DOCUMENTATION ISG-kernel

Functional description
Forward and backward on path

Short Description:
FCT-C7

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

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

Documentation version: 1.06
07/11/2024

Preface

Disclaimer

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

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

Personnel qualifications

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

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

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

Further information

This link

https://www.isg-stuttgart.de/de/isg-kernel/kernel-downloads.html

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

Disclaimer

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

Trade marks and patents

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

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

Copyright

General and safety instructions

Icons used and their meanings

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

Icons in explanatory text

➢ Indicates an action.

⇨ Indicates an action statement.

⚠ DANGER

Acute danger to life!

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

⚠ CAUTION

Personal injury and damage to machines!

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

Attention

Restriction or error

This icon describes restrictions or warns of errors.

Notice

Tips and other notes

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

Example

General example

Example that clarifies the text.

Programing Example

NC programming example

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

Release Note

Specific version information

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

Table of contents

List of figures

1 Overview

Task

The function for forward/backward motion on the path permits backward motion along the originally programmed path by means of a real-time signal when the NC program is active. Backward motion is terminated by resetting the real-time signal. Forward motion is then resumed.

Properties

Different modes are possible with forward/backward motion and they can be set over the PLC interface. They include suppressing the synchronisation of M/H functions or skipping program sections. Restrictions must be observed for certain functions of backward motion. They are described in detail in the section "Known restrictions [▶ 24]".

Forward/backward motion on the path can be used for:

- Plasma cutting: Clean nozzle and switch extinguished torch back on
- Erosion machining: slit width compensation

Parameterisation

The exact response for forward/backward motion can be configured in the start-up and channel parameters. A complete list of parameters described in this document is contained in the section Parameter [▶ 41].

Programming

In the NC program the command #OPTIONAL EXECUTION... flags a sequence of NC blocks as non-executable in forward/backward motion mode.

Mandatory note on references to other documents

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

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

2 Forward/backward motion on the path

The prepared NC commands (function blocks) for forward/backward motion on the path are saved and then executed by the interpolator in the required sequence.

Backward motion is activated online by the PLC using the “backward motion” control unit [▶ 18]. By default, possible synchronisations of existing M/H functions are suppressed during this type of backward motion and treated like the synchronisation type MOS:

By analogy, the “simulate motion” control unit [▶ 18] suppresses the synchronisation of M/H functions in forward motion. This is then referred to as a “simulate motion”.

In addition, the program command #OPTIONAL EXECUTION [▶ 14] ON/OFF can flag NC program sequences that are completely skipped when “simulate motion” or “backward motion” are active.

The following interfaces are available:

- Configuration of memory size
- Configuration of M function synchronisations in forward/backward mode
- Additional NC program commands
- Control units for “backward motion” [▶ 18], “simulate motion” [▶ 18] and Save ON on the HLI

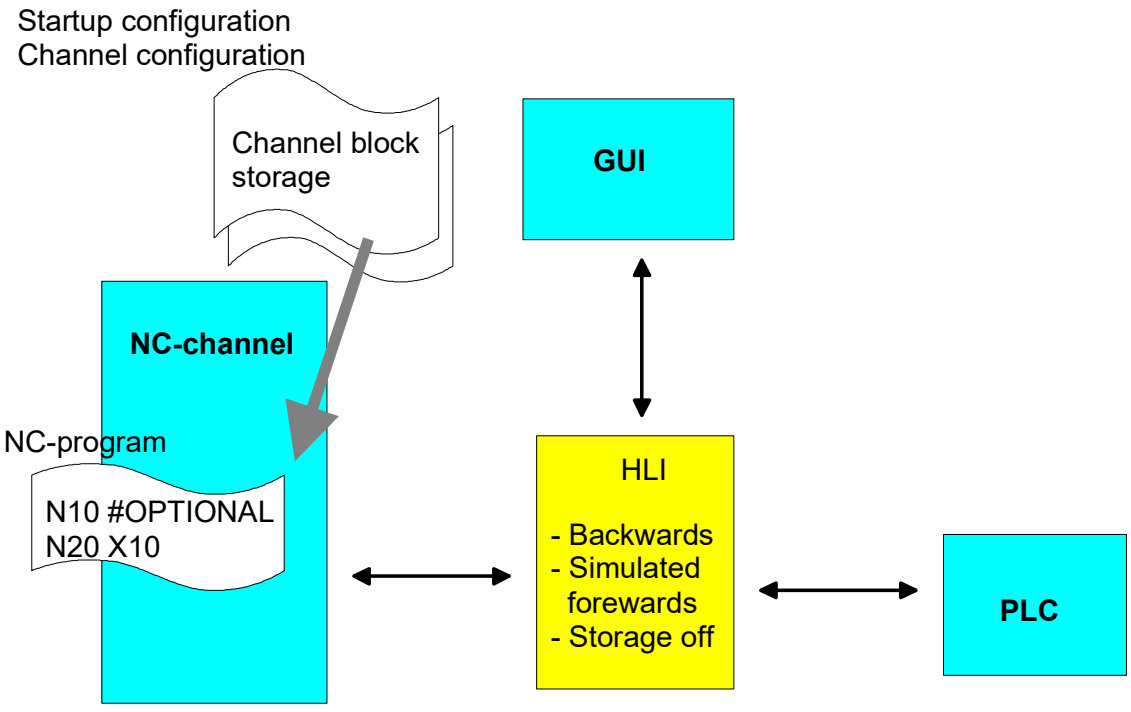

Fig. 1: Interfaces for forward/backward motion on the path

Internal sequence

The figure below illustrates the internal diagram of forward/backward motion on the path. All decoded NC blocks are stored temporarily in a buffer and transferred to the interpolator. With backward motion the already decoded NC commands are inverted and executed in the reverse order.

Notice

The NC program itself is not decoded in the reverse order for backward motion!

After decoding the NC program, all conditions, branches and loops in the NC program are resolved into an linear sequence. In the backward motion memory only this linear resolved NC program can be travelled in forward or backward directions. Therefore, any external influence during decoding (e.g. the actual value of an external variable for a loop break-off condition) is not taken into account a second time for backward motion.

Real-time influences such as feedhold, override and single block mode are treated as normal since their effect is considered by the interpolator at all times.

