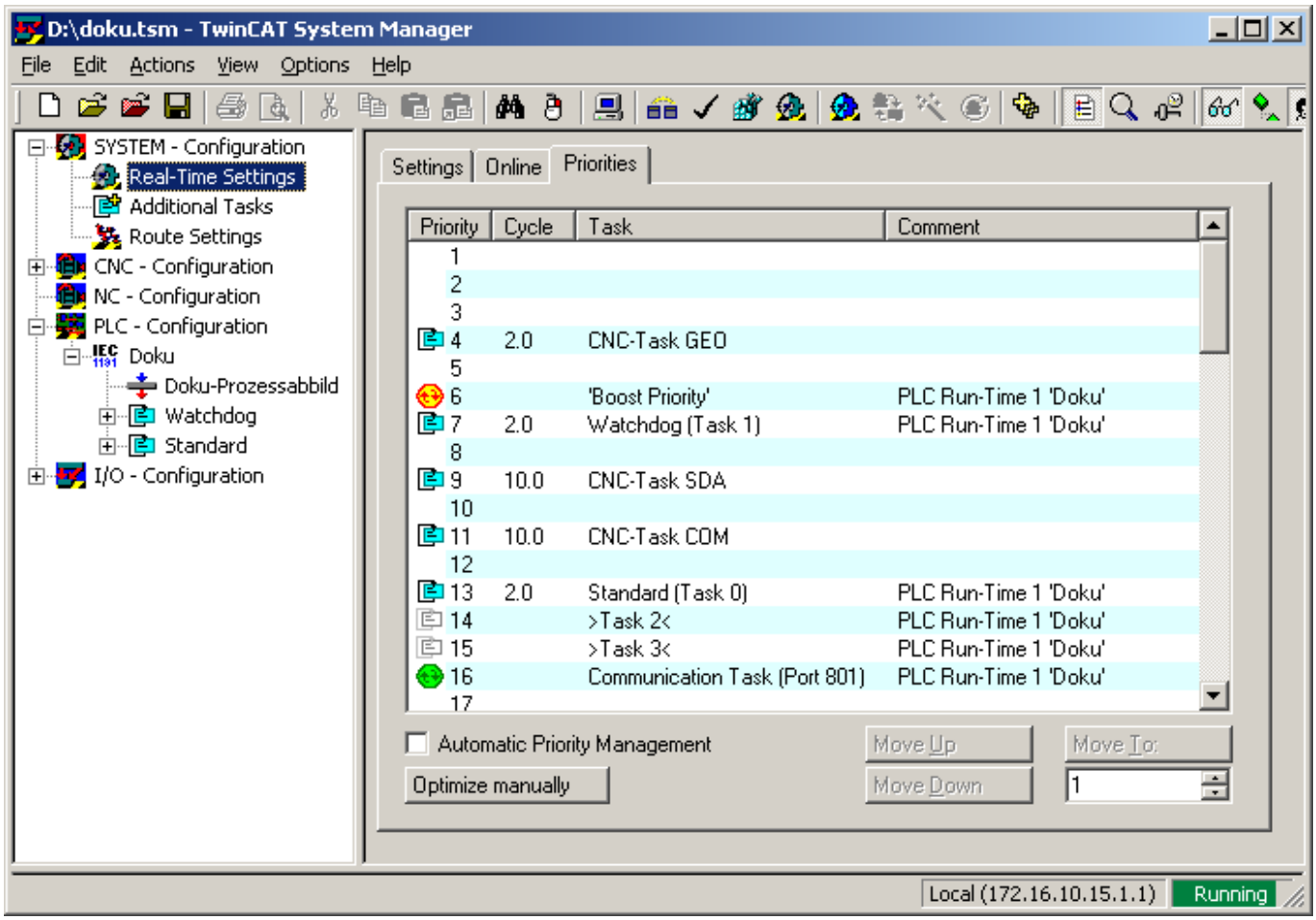


Fig. 30: Priorities for watchdog mechanism (example TwinCAT 2)

