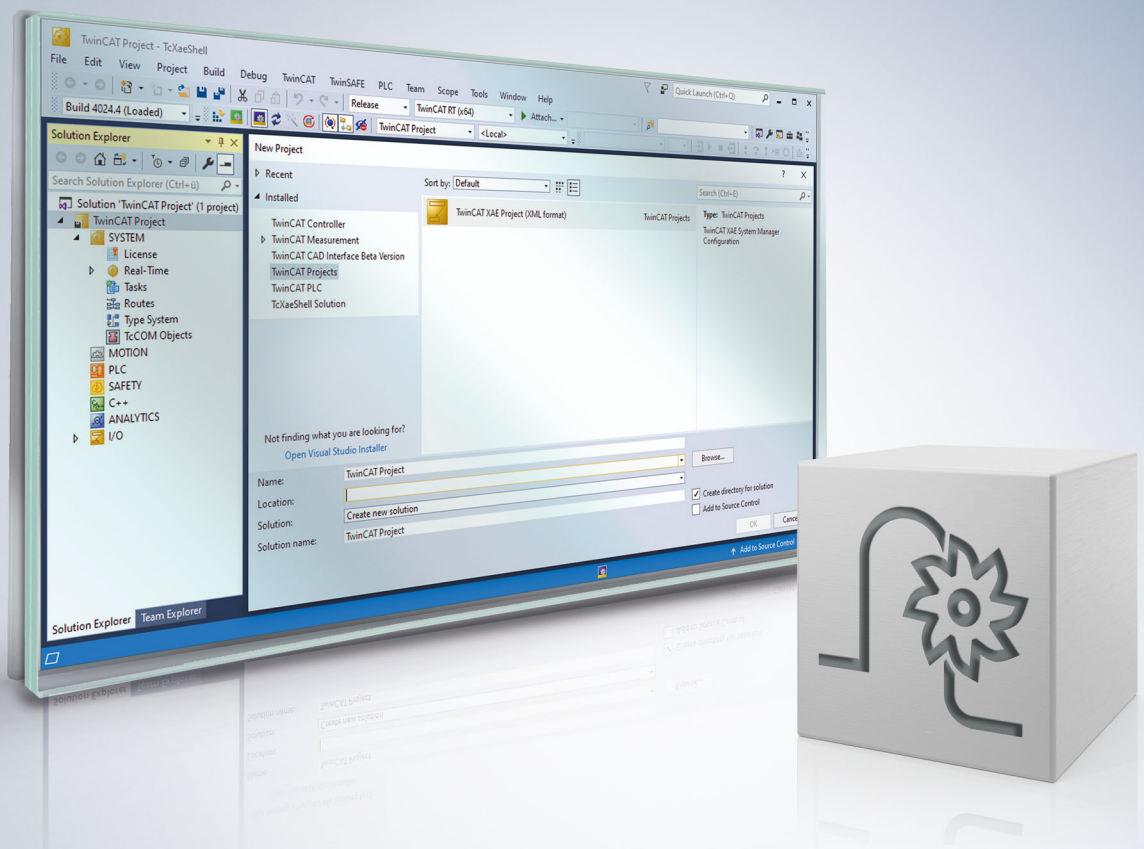
BECKHOFF New Automation Technology

Functional description | EN

TF5200 | TwinCAT 3 CNC

M/H functions



12/18/2023 | Version: 1.06

Notes on the documentation

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

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

Disclaimer

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

Trademarks

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

Patent Pending

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

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

Copyright

Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

General and safety instructions

Icons used and their meanings

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

Icons in explanatory text

1. Indicates an action.

⇨ Indicates an action statement.

⚠ DANGER

Acute danger to life!

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

⚠ CAUTION

Personal injury and damage to machines!

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

NOTICE

Restriction or error

This icon describes restrictions or warns of errors.

Tips and other notes

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

General example

Example that clarifies the text.

NC programming example

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

Specific version information

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

Table of contents

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

Notes on information security ... **4**

General and safety instructions ... **5**

1 Overview ... **10**

2 Description ... **11**

2.1 Range of freely user-definable M/H numbers ... 13
2.2 Default M functions ... 14
2.3 Parameterising the synchronisation methods ... 16
2.3.1 Examples of the synchronisation methods in combination with motions ... 18
2.3.2 Example with MVS_SVS ... 18
2.3.3 Example with MVS_SNS ... 19
2.3.4 Example with MNS_SNS ... 20
2.3.5 Example with MNE_SNS ... 21
2.3.6 Example of MVS_SLM ... 22
2.3.7 Example of MVS_SLP ... 23
2.3.8 Response of 'late' synchronisation (MVS_SLM, MVS_SLP) at program end ... 24
2.3.9 Example with synchronisation output brought forward MEP_SVS and MET_SVS ... 25
2.3.10 Defining the lead distance or time lead ... 27
2.3.11 Example with MEP_MOS and MET_MOS ... 27
2.3.12 Example with MOS_TS ... 29