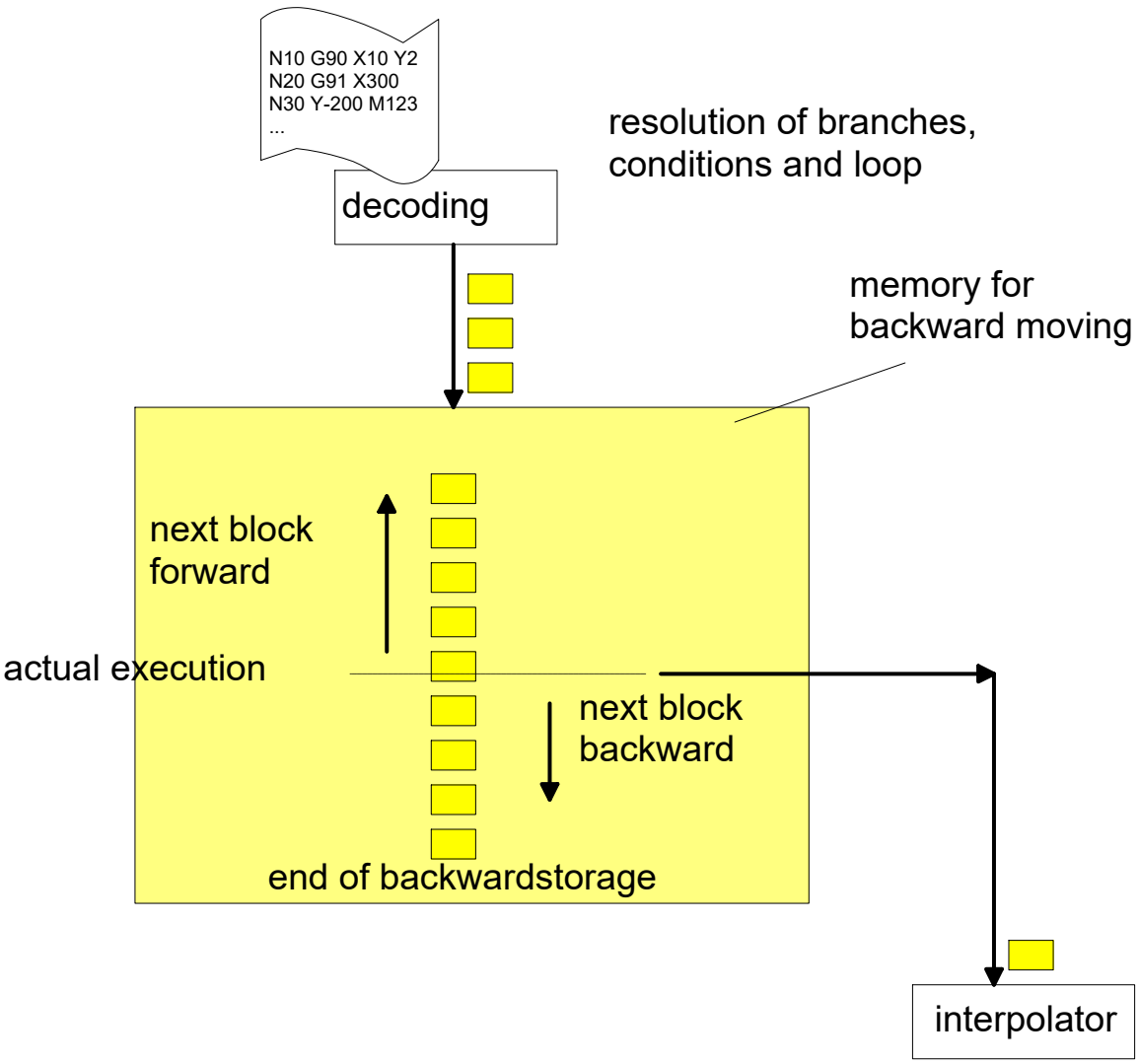

Fig. 2: Saving backward motion commands and transferring them to interpolator on demand

Backward motion memory

The size of the backward motion memory can be defined in the start-up list with P-STUP-00033 (fb_storage_size). When the controller is started up, the NC checks whether the required minimum size is available.

- If this is not the case, a warning is output and the memory size is set to the required minimum value.
- If the size is set to 0, the “forward/backward motion on the path” function is not available.
- The maximum size is only limited by the resources of the automation device (e.g. IPC).

Example

The recommended memory size for the CNC configuration is ~1 Mbyte. Required setting in the start-up list as HEX value:

```
fb_storage_size[0] 0x200000
```

Empirical definition: Increase (or decrease) the memory size until the required path can be travelled backward without an error message.

Notice

The actually required memory depends on the scope of the CNC functions used (e.g. Contouring and special functions to generate additional blocks) and the number of blocks required for backward motion.
Rule of thumb: 1 NC line ≈ 1-5 Kbytes

Memory use for backward motion

The save function for function blocks in the backward motion memory can only be selected/deselected if no NC program is executed and no backward motion is active.

If at program end the memory is deselected and then reselected, the memory is cleared. This ensures that no subsequent backward motion can move back further than this point.

Behaviour at NC reset

In addition, the backward motion memory is cleared at every NC reset.

3 Synchronisation during forward/backward motion

3.1 M/H function handshake with PLC

Synchronisation of M/H functions

In the basic setting, all M and H functions are executed without synchronisation (synchronisation type MOS) when the “backward motion” and “simulate motion” control units are active. If synchronisation is required, an additional bit must be set to parameterise the synchronisation type of the M or H function.

```
BWD_SYNCH       0x400000  (Synchronisation in backward direction )
FWD_SYNCH       0x800000  (Synchronisation in forward direction )
```

They must be set in addition to the other available synchronisation types for M functions (P-CHAN-00041) in the channel parameter list or P-CHAN-00027 for H functions.

Standard forward and repeated forward motion

In the forward direction, the original synchronisation type of the M functions is executed in the same way as in the channel parameters.

Backward - “backward motion” control unit

If the BWD_SYNCH bit is set, the M/H functions are always synchronised by the synchronisation type MVS_SVS for backward motion on the path.

Simulate motion - “simulate motion” control unit

If the FWD_SYNCH bit is set, the M/H function are output with the configured synchronisation types for “simulated” motion occurs.

Combination of backward (“backward motion” control unit) and “simulated” motion (“simulate motion” control unit)

If the BWD_SYNCH bit is set, the M/H functions are always synchronised with the synchronisation type MVS_SVS.

If the FWD_SYNCH bit is set, the M/H functions are always synchronised without synchronisation (MOS).

Both bits are set (BWD_SYNCH and FWD_SYNCH), the M/H functions are always synchronised with the synchronisation type MVS_SVS.

Example

M function parameterisation variants

M2, M101 and M102 are synchronised in backward motion:

```
m_synch[2]              0x00400002
m_synch[101]            0x00400002
m_synch[102]            0x00400004
```

M103 and M104 are synchronised for “simulated” motion:

```
m_synch[103]            0x00800002
m_synch[104]            0x00800004
```

M105 is synchronised for “simulated” motion and backward motion:

```
m_synch[105]            0x00C00002
```

3.2 Suppressing programmed or optional stops

M00/M01 synchronisation

During backward and later forward motion, it may be necessary to suppress a stop due to a programmed M00 or optional M01. For example, the motion may be stopped only during normal forward motion.

The following response of M00/M01 synchronisation can be parameterised in conjunction with forward/backward motions:

- Suppress stop during backward motions
- Suppress stop during next forward motions

The following channel parameters are provided for synchronisation:

Parameter	Parameter name	Value
P-CHAN-00276	`forward_backward.disable_M00_backward`	`0 / 1`
P-CHAN-00277	`forward_backward.disable_M00_2nd_forward`	`0 / 1`
P-CHAN-00278	`forward_backward.disable_M01_backward`	`0 / 1`
P-CHAN-00279	`forward_backward.disable_M01_2nd_forward`	`0 / 1`

M function output to PLC

This does not affect an output of the M function to the PLC and is executed by default as parameterised.

Programing Example

M00 only synchronised in forward motion

M00 should be synchronised in backward and forward directions; by contrast, M01 should not be synchronised at all. The M functions must be parameterised as follows:

```
m_synch[0]          MVS_SVS | BWD_SYNCH | FWD_SYNCH
m_synch[1]          MOS
```

Programing Example

Suppress M01 for repeated forward motion

M00 stop (N900) is suppressed for backward motion. The M00 stop is executed forward both in standard forward motion and repeated forward motion.

The M01 stop is executed for the 1st forward and backward motion. The M01 stop (N901) is suppressed with the next 2nd forward motion.

```
%fbc-m00_m01

N10 X0 Y0 Z0
N20 X100
N30 Y100

N1000 Z3
N1010 X110
N900 M00
N1020 X100
N901 M01
N1030 Z0

N40 X-1
N50 Y-1
...
```

The channel parameters must be set as follows to obtain this response:

```
forward_backward.disable_M00_backward          1
forward_backward.disable_M00_2nd_forward       0
forward_backward.disable_M01_backward          0
forward_backward.disable_M01_2nd_forward       1
```

4 NC program

4.1 Skipping program sequences (#OPTIONAL EXECUTION)

In the NC program, the programming command #OPTIONAL EXECUTION ON/OFF flags a sequence that is to be skipped during backward motion or simulate motion.

Skipping is activated in the PLC. The flagged program part is skipped if

- backward motion is active (backward motion control unit [▶ 18])
- or with simulated (“simulate motion” control unit [▶ 18])

The flagged area is then skipped at interpolator level. However, transition conditions between blocks before and after the skipped area are not recalculated.