11 Management

11.1 Channel-specific interface

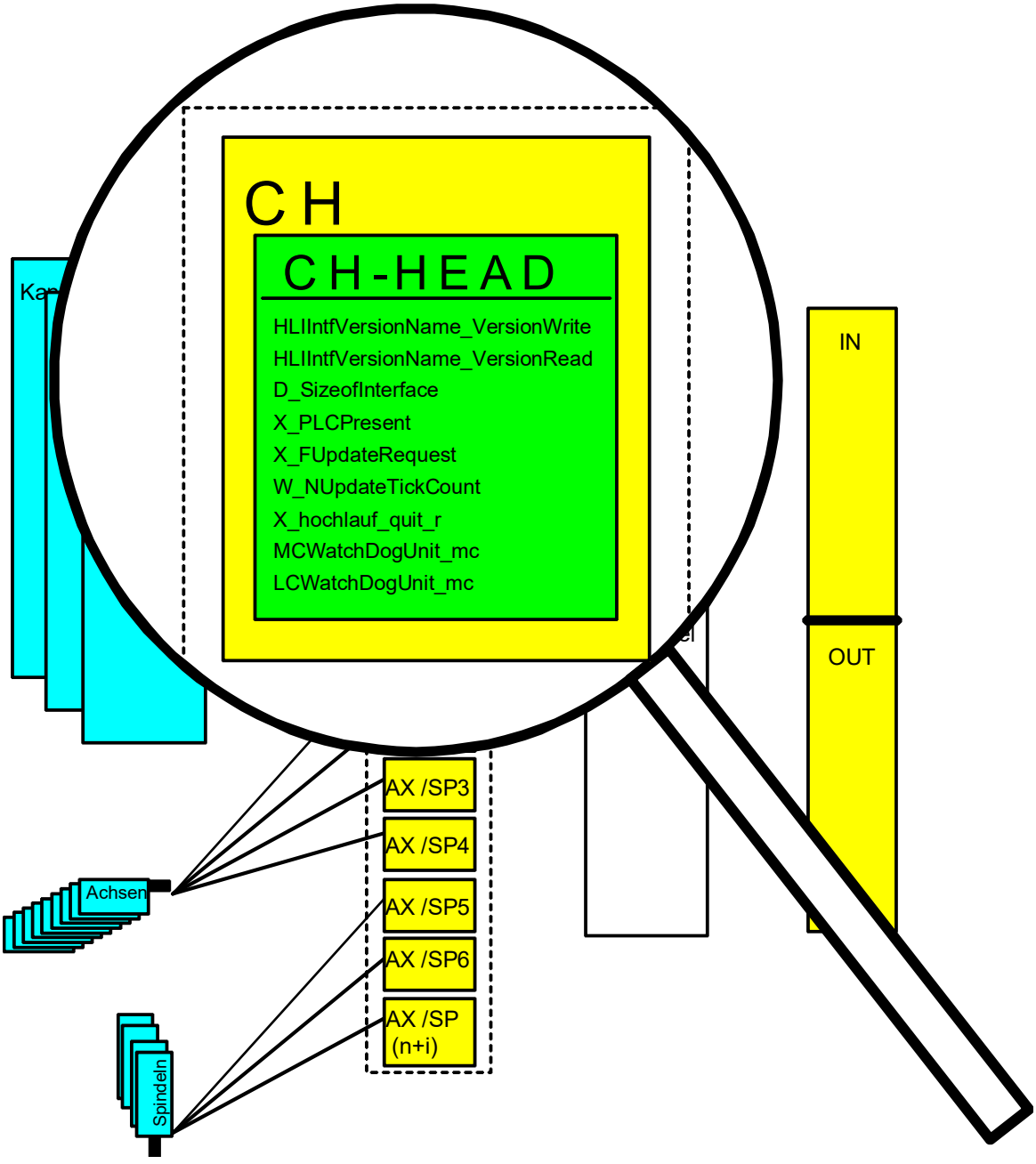


Fig. 31: Management data of the channel-specific interface

Version identifier PLC → CNC	
Description	If the version of the HLI at the NC kernel end is evaluated, the PLC stores the agreed version identifier in this element. Evaluation at the NC kernel end must be agreed for a specific application.
ST path	gpCh[*channel_idx*]^.head.version_w.**zeichen**
Data type	STRING(HLI_INTF_VERSION_NAME_LAENGE)
Access	PLC writes
Special features	Is currently not evaluated in the NC kernel.

Version identifier CNC → PLC	
Description	If the version of the HLI is evaluated at the PLC end, the NC kernel stores the agreed version identifier in this element. Evaluation at the PLC end must be agreed for a specific application.
ST path	gpCh[*channel_idx*]^.head.version_r.**zeichen**
Data type	STRING(HLI_INTF_VERSION_NAME_LAENGE)
Access	PLC reads

Size of the HLI	
Description	The NC kernel writes the size of the entire channel-specific interface area in this element.
ST path	gpCh[*channel_idx*]^.head.**sizeof_interface_r**
Data type	UDINT
Unit	Byte
Access	PLC reads

PLC present	
Description	The PLC signals that it is present by setting the value to TRUE.
ST path	gpCh[*channel_idx*]^.head.**plc_present_w**
Data type	BOOL
Value range	[TRUE = PLC present, FALSE]
Access	PLC writes

Notice

Only if this value is set to TRUE can the PLC influence the NC kernel by channel-specific control units and be supplied with technology commands by the NC kernel.

Channel number	
Description	Number of the channel that is assigned to this interface on the HLI.
ST path	gpCh[*channel_idx*]^.head.**channel_nr**
Data type	DINT
Value range	[1, gNrCh]
Access	PLC reads

NC kernel start-up completed	
Description	By setting this value to TRUE, the NC kernel signals to the PLC that the NC kernel start-up is completed and that the HLI is now supplied cyclically. This means that the display data is valid and the control commands are transferred to the NC kernel.
ST path	gpCh[*channel_idx*]^.head.**hochlauf_quit_r**
Data type	BOOL
Value range	[TRUE = Start-up of NC kernel completed, FALSE]
Access	PLC reads

Data of the channel list parameter P-CHAN-00280	
Description	The data configured by the user in P-CHAN-00280 in the channel list are available here on the HLI.
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.head.**customer_val_r[ ]**
Data type	ARRAY [0..HLI_CHANNEL_CUSTOM_MAXIDX] OF UDINT
Access	PLC reads

Enable: Updating the HLI	
Description	By setting this value to TRUE, the PLC can enable the function to update the entire channel-specific interface area.
ST path	gpCh[*channel_idx*]^.head.**update_request_enable_w**
Data type	BOOL
Value range	[TRUE = Updating the HLI enabled, FALSE]
Access	PLC writes
Special features	**Currently not supported, i.e. the HLI is updated in every NC cycle.**

Updating the HLI	
Description	By setting this value to TRUE, the PLC can initiate an update of the entire channel-specific interface area. This value is reset to FALSE by the NC kernel after the update is completed.
ST path	gpCh[*channel_idx*]^.head.**f_update_request_rw**
Data type	BOOL
Value range	[TRUE = Updating of the HLI activated, FALSE]
Access	Read/write
Special features	**Currently not supported, i.e. the HLI is updated in every NC cycle.**

Update cycle of the HLI	
Description	Using this element, the PLC informs the NC kernel of the number of interrupt cycles the NC kernel may distribute the updating of the axis-specific interface area. This leads to a reduced load on the interrupt task in particular with multi-channel and multi-axis configurations.
ST path	gpCh[*channel_idx*]^.head.**n_update_tick_count_rw**
Data type	UINT
Unit	Number of interpolator cycles
Access	PLC writes
Special features	**Currently not supported, i.e. the HLI is updated in every NC cycle.**

11.2 Axis-specific interface

PLC present	
Description	The PLC signals that it is present by setting the value to TRUE.
Signal flow	PLC → CNC
ST path	gpAx[*axis_idx*]^.head.**plc_present_w**
Data type	BOOL
Value range	[TRUE = PLC present, FALSE]
Access	Write

Notice

Only if this value is set to TRUE can the PLC influence the NC kernel by axis-specific control units and be supplied with technology commands by the NC kernel.

Logical axis number	
Description	Logical number of an axis which is unique within the entire system and identifies the axis. The datum is available at this point after controller start-up irrespective of configured axis operation mode or assignment of the axis to a channel.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.head.**log_ax_n_r**
Data type	UINT
Access	Read

Axis type	
Description	Displays the axis type which was configured for this axis in the associated axis parameter list (see P-AXIS-00018). The value is defined by interpreting the axis parameter list at controller start-up.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.head.**axis_type_r**
Data type	UINT
Access	Read

Axis error	
Description	The CNC sets this value to TRUE if the axis is in error state.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.head.**error_r**
Data type	BOOL
Value range	[TRUE = Axis is in error state, FALSE = No error]
Access	Read

Axis error, not resettable	
Description	The CNC sets this value to TRUE if an unresettable error was issued for an axis.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.head.**reset_error_locked_r**
Data type	BOOL
Value range	[TRUE = Not resettable axis error occurred, FALSE = No error or resettable error occurred]
Access	PLC reads

Data in the axis list parameter P-AXIS-00510	
Description	The data configured by the user in the parameter P-AXIS-00510 in the axis parameter list are available here on the HLI.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.head.**customer_val_r[ ]**
Data type	ARRAY [0..HLI_AXIS_CUSTOM_MAXIDX] OF UDINT
Access	PLC reads