3 Channel and axis-/-specific M/H functions ... **32**

3.1 General programming ... 32
3.2 M/H functions with additional information ... 33
3.3 Production time calculation ... 35
3.4 Pre-output of M functions (MicroJoint) ... 35
3.4.1 Limitations, special cases ... 39
3.4.2 Tolerance parameters for the permitted deviation between real and planned pre-output positions of an M/H function ... 43
3.4.3 Explicit feed programming for MicroJoints (#CHANNEL SET) ... 44
3.5 HMI display ... 48

4 Spindle M functions ... **51**

4.1 Examples of spindle M functions ... 52
4.1.1 Example 1a: CNC spindle, internally synchronised ... 52
4.1.2 Example 1b: CNC spindle, internal and synchronised by the PLC ... 53
4.1.3 Example 2a: PLC spindle ... 54
4.1.4 Example 2b: PLC spindle with output of M functions via the channel range ... 55
4.2 Gear speed selection/parameter set changeover ... 56

5 Programming ... **58**

5.1 Programming using the example of M functions, channel- and axis-specific ... 58
5.1.1 Example 1: Programming a channel-specific M function ... 58
5.1.2 Example 2a: Programming an axis-specific M function in extended syntax ... 59
5.1.3 Example 2b: Programming an axis-specific M function in DIN syntax ... 60
5.2 Other examples ... 61

6 Parameter 63
6.1 Overview 63
6.2 Description 63
6.3 Method to create an M or H function in the channel parameter list 77
7 Support and Service 78
Index 79

List of figures

Fig. 1 M and H functions are enabled for each CNC channel in the channel parameter list via their synchronisation method. ... 11
Fig. 2 Synchronisation type MVS_SVS ... 18
Fig. 3 Synchronisation type MVS_SNS ... 19
Fig. 4 Synchronisation type MNS_SNS ... 20
Fig. 5 Synchronisation method MNE_SNS ... 21
Fig. 6 Synchronisation type MVS_SLM ... 22
Fig. 7 Synchronisation type MVS_SLP ... 23
Fig. 8 Synchronisation types MET_SVS, MEP_SVS ... 25
Fig. 9 Activate residual path/time calculation with M/H code look ahead ... 26
Fig. 10 Planned output time for MET_SVS ... 27
Fig. 11 Synchronisation types MET_MOS and MEP_MOS ... 28
Fig. 12 Synchronisation method MOS_TS ... 29
Fig. 13 Overview of dead times in system chains ... 30
Fig. 14 Diagram of dead time compensation ... 31
Fig. 15 Programmed MicroJoints in the part ... 36
Fig. 16 Limitation of the pre-output path of the M function to 10 blocks ... 40
Fig. 17 Explicit limitation of the lead distance of the M function ... 41
Fig. 18 Theoretical overlapping of MicroJoints in the part ... 42
Fig. 19 Limitation of overlapping of MicroJoints in the part ... 42
Fig. 20 Attempted pre-output with another M/H function of the MVS_SNS type ... 43
Fig. 21 Feed definition with MicroJoints ... 44
Fig. 22 Feed with block global MicroJoints ... 45
Fig. 23 Increased feed with M11, decreased feed with M12 ... 46
Fig. 24 Suppress VECTOR_LIMIT during MicroJoint ... 47
Fig. 25 AmsAdsViewer- Reads channel-specific H functions ... 49
Fig. 26 AmsAdsViewer- Reads axis-specific H functions ... 50
Fig. 27 CNC spindle, internally synchronised ... 52
Fig. 28 CNC spindle, internal and synchronised by the PLC ... 53
Fig. 29 PLC spindle ... 54
Fig. 30 PLC spindle with output of M functions via the channel range ... 55
Fig. 31 Activating gear changing ... 57
Fig. 32 Programming a channel-specific M function ... 58
Fig. 33 Programming a channel-specific M function ... 59
Fig. 34 Programming an axis-spec. M function in DIN syntax ... 60
Fig. 35 Output of M83 ... 61
Fig. 36 Flow diagram to create an M/H function ... 77

1 Overview

Task

The CNC commands **M** and **H** are used to command machine switching and auxiliary functions via the PLC.

- Some of the M functions are defined by standardisation.
- Users can freely use H functions and all other functions.

Characteristics

By default, the PLC assigns M and H functions to the CNC channel in which they are programmed. In addition, these functions can be output to the PLC in an axis-specific range. This makes it simpler especially in systems with

- several CNC channels and
- together with the axes used by the functions

in order to implement PLC applications.

Parametrisation

CNC channels are each parameterised via an initialisation list (see Channel parameter list). All M and H functions must be specified here. Instead of a channel-specific assignment, an axis-specific assignment and a synchronisation method can be defined.

At the interface to the PLC (High Level Interface) [HLI], M and H functions are output in the channel and axis-specific areas and the PLC further processes them and confirms them accordingly.

A complete list of parameters described in this document is contained in the chapter Parameters [▶ 63].

Programming

M and H functions are programmed in compliance with DIN 66025. The M/H function is output at the channel-specific part of the HLI.

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 Description

Machine switching functions

The timing of the output of M and H functions to the PLC and its subsequent reaction can be synchronised in various ways with motions to be executed.