Notice

The P-STUP-00033 [▶ 41] function must be parameterised in order to use it

Syntax:

#OPTIONAL EXECUTION [[**ON**] [**[SIMULATE** | **SIMULATE MASK**=.. **]**]] | **OFF**

ON	Activate skipping
OFF	Deactivate skipping
The syntax below is available as of CNC Build V3.3107.12	
SIMULATE	The programmed sequence between #OPTIONAL EXECUTION ON and OFF is only skipped if the signal of the “simulate motion” [▶ 18] control unit is active.
SIMULATE MASK=..	64-bit mask for the specification. The sequence between #OPTIONAL EXECUTION ON and OFF is only skipoped if the signal of the ”simulate motion” [▶ 18] control unit is active and the programmed mask has bitwise matches with the “simulate motion mask” [▶ 21] control unit.

Therefore any interpolator state, especially the axis positions, must remain unchanged to prevent a discontinuous transition of path, velocity and acceleration.

Notice

Axis positions must be identical before and after the skipped sequence.

If axis position are changed the error message ID 50452 is output.

Synchronisation of M/H functions with #OPTIONAL EXECUTION

The sequences flagged with #OPTIONAL EXECUTION ON/OFF are only skipped if backward motion [▶ 18] or “simulate motion” [▶ 18] are active. No M/H functions are output.

The behaviour/options with M or H functions **outside** the sequence are described in [FCT-C7// „M/H function handshake with the PLC [▶ 10]“]

When the command #OPTIONAL EXECUTION ON [SIMULATE] is used, the “simulate motion” [▶ 18] control unit must be active in order to skip the sequence. Therefore, skipping the sequence with backward motion requires backward motion [▶ 18] and “simulate motion” [▶ 18] to be active.

Programing Example

Skipping program sequences

```
%t_storag.nc
X10 Y0
N10 G91 G00 X10 F1000

N11 #OPTIONAL EXECUTION ON
N12 Z123
N13 S1000 M3
N14 Z-123
N15 M101
N16 #OPTIONAL EXECUTION OFF

N20 G90 G01 X0
N30 G02 I10
N40 G03 J10
M30
```

The CNC checks and monitors only the continuous path of axes whether blocks are skipped or not. Any further conditions must be ensured by the user itself and are not checked by the CNC.

Nesting of multiple commands optional execution on/off is not considered.

Before terminating the program level (M17, M29) where OPTIONAL EXECUTION was selected (ON), the function must also be deselected (OFF). This also applies to terminating the main program level (M30). If the program level is terminated without deselection of the function, the error ID 21719 is output.

It is only possible to terminate complete section. If backward motion [▶ 18] or “simuate motion” [▶ 18] are activated when the program is executed within an OPTIONAL_EXECUTION section, the section is not skipped.

Notice

It is not practical to use the NC command #OPTIONAL EXECUTION together with contour-changing functions, e.g. tool radius compensation or polynomial contouring.

Programing Example

Skipping program sequences with “SIMULATE MASK”

In the NC program below, 3 skipped sequences are flagged and are each provided with an identifier in the form of a binary bitmask. These sequences are only skipped when the “simulate motion” [▶ 18] control unit is active and also when the programmed mask has bitwise matches with the “simulate motion mask” control unit [▶ 21].

```
N010 X10 Y0
N020 G91 G00 X10 F1000
N030 #OPTIONAL EXECUTION ON [SIMULATE MASK='2#000001']
N040 X20
N050 M3
N060 X0
N070 M101
N080 #OPTIONAL EXECUTION OFF

N090 #OPTIONAL EXECUTION ON [SIMULATE MASK='2#000010']
N100 X30
N110 M3
N120 X0
N130 M102
N140 #OPTIONAL EXECUTION OFF

N150 #OPTIONAL EXECUTION ON [SIMULATE MASK='2#000100']
N160 X40
N170 M3
N180 X0
N190 M103
N200 #OPTIONAL EXECUTION OFF

N210 X50
N220 X0
N230 M30
```

4.2 Clearing backward storage (#BACKWARD STORAGE CLEAR)

The NC command #BACKWARD STORAGE CLEAR explicitly clears the previous backward storage. This ensures that the function is stopped after this program position is crossed.

Syntax:

#BACKWARD STORAGE CLEAR

Programing Example

Clearing backward storage

```
%backward-storage

N000 G01 X0 F10000
N010 X100 Y123
N020 X100
N030 X200 Y10
N040 X300 Y20

N050 #BACKWARD STORAGE CLEAR

N060 X400 Y-20
N070 X500 Y-3

N060 #BACKWARD STORAGE CLEAR

N080 X444 Y10
N090 X333 Y3
N100 X222 Y10
N110 X111 Y3
N120 X000 Y10
N130 X-111 Y3

N140 #BACKWARD STORAGE CLEAR

N1000 M30
```

5 PLC interface

5.1 Control units

Control units The PLC can command options for forward/backward motion via three control units:

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
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 [▶ 14].
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit
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
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**

External path velocity specified	
Description	External path velocity specified. The path velocity setting is activated by the control unit ext_command_speed_valid. If the velocity specified in negative, the tool moves backwards along the path.(See FCT-C7// Forward/backward motion on the path [▶ 6])
Data type	MC_CONTROL_UNS32_UNIT, see description of Control Unit
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 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 control unit).
Data type	MC_CONTROL_BOOL_UNIT, see description Control unit
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**

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 [▶ 18] and the NC command #OPTIONAL EXECUTION [▶ 14]. When the “simulate motion” control unit [▶ 18] 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
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**

5.1.1 Control units for CNC Builds up to V2.11.20xx

Backward motion	
Description	Select/deselect backward motion on the path
Data type	MCControlBoolUnit, see description of Control Unit
Access	SPS reads Request + State and writes Command + Enable
ST Path	pMC[*channel_idx*]^.addr^.MCControlBahn_Data.**MCControlBoolUnit_BackwardMotion**
Commanded, requested and return values	
ST Element	**.X_Command** **.X_Request** **.X_State**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST Element	**.X_Enable**

Reset backward motion memory	
Description	Deselects backward motion memory No further NC block is saved in the memory. The memory is deleted.
Data type	MCControlBoolUnit, see description of Control Unit
Access	SPS reads Request + State and writes Command + Enable
ST Path	pMC[*channel_idx*]^.addr^.MCControlBahn_Data.**MCControlBoolUnit_ResetBackward-Storage**
Commanded, requested and return values	
ST Element	**.X_Command** **.X_Request** **.X_State**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST Element	**.X_Enable**

Simulated forward motion	
Description	Select/deselect simulated forward motion on the path For example, M function synchronisations are treated differently.
Data type	MCControlBoolUnit, see description of Control Unit
Access	PLC reads Request + State and writes Command + Enable
ST Path	pMC[*channel_idx*]^.addr^.MCControlBahn_Data.**MCControlBoolUnit_SimulateMotion**
Commanded, requested and return values	
ST Element	**.X_Command** **.X_Request** **.X_State**
Data type	BOOL
Value range	[TRUE, FALSE]
Redirection	
ST Element	**.X_Enable**

5.2 Notes

M functions during “forward/backward motion on the path”

By default the synchronisations of M functions are suppressed for backward motion on the path. However, if required, synchronisation can be enabled by specific bits.

When these bits are used, all synchronised M functions are treated as synchronisation type MVS_SVS (application before block, synchronisation before block) during backward motion. In “simulated forward motion” mode the synchronisation type is used as specified in the channel parameters.

Memory use for backward motion

The save function for function blocks in the backward motion memory can only be selected/deselected if no NC program is executed and no backward motion is active.

If at program end the memory is deselected and then reselected, the memory is cleared. This ensures that no subsequent backward motion can move back further than this point.

NC reset

In addition the backward motion memory is cleared at every NC reset.