Updating the HLI	
Description	By setting this value to TRUE, the PLC can initiate an update of the entire axis-specific interface area. This value is reset to FALSE by the NC kernel after the update is completed.
Signal flow	CNC → PLC
ST path	gpAx[*axis_idx*]^.head.**f_update_request_rw**
Data type	BOOL
Value range	[TRUE = Updating of the HLI activated, FALSE]
Access	Read/write
Special features	**Still currently supported, i.e. the HLI is updated in every NC cycle.**

Update cycle of the HLI	
Description	Using this element, the PLC informs the NC kernel of the number of interrupt cycles the NC kernel may distribute the updating of the axis-specific interface area. This leads to a reduced load on the interrupt task in particular with multi-channel and multi-axis configurations.
Signal flow	PLC → CNC
ST path	gpAx[*axis_idx*]^.head.**n_update_tick_count_rw**
Data type	UINT
Unit	Number of interpolator cycles
Access	Write
Special features	**Currently not supported, i.e. the HLI is updated in every NC cycle.**

12 Error messages

Errors occurring in the NC kernel can be displayed on the HLI interface in order to provide the PLC with the option of responding to the error and logging it.

The significance of the individual structure elements is documented in [DIAG].

Each error message is identified by a unique error number and the error messages are documented in [DIAG].

12.1 Management data of an error message

12.1.1 Error message CNC -> PLC

Error message CNC → PLC management data	
Description	This structure contains user data and data for flow control of an error message.
Data type	MC_ERROR
ST path	gpCh[*channel_idx*]^.**mc_error**
User data	
Description	User data of an error message.
ST element	**.satz_r**
Data type	HLI_ERROR_SATZ
Access	PLC reads
Flow control of user data	
Description	Validity flag of an error message.
ST element	**.semaphor_rw**
Data type	BOOL
Value range	[TRUE = User data is valid, FALSE]
Access	The CNC sets this value to TRUE if the user data of an error message is written and the error message is then valid. PLC must set the value of this element to FALSE if the “Redirection” [▶ 199] element has the value TRUE and the PLC accepted the error message.
Redirection	
ST element	**.enable_w**
Data type	BOOL

12.1.2 Error message PLC -> CNC

Error message PLC → CNC management data	
Description	This structure contains user data and data for flow control of an error message.
Data type	LC_ERROR
ST path	gpCh[*channel_idx*]^.**lc_error**
User data	
Description	User data of an error message.
ST element	**.satz_r**
Data type	HLI_ERROR_SATZ
Access	PLC writes
Flow control of user data	
Description	Validity flag of an error message.
ST element	**.semaphor_rw**
Data type	BOOL
Value range	[TRUE = User data is valid, FALSE]
Access	PLC sets this value to TRUE if the user data of an error message is written and therefore the error message is valid. CNC sets the value of this element to FALSE when the CNC has accepted the error message.

12.2 User data of an error message

Error number	
Description	Unique error number.
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**error_id**
Data type	UDINT
Access	PLC reads
Special features	Used for internal diagnostic purposes.

Module name of the module signalling an error	
Description	Module name of the module signalling an error
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**modul_name.zeichen**
Data type	STRING(HLI_MODUL_NAME_LAENGE)
Access	PLC reads
Special features	Used for internal diagnostic purposes.

Line number	
Description	Line number in the module on which the error occurred.
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**line**
Data type	INT
Access	PLC reads
Special features	Used for internal diagnostic purposes.

Error number of a utility function	
Description	Error number when a utility function is used.
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**util_error_id**
Data type	UDINT
Access	PLC reads
Special features	Used for internal diagnostic purposes.

Module name of the module with utility functions signalling an error	
Description	Module name of the module with utility functions signalling an error
Signal flow	CNC → PLC
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**util_modul_name.zeichen**
Data type	STRING(HLI_MODUL_NAME_LAENGE)
Access	PLC reads
Special features	Used for internal diagnostic purposes.

Line number of a utility function	
Description	Line number of the line in which the error occurred in a module with utility function.
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**util_line**
Data type	INT
Special features	Used for internal diagnostic purposes.

Multiple error number	
Description	Error messages may be issued at several different points in the NC kernel. A unique multiple error number is issued to distinguish multiple usage.
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**multiple_id**
Data type	UINT
Special features	Used for internal diagnostic purposes.

Type of commandable function	
Description	Type of commandable function in which an error occurred.
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**bf_type**
Data type	UINT
Special features	Used for internal diagnostic purposes.

Channel number	
Description	Channel number of the channel in which the signalled error occurred.
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**cnc_channel**
Data type	UINT

Communication ID	
Description	Communication ID of the BF signalling an error in the CNC
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**kommu_id**
Data type	UINT
Access	PLC reads
Special features	Used for internal diagnostic purposes.

Time specification on output of an error message:	
Description	Date of the instant of the error message
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**fb_zeitangabe.date_counter**
Data type	UDINT
Special features	Currently not implemented, value is always 0.

Time specification on output of an error message: Interrupt cycles since system start	
Description	Number of interrupt cycles since system start at the instant of an error message
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**fb_zeitangabe.zykl_counter**
Data type	UDINT
Access	PLC reads

CNC version name	
Description	Version name of the CNC specified in the error message.
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**version_name.zeichen**
Data type	STRING(HLI_VERSION_NAME_LAENGE)

Recovery class	
Description	Recovery class of an error
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**behebungs_klasse**
Data type	UINT
Value range	[0, 8]

Reaction class of an error	
Description	Reaction class of an error
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**reaktions_klasse**
Data type	UINT
Value range	[0, 8]

Body type of an error		
Description	Body type of an error. Depending on the error type, the error set body contains further information on the error which occurred. The individual structure elements are described in [DIAG].	
ST path	gpCh[*channel_idx*]^.mc_error.satz_r.kopf.**rumpf_typ**	
Value range	**Value**	**Constant**
	1	HLI_RUMPF_TYP_NC_PROG
	2	HLI_RUMPF_TYP_MDS
	3	HLI_RUMPF_TYP_KOMMU
	4	HLI_RUMPF_TYP_RAMDISK
	5	HLI_RUMPF_TYP_FILE
	6	HLI_RUMPF_TYP_INTPR_FILE
	7	HLI_RUMPF_TYP_LISTE_BINAER
	8	HLI_RUMPF_TYP_GCM
	9	HLI_RUMPF_TYP_LEER
	10	HLI_RUMPF_TYP_HLI
	11	HLI_RUMPF_TYP_NC_PROG_LR

12.2.1 Error message content, body nc program

NC program	
Description	Error information in relation to NC program
Data type	HLI_RUMPF_NC_PROG
ST path	pNcProgErr : POINTER TO HLI_RUMPF_NC_PROG; **pNcProgErr** := ADR(gpCh[*channel_idx*]^.mc_error.satz_r.rumpf.maske.err_mask[0]);
Description	Logical path number (see start-up list).
ST path	**.log_pfad_nr**
Data type	UINT
Value range	s. Start-up list
Description	Program name
ST path	**.prog_name**
Data type	HLI_PROG_NAME
Description	File name
ST path	**.file_name**
Data type	HLI_PROG_NAME
Description	File offset in bytes
ST path	**.fileoffset**
Data type	UDINT
Description	Position in NC block in bytes.
ST path	**.satzoffset**
Data type	UINT
Description	Token offset in current NC line
ST path	**.tokenoffset**
Data type	UINT
Description	Block number of current NC line
ST path	**.satz_nr**
Data type	UDINT

Description	Additional program information
ST path	**.add_prog_info**
Data type	HLI_S_ADD_PROG_INFO
Special feature	**Not implemented in all versions.**

12.2.2 Error message content, body machine data

Machine data		
Description	Error information in relation to an update of machine data	
Data type	HLI_RUMPF_MDS	
ST path	pMachineDataErr : POINTER TO HLI_RUMPF_MDS; **pMachineDataErr** := ADR(gpCh[*channel_idx*]^.mc_error.satz_r.rumpf.maske.err_mask[0]);	
Description	Significance of list	
ST path	**.listen_typ**	
Data type	UINT	
Value range	**Value**	**Meaning**
	1	Manual mode list
	2	Channel parameter list
	3	Axis parameter list
	4	Zero offset list
	5	Tool parameter list
	6	Position offset list
	7	Start-up list
	9	Axis compensation list
	12	List of external variables
Description	Name of wrong structure	
ST path	**.strukt_name.zeichen**	
Data type	STRING(HLI_STRUKT_NAME_LAENGE)	

12.2.3 Error message content, body communication

Communication	
Description	Error information in relation to a protocol data unit (message)
Data type	HLI_RUMPF_KOMMU
ST path	pCommuErr : POINTER TO HLI_RUMPF_KOMMU; **pCommuErr** := ADR(gpCh[*channel_idx*]^.mc_error.satz_r.rumpf.maske.err_mask[0]);
Description	Type of list
ST path	**.medium**
Data type	UINT
Value range	1 – function block, 2 – PDU
Description	Code of message
ST path	**.typ**
Data type	UDINT
Description	Receiver or sender of message
ST path	**.partner**
Data type	UDINT

12.2.4 Error message content, body RAM disk

RAM disk	
Description	Error information in relation to RAM disk access
Data type	HLI_RUMPF_RAMDISK
ST path	pRamDiskErr : POINTER TO HLI_RUMPF_RAMDISK; **pRamDiskErr** := ADR(gpCh[*channel_idx*]^.mc_error.satz_r.rumpf.maske.err_mask[0]);
Description	Listentyp
ST path	**.medium**
Data type	UINT
Value range	1 – function block, 2 – PDU
Description	Code of message
ST path	**.typ**
Data type	UDINT
Value range	
Description	Receiver or sender of message
ST path	**.partner**
Data type	UDINT
Value range	
Description	File name
ST path	**.file_name.zeichen**
Data type	STRING(HLI_NAME_SIZE)
Description	File offset in bytes
ST path	**.fileoffset**
Data type	UDINT

12.2.5 Error message content, body file

File	
Description	Error information in relation to file access
Data type	HLI_RUMPF_FILE
ST path	pFileErr : POINTER TO HLI_RUMPF_FILE; **pFileErr** := ADR(gpCh[*channel_idx*]^.mc_error.satz_r.rumpf.maske.err_mask[0]);
Description	File name
ST path	**.file_name.zeichen**
Data type	STRING(HLI_NAME_SIZE)
Description	File offset in bytes
ST path	**.fileoffset**
Data type	UDINT

12.2.6 Error message content, body interpreted file list

List interpretation		
Description	Error information in relation to interpretation of ASCII parameter list	
Data type	HLI_RUMPF_INTPR_FILE	
ST path	pIntprFileErr: POINTER TO HLI_RUMPF_INTPR_FILE; **pIntprFileErr** := ADR(gpCh[*channel_idx*]^.mc_error.satz_r.rumpf.maske.err_mask[0]);	
Description	File name	
Data type	**.file_name.zeichen**	
ST path	STRING(HLI_NAME_SIZE)	
Description	File offset in bytes	
Data type	**.fileoffset**	
ST path	UDINT	
Description	Line number in current file	
Data type	**.file_line_number**	
ST path	UDINT	
Description	Listentyp	
Data type	**.listen_typ**	
ST path	UINT	
Value range	**Value**	**Meaning**
	1	Manual mode list
	2	Channel parameter list
	3	Axis parameter list
	4	Zero offset list
	5	Tool parameter list
	6	Position offset list
	7	Start-up list
	9	Axis compensation list
	12	List of external variables

Description	Name of wrong structure
Data type	**.strukt_name.zeichen**
ST path	STRING(HLI_STRUKT_NAME_LAENGE)

12.2.7 Error message content, body binary list

Binary list	
Description	Error information in relation to binary list update
Data type	HLI_RUMPF_LISTE_BINAER
ST path	pBinaryListErr : POINTER TO HLI_RUMPF_LISTE_BINAER; **pBinaryListErr** := ADR(gpCh[*channel_idx*]^.mc_error.satz_r.rumpf.maske.err_mask[0]);
Description	Name of wrong structure
ST path	**.strukt_name.zeichen**
Data type	STRING(HLI_STRUKT_NAME_LAENGE)

12.2.8 Error message content, body global channel manager

Global channel manager	
Description	Error information in relation to global channel manager
Data type	HLI_RUMPF_GCM
ST path	pGcmErr : POINTER TO HLI_RUMPF_GCM; **pGcmErr** := ADR(gpCh[*channel_idx*]^.mc_error.satz_r.rumpf.maske.err_mask[0]);
Description	Wrong keyword or parameter in the interpreted character string
ST path	**.token.zeichen**
Data type	STRING(HLI_VAR_STRING_LAENGE)
Description	Name of GCM command file
ST path	**.file_name.zeichen**
Data type	STRING(HLI_NAME_SIZE)
Description	Interpreter number
ST path	**.interp_no**
Data type	UDINT
Description	Line number
ST path	**.line_no**
Data type	UDINT
Description	Column number
ST path	**.col_no**
Data type	UDINT
Description	Offset from file start
ST path	**.offset**
Data type	UDINT
Description	Number of command
ST path	**.command_no**
Data type	UDINT

Description	Workpiece ID
ST path	**.part_id**
Data type	UDINT
Description	Step ID
ST path	**.step_id**
Data type	UDINT
Description	Level of IF brackets
ST path	**.if_level**
Data type	UDINT