6 Known restrictions

- A motion with an external measurement signal (G100) is only executed completely the first time. The external measurement signal is then ignored for every further backward/forward motion.
- It is not possible to move a homing motion backwards since this motion has an external influence on axis motions (cam signal) and they cannot be executed in a backwards direction. Even digital drives can execute the homing motion independently without CNC. Therefore, G74 is not permitted for backward motion. All saved commands are deleted and the warning P-ERR-50449 is output.
- Manual mode without parallel interpolation (G200) and manual mode with superimposed interpolation (G201) are not treated in backward/forward mode.
- Motions in manual mode are not saved or inverted in forward/backward mode due to pressed keys or handwheel increments.
- Spindle-specific commands (speed/position) are not inverted.
- Explicit synchronisation for independent axis cannot be inverted during backward motion.

7 Backward movement with external position offsets

External position offset

During “forward/backward on the path” the programmed path may be offset by external online influences.

This position offset can be signalled to the entire NC channel (see #CHANNEL INIT[CMDPOS]) by subsequently synchronising the channel. This deletes the offset, i.e. an absolute programmed position then does not include an offset.

Backward motion by position initialisation with offsets

In backward/forward motion, external offsets are not moved back in the same way as other motions specified in the NC program. If a backward motion moves to a position initialisation with an external offset, there are 2 options. They can be set using the channel parameter P-CHAN-00275:

1. P-CHAN-00275 is not set (default); no further backward motion is permitted since the positions specified in the NC program must also be approached without offset, even in a backward direction.
2. P-CHAN-00275 is set; the offset is retained and the backward motion may resume. The current absolute positions defined in NC program are no longer valid since they are shifted by the currently valid offset.

Notice

If P-CHAN-00275 is set, backward motion reverts to an offset path starting at the position initialisation. See the figure below

Possible offsets are:

- Manual mode actions
- Offsetting path by 'Jog of Path' (see [FCT-C15] “Insert a program“)
- Values are added to the position controller via the additional interface
- Axis coupling via PLC
- Corrections via the online tool compensation

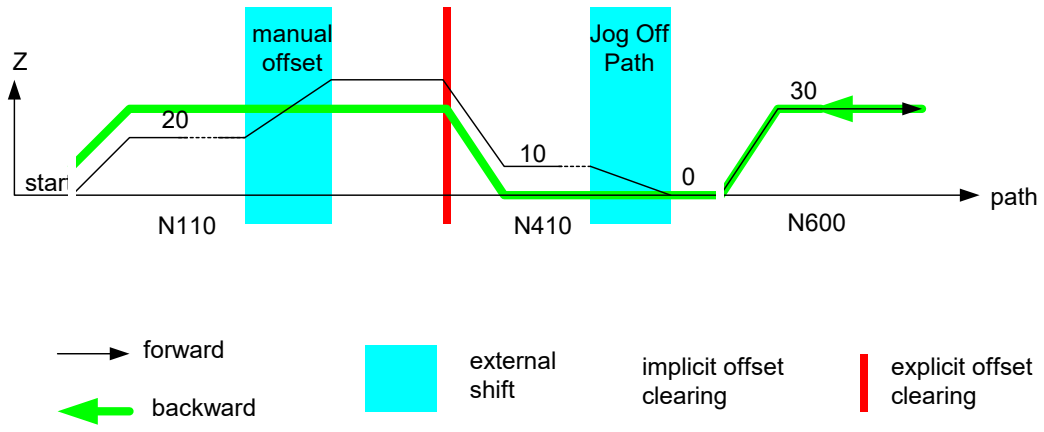

Fig. 3: Motion in backward/forward direction with offsets

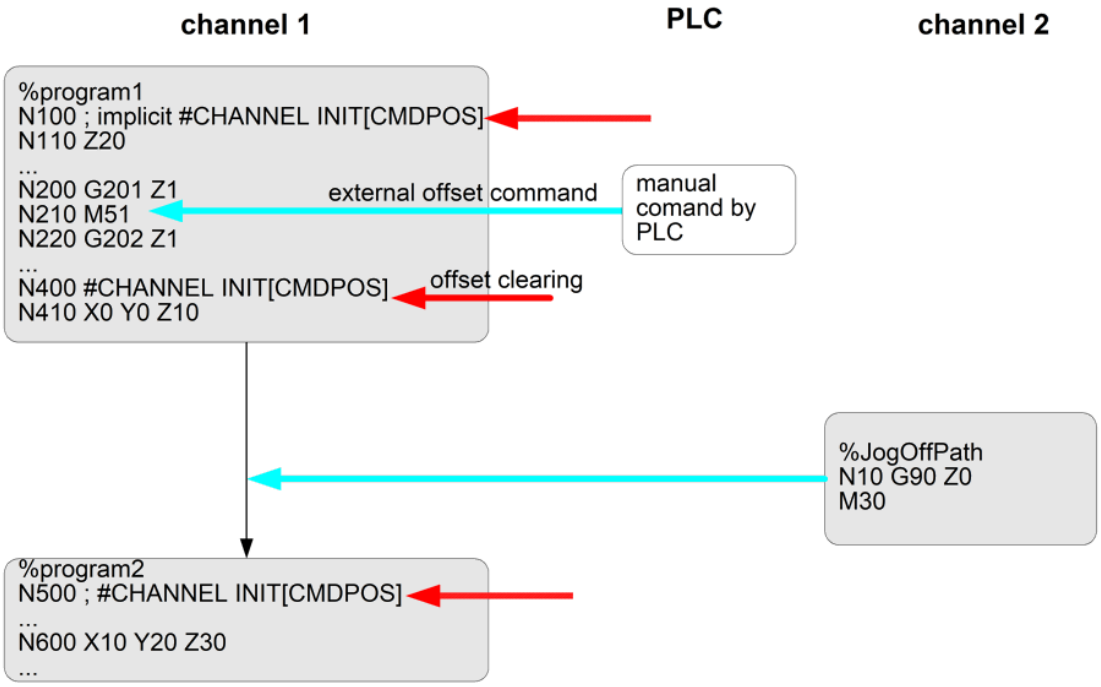

Fig. 4: Program execution with offset overlapping

Attention

An active kinematic transformation which is especially dynamically different depending on the position, may cause a dynamic overload (or an incomplete loading) of the axes.

Background: If a backward motion has an offset, a deviation is made from the original path. However, dynamic planning was executed with the original path motion in forward direction without offset.

Notice

Online influences during backward motion such as

- Measurement with G100,

- Homing with G74 or

- Deselecting online tool compensation by #OTC OFF if P-CHAN-00275 when P-CHAN-00275 is not set

The backward motion is ended and the warning is output.

8 Backward motion after block search

If a backward motion is executed after block search is used, the actually programmed NC program is interpolated in backward direction as of the start position from the block search.

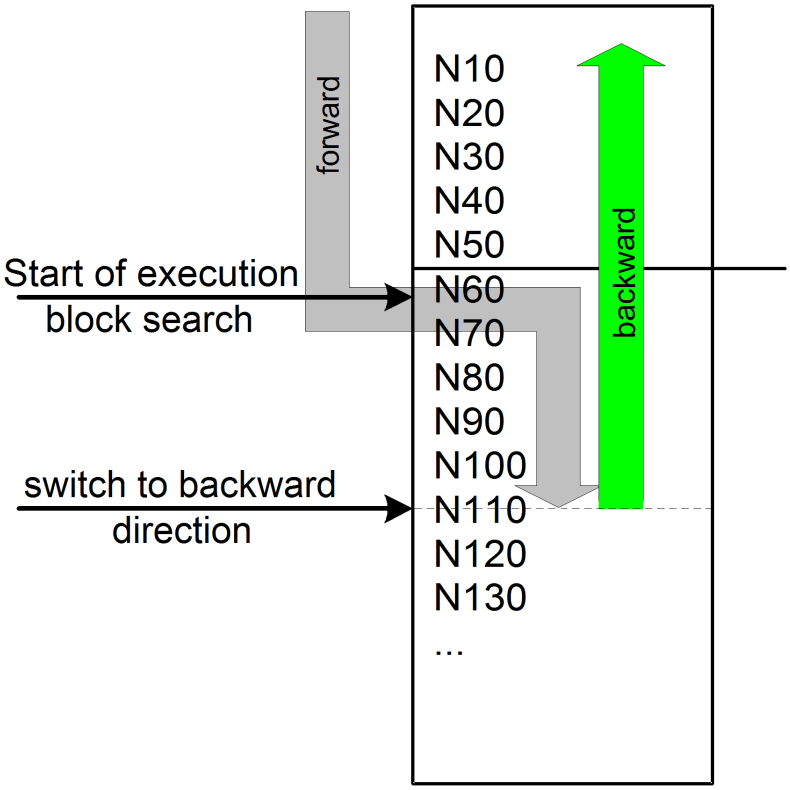

Fig. 5: Backward motion after block search

9 Automatic reversal after stop (#STOP REVERSIBLE)

Release Note

This option is available as of Build V3.1.3039.01.