12.2.9 Additional error information value 1 - 5

Additional error information	
Description	Individual error information
ST path	HLI_WERT_B
Data type	gpCh[*channel_idx*]^.mc_error.satz_r.rumpf.**wert_1 ... wert_5**
Description	Data type
ST path	**.typ**
Data type	INT
Value range	See Value range of variable ‘Type’ of additional error information [▶ 244]
Description	Dimension of datum
ST path	**.dimension**
Data type	INT
Value range	See Value range of ‘Dimension’ variable of the additional error information [▶ 245]
Description	Significance of datum
ST path	**.bedeutung**
Data type	INT
Value range	See Value range of ‘Significance’ variable of additional error information [▶ 246].
Description	Datum itself
ST path	**.inhalt.data**
Data type	ARRAY [0..HLI_WERT_B_DATA_MAXIDX] OF BYTE (as union, must be output type-caste corresponding to data type)

12.2.9.1 Value range of variable ‘Type’ of additional error information

Constant	Value	Description
HLI_TYP_BOOLEAN	1	Boolean value
HLI_TYP_UNS08	2	Unsigned 1 byte value
HLI_TYP_SGN08	3	Signed 1 byte value
HLI_TYP_UNS16	4	Unsigned 2 byte value
HLI_TYP_SGN16	5	Signed 2 byte value
HLI_TYP_UNS32	6	Unsigned 4 byte value
HLI_TYP_SGN32	7	Signed 4 byte value
HLI_TYP_UNS64	8	Unsigned 8 byte value
HLI_TYP_SGN64	9	Signed 8 byte value
HLI_TYP_REAL64	10	8 byte decimal value*
HLI_TYP_POINTER	12	Address
HLI_TYP_CHAR	18	Character, 1 byte
HLI_TYP_STRING	19	Character string
HLI_TYP_ADRESSE	20	Address
HLI_TYP_A3_REAL64	23	Array containing three 8-byte decimal values
HLI_TYP_HIGH_RES_SGN_POS	24	Signed value, size depends on system and may be 4 or 8 bytes.
HLI_TYP_BITARRAY_32	25	Array of 4-byte values
HLI_TYP_BITARRAY_16	26	Array of 2-byte values

Back to description of ‘Type’ variable. [▶ 243]

12.2.9.2 Value range of ‘Dimension’ variable of the additional error information

Constant	Value	Unit	Description
HLI_NO_DIMENSION	-1	-	This is not a dimension specification
HLI_DIM_DIMENSIONSLOS	0	-	Without dimension
HLI_DIM_POSITION	1	10^{-4} mm or 10^{-4} °	Position
HLI_DIM_GESCHWINDIGKEIT	2	10^{-3} mm/s or 10^{-3} °	Velocity
HLI_DIM_BESCHLEUNIGUNG	3	1 mm/s^2 or 1°/s^2	acceleration
HLI_DIM_RUCK	4	1 mm/s^3 or 1°/s^3	jerk
HLI_DIM_ZEIT	5	1 µs	time
HLI_DIM_PROZENT	6	0,1 %	Prozent
HLI_DIM_ADRESSE	7	-	Address
HLI_DIM_INKREMENTE	8	Increments	Path increment
HLI_DIM_UMDR_VORSCHUB	9	10^{-4} mm/rev	Feedrate per revolution
HLI_DIM_V_SCHNITT	10	10^{-3} mm/s	Cutting velocity
HLI_DIM_WEG_AUFLOESUNG	11	Increments/10^{-3} mm	Path resolution
HLI_DIM_INKR_UMDREHUNG	12	Increments/rev.	Increments per revolution
HLI_DIM_BYTE	13	-	Number of bytes
HLI_DIM_PROPORTIONAL_GAIN	14	0.01/s	Proportional gain
HLI_DIM_FREQUENCY	15	Hz	Frequency
HLI_DIM_LOAD	16	1 kg or 1 kg*m^2	Motor load
HLI_DIM_POSITION_HIGH_RES	17	10^{-8} mm or 10^{-8} °	Position, high resolution
HLI_DIM_INKREMENTE_HIGH_RES	18	10^{-4} increments	Increment, high resolution

Back to description of variable dimension. [▶ 243]

12.2.9.3 Value range of ‘Significance’ variable of additional error information

Constant	Value	Description
HLI_BEDEUT_IGNORE	0	Ignore value may not be changed: Initialise with 0
HLI_BEDEUT_GRENZ_WERT	1	Limit value
HLI_BEDEUT_AKT_WERT	2	Current value
HLI_BEDEUT_FEHL_WERT	3	Error value
HLI_BEDEUT_ERWARTET_WERT	4	Expected value
HLI_BEDEUT_KORR_WERT	5	Corrected value
HLI_BEDEUT_LOG_ACHS_NR	6	logical axis number
HLI_BEDEUT_ANTR_TYP	7	Drive type
HLI_BEDEUT_LOG_BED_ELEM_NR	8	logical identifier of control element
HLI_BEDEUT_ZUSTAND	9	State
HLI_BEDEUT_TRANSITION	10	Transition
HLI_BEDEUT_SENDER	11	Sender
HLI_BEDEUT_KLASSE	12	Class
HLI_BEDEUT_INSTANZ	13	Instance
HLI_BEDEUT_IDENT_NR	14	Identification number
HLI_BEDEUT_STATUS	15	Status
HLI_BEDEUT_RING_NR	16	Ring number
HLI_BEDEUT_SATZ_NR	17	Block number
HLI_BEDEUT_MIN_LIMIT	18	Lower limit value
HLI_BEDEUT_MAX_LIMIT	19	Upper limit value
HLI_BEDEUT_START_WERT	20	Initial value
HLI_BEDEUT_ZIEL_WERT	21	Final value
HLI_BEDEUT_FILENAME	22	File name
HLI_BEDEUT_LINE_NUMBER	24	Line number in a file
HLI_BEDEUT_COLUMN_NUMBER	25	Column number in a file
HLI_BEDEUT_ARGUMENT	26	Identifier of an argument
HLI_BEDEUT_PARAMETER	27	Identifier of a parameter
HLI_BEDEUT_AXIS	28	Identifier of an axis
HLI_BEDEUT_COMPENSATION	29	Compensation index
HLI_BEDEUT_IDENTIFIER	30	Identifier
HLI_BEDEUT_CHAIN	31	Kinematic chain

Back to description of variable significance. [▶ 243]

12.3 Activating error filters

Release Note

The function is available as of CNC Build V3.00.xx.