This function defines a STOP mark in the NC program at which the machining direction can be inverted without acknowledgement by the PLC.

Use and response of the reversible STOP

- **Definition** of STOP mark in the NC program
  If the STOP mark is reached during forward machining, the CNC decelerates the velocity = 0 in good time and waits for the enable by the PLC to resume the motion.
- **Enabling** STOP marks by PLC
  If the STOP marks are not enabled, no deceleration or stop is executed when the mark is reached (compare M01, optional stop) as if the stop was not programmed in the NC program.
- A LEVEL bit can be specified at each STOP mark. This permits the combination of **stops in groups** so that they can be enabled individually by the PLC.
- **Resuming** after stopping at the STOP mark by the PLC
  When a stop occurs due to a stop condition, it is possible to resume the motion in the original direction due to the **falling edge** of the "continue motion" command (compare Continue motion after M00/M01)
- **Reversing** at stop
  If the machining direction reverses at this stop/wait, this action occurs directly without waiting for the enable to resume. The motion is executed in the reverse direction.

Optional parameters

The CNC transfers an additive value directly to the PLC. The PLC can save various functions based on this value.

#STOP REVERSIBLE [USR_VAL <*val*>]

Similar to M00/M01 a suppression of STOP marks can be configured in the channel parameter list. For example STOP marks can be suppressed only once in the forward or backward motion.

```
forward_backward.disable_stop_1st_forward      0 / 1
forward_backward.disable_stop_2nd_forward      0 / 1
forward_backward.disable_stop_backward         0 / 1
```

The effectiveness of the stops can be globally predefined in the channel parameter list and then individually overwritten in the NC command.

9.1 Interfaces / NC commands

The following interfaces are available:

- NC command #STOP
- HLI::IPO-Command::StopLevel
- HLI::IPO-Command::ContinueMotion
- HLI::IPO-State::IsStopped

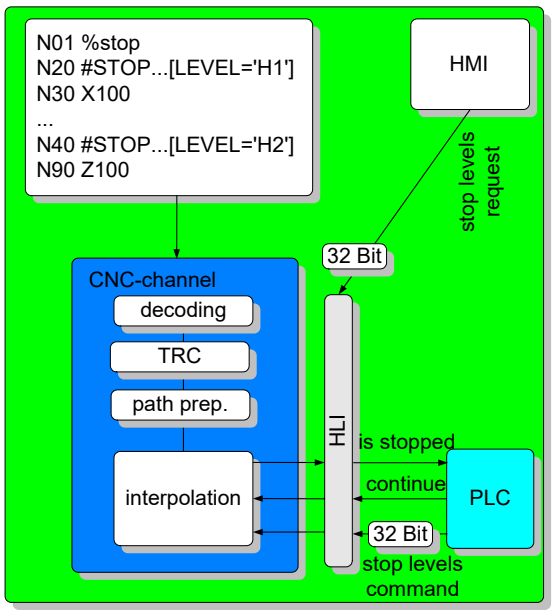

Fig. 6: Interfaces for stop for backward/forward motion

9.1.1 NC command

A STOP flag is programmed with the following commands:

#STOP REVERSIBLE	**[** [**LEVEL** <32bit>] [**1ST_FORWARD** <val>] [2ND_FORWARD <val>] [BACKWARD <val>] [USR_VAL <32bit-val>]]
LEVEL	Bit value to enable the STOP flag via the PLC interface. Default: 0, stop is always executed, i.e. without explicit PLC enable.
1ST_FORWARD	The stop condition is executed or suppressed during standard forward motion. Default: Setting in the channel parameter list: `forward_backward.disable_stop_1st_forward`
2ND_FORWARD	The stop condition is executed for repeated forward motions after a preceding backward motion. Default: Setting in the channel parameter list: `forward_backward.disable_stop_2nd_forward`
BACKWARD	The stop condition is executed during a backward motion. Default: Setting in the channel parameter list: `forward_backward.disable_stop_backward`
USR_VAL	Additional value directly transferred by the CNC to the PLC. Default: 0

9.1.2 HLI state

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 [▶ 31] 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” is reached.
HLI_SC_DEC_SYN_CHAN_EMPTY	0x10000000	Decoder is waiting for synchronisation. NC channel has no jobs.

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) [▶ 32] and the functional description [FCT-C7// Automatic reversing after stop [▶ 28]].
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

9.1.3 HLI commands

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" (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" (wait_axes_in_position_r).
Data type	MC_CONTROL_UNS32_UNIT, see description Control Unit
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**

Resumption of motion	
Description	If program execution is interrupted by selecting "Single block mode" or"Optional stop" 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
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**

9.2 Examples

Example

Automatic reversing after stop

9.2.1 Example 1: Reversible stop

If the stop condition N45 is acknowledged once in the next NC program, only a forward/backward motion can be executed in the N50 - N90 range without the need for a further acknowledgement.

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
N50 X0 Y0 Z0
N60 X100
N70 Y100
N80 X0
N90 Y0
N95 #STOP REVERSIBLE
M30
```

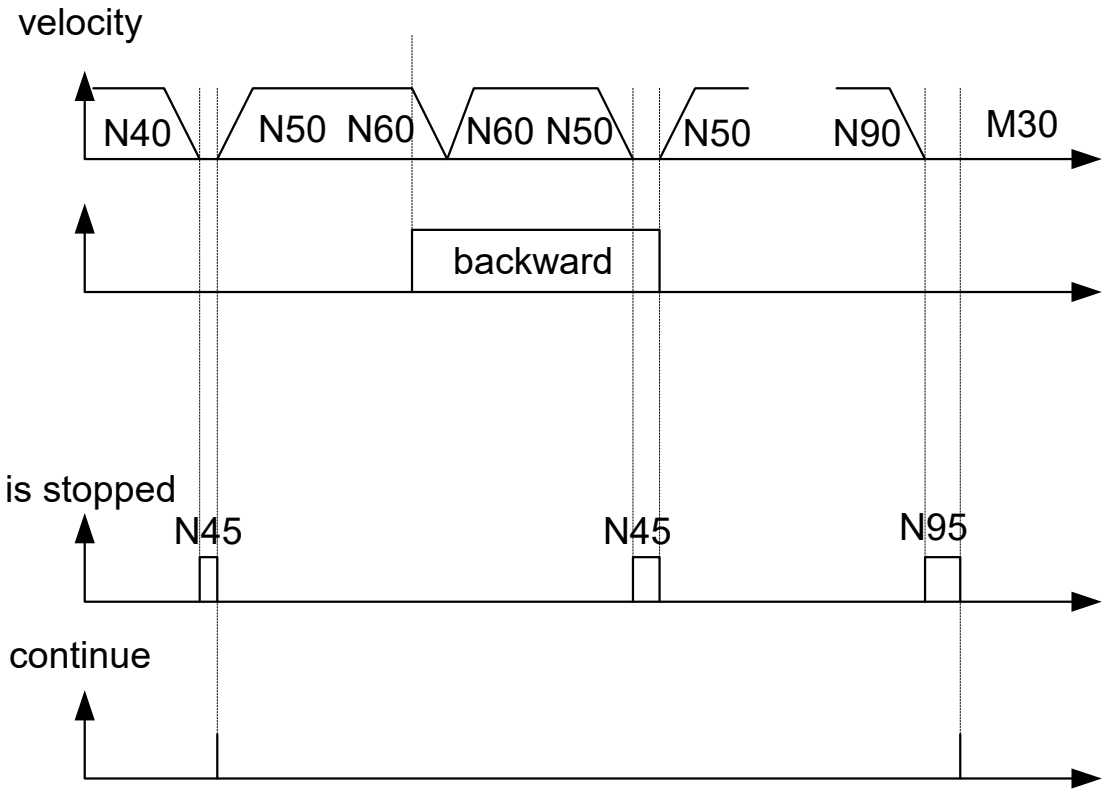

Fig. 7: Schematic diagram of Reversible Stop

9.2.2 Example 2: Option to disable the reversible STOP

In the next NC program the parameterisation suppresses the stop condition N45 in the backward direction and in the forward direction during the second motion.

Channel parameters