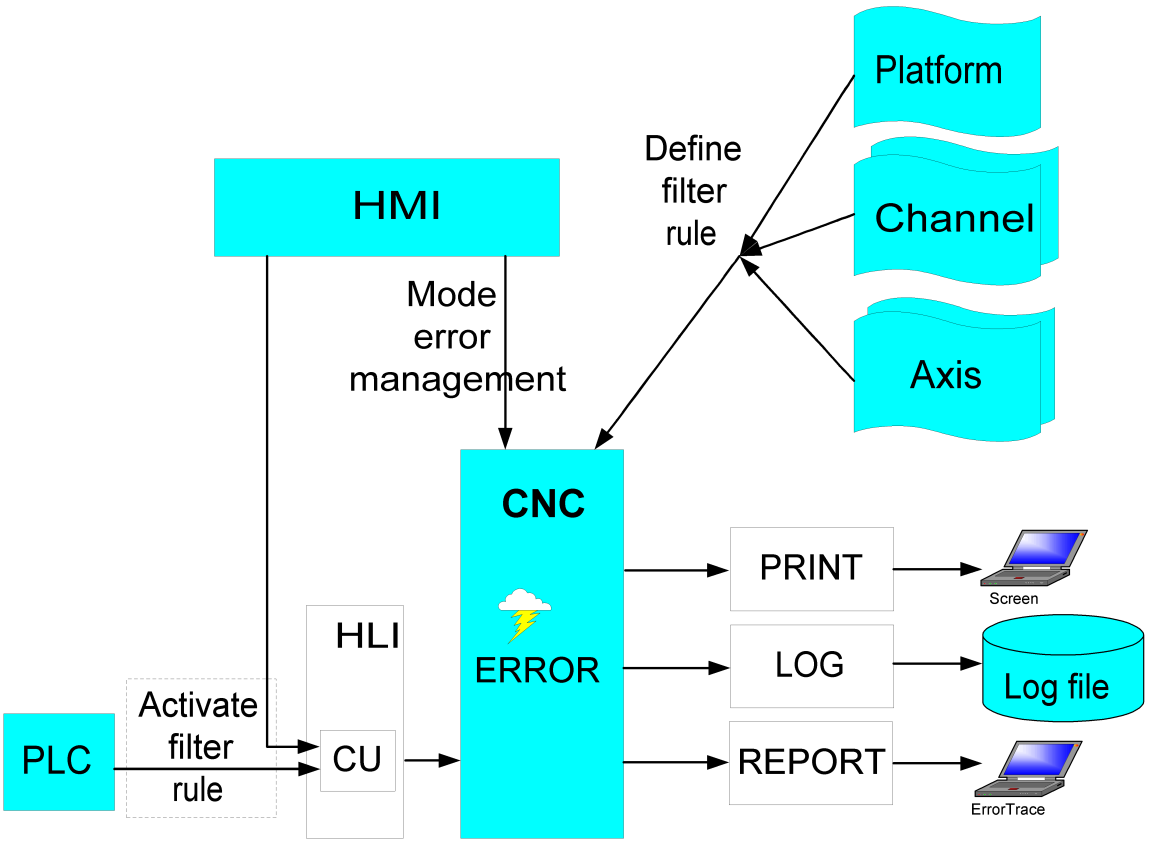


Fig. 32: Activating the filter

Activating error filter rules - Platform	
Description	Here, the individual error filters can be activated/deactivated by the 32-bit mask according to their activation bits. For example the following rule is activated by setting the first bit (command_w = 0x00000001): `error_filter[0].activation_bit 0x1`
Data type	MC_CONTROL_UNS32_UNIT, see description of control unit
Unit	[-]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpPform^.error_filter
Commanded and requested value	
ST element	**.command_w** **.request_r**
Data type	UDINT
Return value	
ST element	**.state_r**
Data type	UDINT
Redirection	
ST element	**.enable_w**

Activating error filter rules - Channel	
Description	Here, the individual error filters can be activated/deactivated by the 32-bit mask according to their activation bits. For example the following rule is activated by setting the first bit (command_w = 0x00000001): `error_filter[0].activation_bit 0x1`
Data type	MC_CONTROL_UNS32_UNIT, see description of control unit
Unit	[-]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpCh*[channel_idx]*^.head.error_filter
Commanded and requested value	
ST Element	**.command_w** **.request_r**
Data type	UDINT
Return value	
ST Element	**.state_r**
Data type	UDINT
Redirection	
ST Element	**.enable_w**

Activating error filter rules - Axis	
Description	Here, the individual error filters can be activated/deactivated by the 32-bit mask according to their activation bits. For example the following rule is activated by setting the first bit (command_w = 0x00000001): `error_filter[0].activation_bit 0x1`
Data type	MC_CONTROL_UNS32_UNIT, see description of control unit
Unit	[-]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpAx*[axis_idx]*^.head.error_filter
Commanded and requested value	
ST Element	**.command_w** **.request_r**
Data type	UDINT
Return value	
ST Element	**.state_r**
Data type	UDINT
Redirection	
ST Element	**.enable_w**

13 Platform data

Platform data is data which cannot be assigned to a specific axis or a channel but has an effect on the entire NC control.

13.1 Status information of a platform

PLC present	
Description	The PLC signals that it is present by setting the value to TRUE.
ST path	gpPform^.**plc_present_w**
Data type	BOOL
Value range	[TRUE = PLC present, FALSE]
Access	PLC is writing

Notice

Only if this value is set to TRUE can the PLC influence the NC kernel by platform-specific control units.

Version identifier PLC → CNC	
Description	PLC writes the version identifier string of its HLI definition in this element.
ST path	gpPform^.nc_config.**hli_version_w.zeichen**
Data type	STRING(HLI_INTF_VERSION_NAME_LAENGE)
Access	PLC is writing

Version identifier CNC → PLC	
Description	CNC writes the version identifier string of its HLI definition in this element.
ST path	gpPform^.nc_config.**hli_version_r.zeichen**
Data type	STRING(HLI_INTF_VERSION_NAME_LAENGE)
Access	PLC is reading

Version identifier CNC → PLC	
Description	CNC writes the NC kernel version in this element.
ST path	gpPform^.nc_config.**nc_kernel_version_r.zeichen**
Data type	STRING(HLI_VERSION_NAME_LAENGE)
Access	PLC is reading

Data of the start-up parameter P-STUP-00120	
Description	Data configured by the user in the start-up list with the parameter P-STUP-00120 are available on the HLI.
Signal flow	CNC -> PLC
ST path	gpPform^.nc_config.**customer_val_r[ ]**
Data type	ARRAY [0..HLI_PLATFORM_CUSTOM_MAXIDX] OF UDINT
Access	PLC is reading

13.2 Diagnosis upload

Diagnosis upload	
Description	While the CNC is running, the PLC can command an upload of diagnosis data using this control unit. The control unit is enabled by enable_w = TRUE.
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit [▶ 13]
Access	PLC reads request_r + state_r and writes command_w + enable_w
ST Path	gpPform^.**diagnosis_upload**
Commanded, requested and return values	
ST Element	**.command_w** **.request_r** **.state_r**
Data type	BOOL
Value range	[TRUE = diagnosis upload activated, FALSE = diagnosis upload off]
Redirection	
ST Element	**.enable_w**
Special feature	**Note:** The data item command_w must remain at TRUE until state_r reverts to FALSE. Otherwise, the data is not complete since the diagnosis data upload is aborted.

Diagnosis upload state		
Description	State value for a diagnosis upload. If the diagnosis file cannot be opened, state 2 is retained and is reset in the next successful diagnosis upload. For more information on diagnosis upload, see Diagnose-Upload [FCT-M9// Diagnosis upload]	
ST path	gpPform^.**diagnosis_upload_state**	
Data type	UDINT	
Value range	**Value**	**Constant**
	0	HLI_DIAG_STATE_IDLE: Basic state, diagnosis upload not active
	1	HLI_DIAG_STATE_ACTIVE: Diagnosis upload active
	2	HLI_DIAG_STATE_FILE_ERROR: File cannot be opened
	3	HLI_DIAG_STATE_WAIT_END_ACK: Wait for acknowledgement of the diagnosis upload from the PLC
Access	PLC is reading	
Note	Available as of CNC Builds V2.11.2844, V3.1.3081.4 or V3.1.3110.	

14 Messages

14.1 Control Unit

Message to/from PLC	
Description	This control unit receives messages sent by the CNC to the PLC and messages sent by the PLC to other users of the communication process provided by the CNC. The CNC can send a message to the PLC via the commands **#MSG PLC** ["…"] or **#MSG SYN PLC** ["..."] (see [PROG]). The information on which of the commands was used to send a message is also sent in the user data of this control unit.
Data type	MC_CONTROL_MSG_UNIT, see description Control Unit with usage check [▶ 14]
ST path	gpCh[*channel_idx*]^.**msg_mc_control**
Commanded, requested data	
ST element	**.command_w** **.request_r**
Data type	HLI_MSG_SENDUNG [▶ 255]
Access	PLC writes command_w and reads request_r
Flow control of commanded value	
ST element	**.command_semaphor_rw**
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	PLC can write data for commanding if command_semaphor_rw has the value FALSE. When all command data is written, the PLC sets command_semaphor_rw to TRUE. CNC takes the commanded data if command_semaphor_rw is TRUE and then sets the flag to FALSE.
Flow control of requested data	
ST element	**.request_semaphor_rw**
Data type	BOOL
Value range	[TRUE, FALSE]
Special features	**Consumption data item**
Access	The data requested by the GUI can be written if request_semaphor_rw is FALSE. This element is then set to TRUE. PLC reads the data requested by GUI if request_semaphor_rw is TRUE. After the PLC fully accepts the data, the PLC sets this element to FALSE.
Redirection	
ST path	**.enable_w**

14.2 User data

14.2.1 Requested and commanded user data

Content of message to/from PLC	
Description	Data of message which was received or sent.
ST path	**Commanded value** gpCh[*channel_idx*]^.msg_mc_control.command_w **Requested value** gpCh[*channel_idx*]^.msg_mc_control.request_r
ST element	**.contents**
Data type	STRING(HLI_MSG_CONTENT_STRL)

Type of message to/from PLC		
Description	Marks whether a message is sent by the CNC to the PLC by a #MSG- or #MSGINFO command (see [PROG]).	
ST path	**Commanded value** gpCh[*channel_idx*]^.msg_mc_control.command_w **Requested value** gpCh[*channel_idx*]^.msg_mc_control.request_r	
ST element	**.typ**	
Data type	UDINT	
Value range	**Value**	**Meaning**
	0	Initial value
	1	Type of message which was sent by a **#MSG** command (if it was sent by the CNC)
	2	Type of message which was sent by a **#MSGINFO** command (if it was sent by the CNC)

15 Implementation as PLC library

To access the HLI, a PLC library is provided for use by the PLC application developer. This library is implemented specifically for various PLC systems. The table below contains the names of the libraries for the supported PLC systems.

PLC system	KW	3S	TwinCAT 2	TwinCAT 3
HLI PLC library	hli.zwt	hli.lib	TcCncHli.lib	Tc2_CncHli

The following is implemented in this library...

- data structures which are components of the HLI.
- global variables that are used to access the elements of the HLI
- function blocks which are used to initialise global pointers

15.1 Access to HLI

15.1.1 PLC system 3S and TwinCAT

In these two systems, the global pointers to specific areas of the HLI are created in the HLI-PLC library. These pointers are used to access control units and status data or other data of such an HLI area. The table below shows the uses of these pointers and in each of the individual PLC systems:

Global pointers to HLI areas

	Explanation	PLC systems	
Global pointers		**3S**	**TwinCAT**
gpPform	Platform-specific area	X	X
gpCh	Array of pointers to channel-specific areas	X	X
gpAx	Array of pointers to axis-specific areas	X	X
gpCTMan	Array of pointers to job planning areas	-	X
gpIoStation	Array of pointers to areas for SERCOS I/O stations	X	-
gpVEGlobal	Pointers to global V.E variables	X	X
gpVECh	Array of pointers to channel-specific V.E variables	X	X
gpHmiPlc	Pointer to area exchange data between HMI and PLC	X	-

Make sure only to initialise the pointers which are supported by the PLC system and which make sense depending on the configuration of the NC kernel (number of channels, number of axes, etc.). All other pointers are ZERO pointers.

15.1.2 PLC system by KW-Software

The PLC System from KW-Software contains the global variable **hli** as %M3.xxxvariable in the PLC application for access to the HLI. This variable is the starting point to reach all areas of the HLI.

15.2 Function blocks within the PLC library

15.2.1 Overview of implemented function blocks

The table below provides an overview of the availability of the function blocks implemented in the PLC systems supported by ISG.

Function block	PLC systems		
	KW	3S	TwinCAT
MCV_HliInterface	-	X	X

15.2.2 MCV_HliInterface

An instance of the MCV_HliInterface must be **definitely** invoked by every PLC application for whose system this FB was implemented and which wants to access the HLI elements.

The NC kernel requests information about the HLI configuration (number of channels, number of axes, etc.) on a rising edge at the “Start” input and compares the HLI configuration with the PLC configuration. If differences occur, the “Error” output is set to TRUE and an error message identifier is output at the “ErrorID” output. On the other hand, if the attributes of the HLI are identical on both sides, the “Initialised” output is set to TRUE and the PLC application can access the relevant areas of the HLI via the global defined pointers (see: PLC system 3S and TwinCAT [▶ 256]) (see PLC main program frame [▶ 261]).