```
forward_backward.disable_stop_backward      1
forward_backward.disable_stop_2nd_forward   1
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
N50 X0 Y0 Z0
N60 X100
N70 Y100
N80 X0
N90 Y0
N95 #STOP REVERSIBLE
M30
```

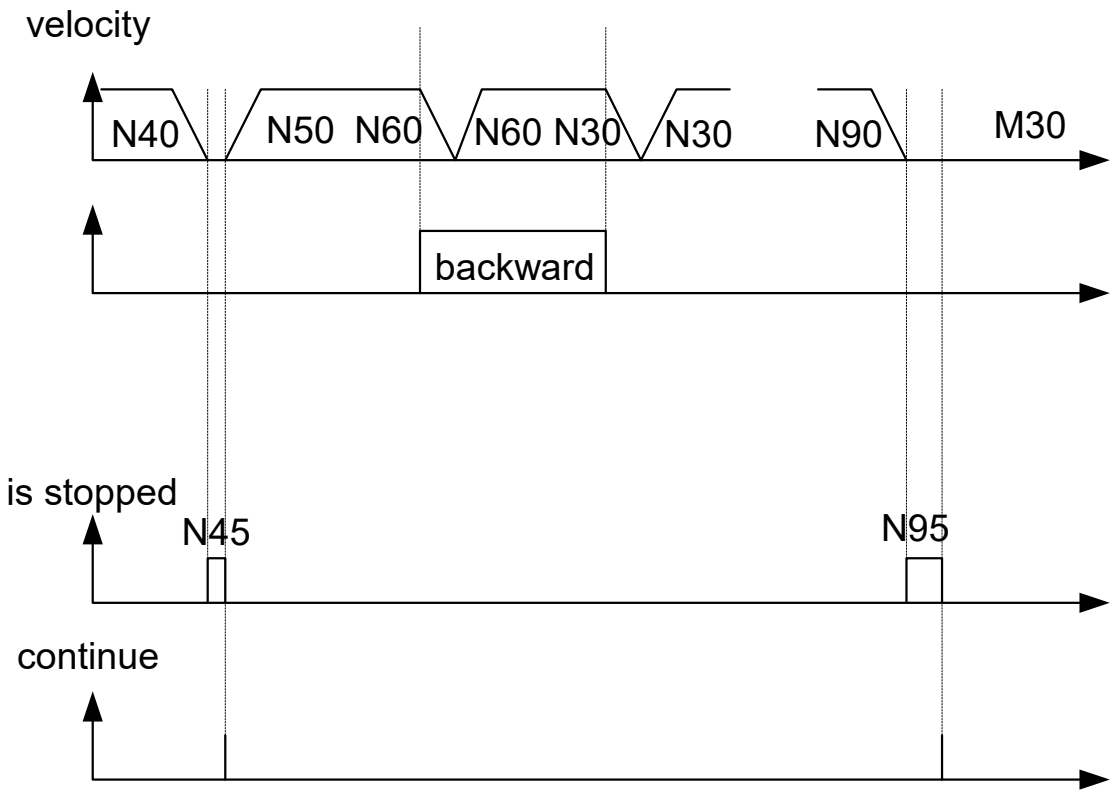

Fig. 8: Deactivating reversible stop

9.2.3 Example 3: Additional parameters for optional stop

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

Channel parameters

```
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
```

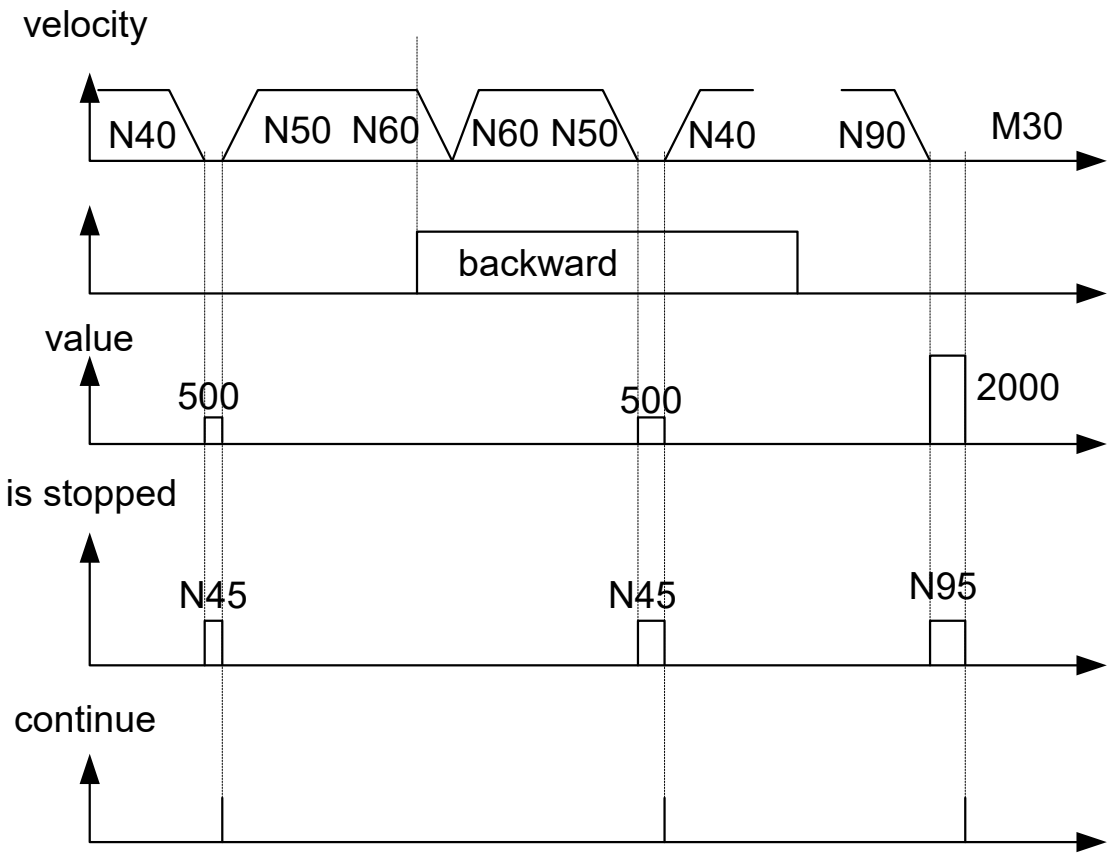

Fig. 9: Additional parameters for optional stop

9.2.4 Example 4: Enabling the programmed STOP by the PLC

The stop condition N45 is suppressed in the backward direction since the LEVEL bit = 16#1 is not set for the motion.

NC program