As of CNC Builds V3.01.3000

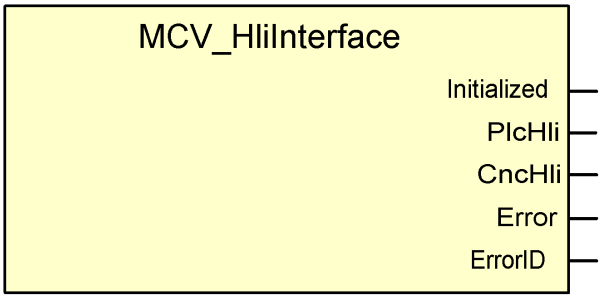


Fig. 33: FB MCV_HliInterface parameters

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Initialised	BOOL	Check was terminated successfully and only now **may** there be access to the HLI.
PlcHli	HLI_SHAPE	Information regarding the HLI configuration on the PLC side. The data structure contains the number of main areas of the HLI and the version identifier of the HLI definitions.
CncHli	HLI_SHAPE	Contains the HLI configuration on the NC kernel side.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier. For possible values see Behaviour of the FB: Table of error identifiers [▶ 260].

(* used in the PLC code:*)

```
Hli();
```

for CNC Builds V2.11.28xx

Block diagram

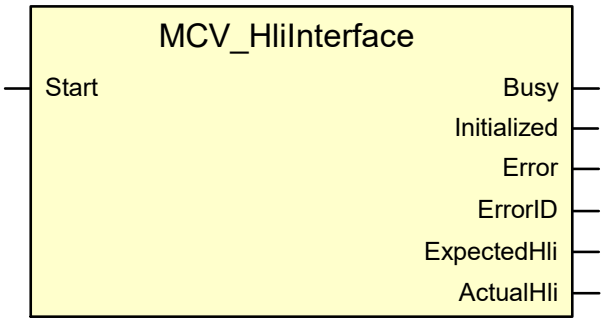


Fig. 34: FB MCV_HliInterface parameters

VAR_INPUT		
Variable name	**Data type**	**Description**
Start	BOOL	Rising edge triggers a check whether the HLI definition on the PLC side is the same as on the NC kernel side.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Busy	BOOL	Request and checking process are still active.
Initialised	BOOL	Check was terminated successfully and only now **may** there be access to the HLI.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	UDINT	Error identifier. For possible values see Behaviour of the FB: Table of error identifiers [▶ 260].
ExpectedHli	HLI_DIAGNOSTIC	Information regarding the HLI configuration on the PLC side. The data structure contains the number of main areas of the HLI and the version identifier of the HLI definitions.
ActualHli	HLI_DIAGNOSTIC	Contains the HLI configuration on the NC kernel side.

(* used in the PLC code:*)

```
Hli(Start := TRUE);
```

15.2.3 Behaviour of the FB: Table of error identifiers

Error value from MCV_HliInterface

Error value	Global constant defined in PLC library	Description
0	ERR_PLC_NO_ERROR	No error occurred
40014	ERR_PLC_PFORM_PTR_ZERO	Pointer to platform-specific area is ZERO and therefore invalid.
40015	ERR_PLC_WRONG_VERSION	Version identifier of HLI from the PLC and NC kernel are different.
40016	ERR_PLC_WRONG_PARAMETER	Different characteristic (number of possible channels, number of possible axes, etc.) of HLI in PLC and NC kernel. .
40017	ERR_PLC_WRONG_AXIFC_SIZE	Size (bytes) of axis-specific HLI area differs between PLC and NC kernel.
40018	ERR_PLC_WRONG_CHIFC_SIZE	Size (bytes) of channel-specific HLI area differs between PLC and NC kernel.
40019	ERR_PLC_HLIINFO_PTR_ZERO	The PLC received no information from the NC kernel on its HLI characteristic.
40020	ERR_PLC_AXIS_NR_UNEQUAL	Number of initialised pointers to axis-specific HLI areas differ from number of existing axes in the NC kernel.
40021	ERR_PLC_CHANNEL_NR_UNEQUAL	Number of initialised pointers to channel-specific HLI areas differ from the number of existing channels.
40022	ERR_PLC_WRONG_PFIFC_SIZE	Size (bytes) of platform-specific HLI area differs between PLC and NC kernel.

16 Programming examples

16.1 PLC main program frame

Programing Example

MCV HLI Interface

```
PROGRAM MAIN
VAR
  HLI                       : MCV_HliInterface [▶ 257];

  HliInitError              : BOOL := FALSE; (* Error at initialisation of
HLI *)
  UserInitialisationDone    : BOOL := FALSE; (* User initialisation done
*)
END_VAR


(* Request description of the HLI from the CNC *)
Hli();

(* Check if initialisation of HLI ended successfully and if
   errors occurred during initialisation phase. *)

IF Hli.Initialised = TRUE AND Hli.Error = FALSE
THEN
  (* Do the initialisation we do once the PLC starts up. *)
  IF UserInitialisationDone = FALSE THEN
    (* Get the result of the user defined initialisation *)
    UserInitialisationDone := UserInitialisations [▶ 262](dummy:=TRUE);
  END_IF;


  (* ---------------------------------------------------- *)
  (* Insert your PLC application code after this comment *)
  (* ---------------------------------------------------- *)


  IF Hli.Error = TRUE THEN
    (* Error on initialisation of the HLI *)
    (* iErrorId contains error number *)

    HliInitError := TRUE;
  END_IF;
END_IF;
```

16.1.1 Initialisation function UserInitialisations()

```
FUNCTION UserInitialisations : BOOL

VAR_INPUT
  dummy : BOOL := FALSE; (* not_used *)
END_VAR
VAR
  AxIdx : UDINT;
  ChIdx : UDINT;
END_VAR


(* Register PLC at all axes interfaces *)
FOR AxIdx := 0 TO gNrAx - 1 DO
  (* Set plc_present_w at each axis *)
  gpAx[AxIdx]^.head.plc_present_w := TRUE;

  (* Register at all axis specific control units you want to handle by
PLC *)
  (* Register at all control units to enable a drive *)
  gpAx[AxIdx]^.lr_mc_control.torque_permission.enable_w := TRUE;
  gpAx[AxIdx]^.lr_mc_control.release_feedhold.enable_w  := TRUE;
  gpAx[AxIdx]^.lr_mc_control.drive_on.enable_w          := TRUE;

END_FOR;


(* Register PLC at all channel interfaces *)
FOR ChIdx := 0 TO gNrCh - 1 DO
  (* Set plc_present_w at each channel *)
  gpCh[ChIdx]^.head.plc_present_w := TRUE;

  (* Register at all channel specific control units you want to handle
by PLC *)

END_FOR;

UserInitialisations := TRUE;
```

Keyword index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z

17 Appendix

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

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