```
%stop_reversible
N01 X0 Y0 Z0
N10 X100
N20 Y100
N30 X0
N40 Y0
N45 #STOP REVERSIBLE[ LEVEL = '16#01']
N50 X0 Y0 Z0
N60 X100
N70 Y100
N80 X0
N90 Y0
N95 #STOP REVERSIBLE[ LEVEL = '16#4000']
M30
```

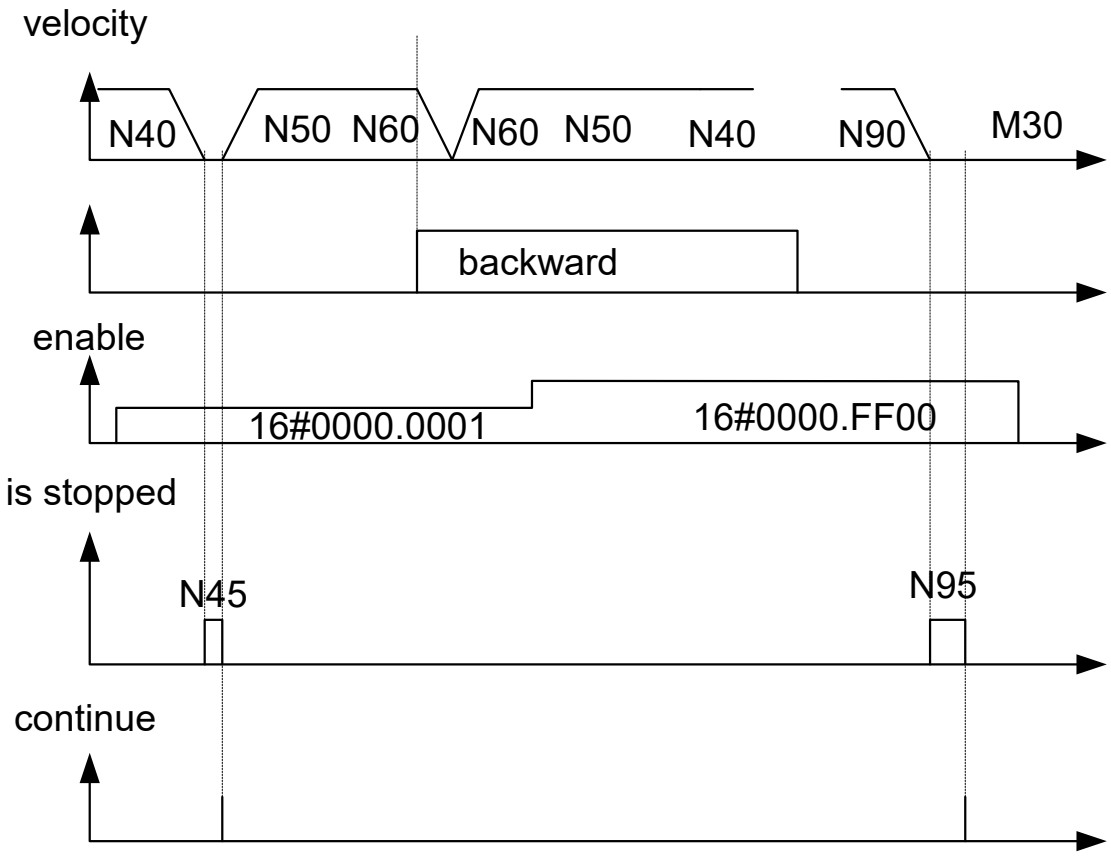

Fig. 10: Enabling the programmed STOP by the PLC

9.2.5 Example 5: Parameterising the programmed stop

The effectiveness of the stop can be preassigned in the channel parameter list and overwritten in individual NC commands.

In the example the stop condition N25 is suppressed in the first motion. In the same way, the stop condition N65 is suppressed in the backward direction.

Channel parameters

```
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
N25 #STOP REVERSIBLE[ 1ST_FORWARD=0]
N30 X0
N40 Y0
N45 #STOP REVERSIBLE[ 2ND_FORWARD=0]
N50 X0 Y0 Z0
N60 X100
N65 #STOP REVERSIBLE[ BACKWARD=0]
N70 Y100
N80 X0
N90 Y0
M30
```

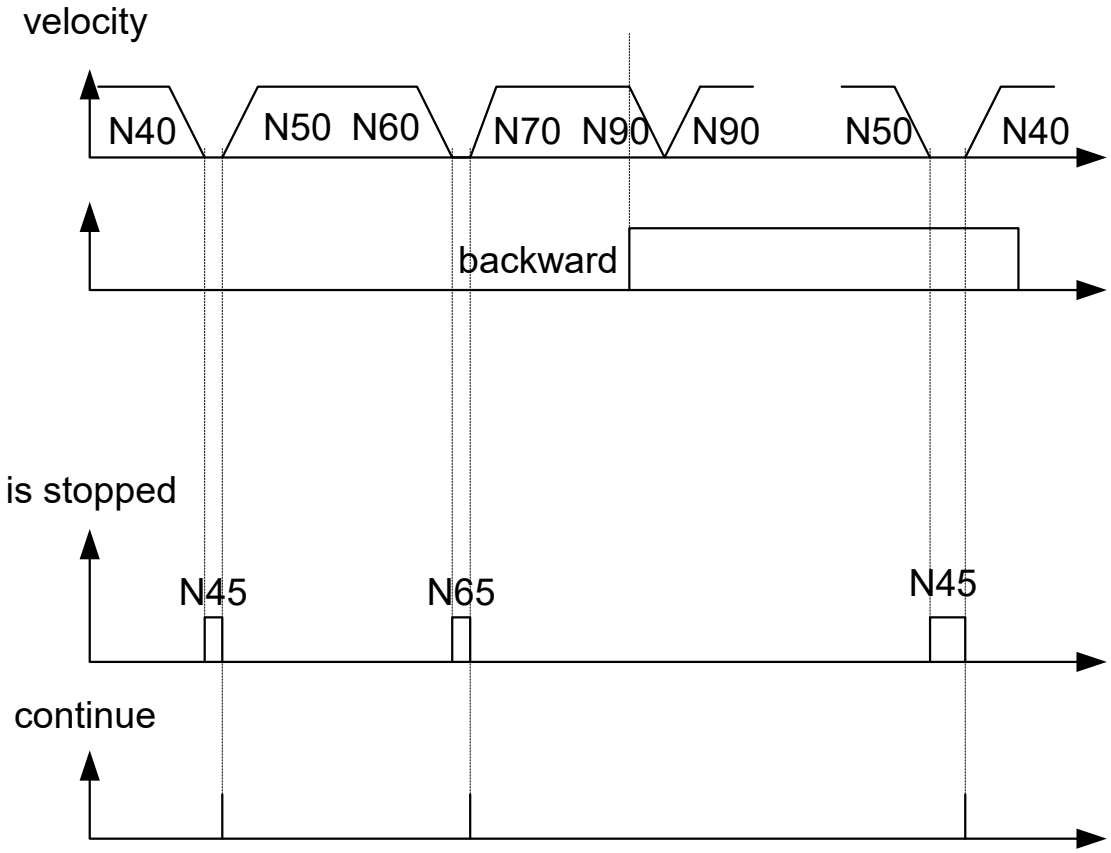

Fig. 11: Suppressing the stop interaction in backward/forward motion

10 Parameter

10.1 Overview

Constant	Description
M_FKT_ANZ	Maximum number of M functions [SYSP, Section "System parameters CNC"]
KANAL_ANZ_RND	Maximum number of independent channels [SYSP, Section "System parameters CNC"]

ID	Meaning
P-STUP-00033	Memory size for backward motion
P-CHAN-00041	Synchronisation types for M functions
P-CHAN-00275	Backward motion with offset
P-CHAN-00276	Programmed M00 stop during backward motions
P-CHAN-00277	Programmed M00 stop during next forward motions
P-CHAN-00278	Programmed M01 stop during backward movement
P-CHAN-00279	Programmed M01 stop during next forward movement
P-CHAN-00308	Response at STOP marks during forward movement
P-CHAN-00309	Response at STOP marks during repeated forward movement
P-CHAN-00310	Response at STOP marks during backward movement

10.2 Description

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

P-CHAN-00041	Synchronisation type of M functions
Description	In the array m_synch[i], the synchronization type of the corresponding M function is defined. Here, the field index 'i' defines the number of the M function. The value indicates the synchronisation type of the M function, i.e. when a check is made for presence of a PLC acknowledgement. A motion is not executed, or is stopped at the latest towards the end of the block, if no acknowledgement has arrived from the PLC. The synchronisation type is defined as a string constant or a hexadecimal value.
Parameter	m_synch[i] where i = 0 999 (maximum number of M functions, application-specific)
Data type	STRING
Data range	See the table below.
Dimension	----
Default value	NOT_VALID *
Remarks	M functions are consumption information and must be fetched (read) by the PLC. This also applies to M functions of the type MOS, MOS_TS, MEP_MOS and MET_MOS. Otherwise, this results in a blocked interface to the HLI in the CNC and as a consequence to an unexpected processing stop. ***Note:** **The default value for internal M functions (M0, M1, M2, M17, M29, M30, M3, M4, M19) is NO_SYNCH.** **Caution:** The following applies to synchronisation types with associated time and path-related pre-output (MET_SVS, MET_MOS, MEP_SVS, MEP_MOS): If one of these synchronisation types is later changed into one which requires no pre-output value, P-CHAN-00070 (m_pre_outp[i]) must be assigned to 0. Otherwise, a license error is generated in case of microjoints if this function is not licensed or not enabled (seeP-CHAN-00600 Alternatively: P-STUP-00060) is inactive. Example: m_synch[12] MVS_SVS 0x00000002 Note: Programming a UNS32 variable is permissible for downward compatibility reasons. Example: m_synch[12] 0x00000002

Constant	Value	Meaning
NOT_VAILD	-1	No valid M function
NO_SYNCH	0x00000000	No output of M function to PLC
MOS	0x00000001	Output of M function to PLC without synchronisation. If the M function is programmed within a motion block, the output of the M function is executed before the movement. M function must be fetched from PLC!
MVS_SVS	0x00000002	Output of the M function to the PLC before the motion block, synchronisation before the motion block
MVS_SNS	0x00000004	Output of the M function to the PLC before the motion block, synchronisation after the motion block
MNS_SNS	0x00000008	Output of the M function to the PLC after the motion block, synchronisation after the motion block
MNE_SNS	0x00000020	Output of M function to PLC after measurement event and removal of distance to go, synchronisation after motion block (for edge banding option only)
MVS_SLM	0x00004000	Late synchronisation, output of M function to PLC within the block, synchronisation during transition to G01/G02/G03 (implicit synchronisation)
MVS_SLP	0x00008000	Late synchronisation, output of M function to PLC within the block, synchronisation by NC command #EXPL SYN (explicit synchronisation)
MOS_TS	0x00040000	Output of the M function to the PLC before motion block without synchronisation, CNC calculates sampling time offset for high-precision time output in the PLC. M function must be fetched from PLC.
MEP_MOS	0x00100000	Pre-output of M function with specified path, without synchronisation. M function must be fetched from PLC.
MET_MOS	0x00200000	Pre-output of M function with specified time, without synchronisation. M function must be fetched from PLC.
BWD_SYNCH	0x00400000	Synchronisation of M function during backward motion with MVS_SVS
FWD_SYNCH	0x00800000	Synchronisation of M function during 'Simulated forward motion' based on the defined synchronisation type
MEP_SVS	0x01000000	Output of M function with specified path, synchronisation before next block
MET_SVS	0x02000000	Pre-output of M function with specified time, synchronisation before next block
FAW_SYNCH	0x10000000	Decoding stop (Flush and Wait): Output of M function to PLC and stop of program decoding at block end until program run is completed. FAW_SYNCH can be set in addition to other synchronisation types. M functions with FAW_SYNCH may not be used when tool radius compensation (TRC), polynomial contouring and HSC mode are active.

P-CHAN-00275	Backward movement with external position offsets
Description	The programmed path contour can be shifted by external online influences. Subsequent synchronisation can signal this position offset to the entire NC channel (see #CHANNEL INIT[CMDPOS]). This deletes the offset, i.e. an absolute programmed position then does not include an offset. During forward/backward movement, external offsets are not moved backward like other motions defined in NC program. If there is backward movement on a program position with an external offset, there are two options: 1. A further backward movement is not allowed since the positions specified in the NC program must also be reached in the backward direction without offset. 2. The offset is retained and backward movement is allowed to continue. The current absolute positions defined in NC program are no longer valid since they are shifted by the currently valid offset. These offsets may be caused by: • Manual mode actions • Shifting the contour with 'Jog of Path' (see [FCT-C15]) • Compensations executed online (see [FCT-C20]
Parameter	forward_backward.with_offset
Data type	BOOLEAN
Data range	0: No backward movement is possible over the program position of an external offset. The backward movement storage is cleared automatically (as for a #BACKWARD STORAGE CLEAR). 1: A backward movement is possible beyond the program position of an external offset is possible. The active offset at the time of reversal is retained.
Dimension	----
Default value	0
Remarks	The backward movement is stopped when the following online influences occur: - Measurement with G100 - Homing with G74 - Deselection of Online Tool Compensation with #OTC OFF

P-CHAN-00276	Programmed M00 stop during backward movements
Description	Suppress stop during backward movements with M00.
Parameter	forward_backward.disable_m00_backward
Data type	BOOLEAN
Data range	0: The process also stops in the backward direction with M00. 1: Stopping with M00 is omitted in the backward direction.
Dimension	----
Default value	0
Remarks	

P-CHAN-00277	**Programmed M00 stop during next forward movements**
Description	Suppress stop in next forward movements with M00.
Parameter	forward_backward.disable_m00_2nd_forward
Data type	BOOLEAN
Data range	0: Motion is also stopped with M00 in the next forward motion. 1: Stopping with M00 is omitted in the next forward direction.
Dimension	----
Default value	0
Remarks	

P-CHAN-00278	**Programmed M01 stop during backward movement**
Description	Suppress stop during backward movement with M01.
Parameter	forward_backward.disable_m01_backward
Data type	BOOLEAN
Data range	0: The process also stops in backward direction with M01. 1: Stopping with M01 is omitted in the backward direction.
Dimension	----
Default value	0
Remarks	

P-CHAN-00279	**Programmed M01 stop during next forward movement**
Description	Suppression of stop on next forward movement with M01.
Parameter	forward_backward.disable_m01_2nd_forward
Data type	BOOLEAN
Data range	0: The process is also stopped in next forward direction with M01. 1: Stopping with M01 omitted in next forward direction.
Dimension	----
Default value	0
Remarks	

P-CHAN-00308	Response at STOP marks during backward movement
Description	Suppress stop at STOP during backward movement.
Parameter	forward_backward.disable_stop_backward
Data type	BOOLEAN
Data range	0: The movement stops at the STOP mark during backward movement. 1: The STOP mark is ignored during backward movement
Dimension	----
Default value	0
Remarks	This parameter is available as of CNC Build V3.1.3039.01.

P-CHAN-00309	Response at STOP marks during forward movement
Description	Suppress stop at STOP during forward movement.
Parameter	forward_backward.disable_stop_1st_forward
Data type	BOOLEAN
Data range	0: The motion stops at the STOP mark during forward movement. 1: The STOP mark is ignored during forward movement.
Dimension	----
Default value	0
Remarks	This parameter is available as of CNC Build V3.1.3039.01.

P-CHAN-00310	Response at STOP marks during repeated forward movement
Description	Suppress stop at STOP during forward movement after previous backward movement.
Parameter	forward_backward.disable_stop_2nd_forward
Data type	BOOLEAN
Data range	0: The motion always stops at the STOP mark during repeated forward movement. 1: The STOP mark is ignored during forward movement after previous backward movement.
Dimension	----
Default value	0
Remarks	This parameter is available as of CNC Build V3.1.3039.01.

11 Appendix

11.1 Suggestions, corrections and the latest documentation

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

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

The link above forwards you to:

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

Notice

Change options for favourite links in your browser;

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

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

PDFs for download:

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

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

E-Mail: documentation@isg-stuttgart.de

Keyword index

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

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