To simulate the machining time of an M or H function, a time period can be specified in order to determine the run-time of a CNC program.

Characteristics of M and H functions

- Possible synchronisation methods
- Freely available or default in compliance with DIN 66025
- Machining time can be determined
- Channel or axis-specific output
- Spindle-specific

The CNC channels are each parameterised via an initialisation list (see Channel parameter list). All M and H functions must be specified here. Instead of a channel-specific assignment, an axis-specific assignment and a synchronisation method can be defined.

At the interface to the PLC (High Level Interface) [HLI], M and H functions are output in the channel and axis-specific areas and the PLC further processes them and confirms them accordingly.

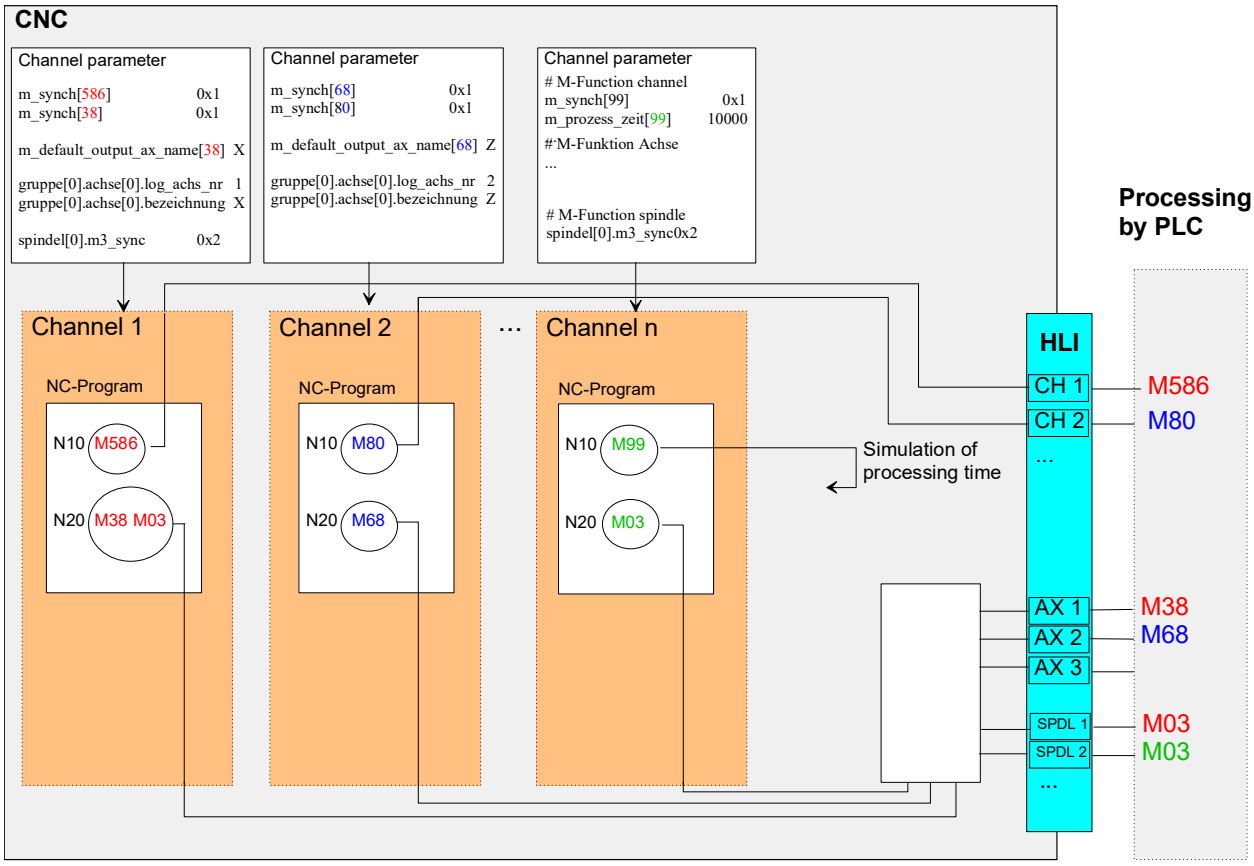


Fig. 1: M and H functions are enabled for each CNC channel in the channel parameter list via their synchronisation method.

For safety reasons, M and H functions cannot be used in the CNC program if a synchronisation method is not specified.

In this case, the CNC generates the message P-ERR-20157.

Output

The following options are provided for output to the PLC:

- No output
- Before / after a motion
- After an event

Output to the PLC (HLI) can take place in different ranges of the interface:

- In the channel range
- In the axis range

Synchronisation

The following options are available for synchronisation with a motion to be executed:

- No synchronisation
- Before / after a motion
- Synchronisation at the transition to the next motion block with machining feed rate ('late' synchronisation implicit)
- For an explicit request ('late' synchronisation explicit)
- During a motion block (example: edge banding)

2.1 Range of freely user-definable M/H numbers

The maximum number range from 0 to 65.535 is provided for M / H numbers.

The actually available range is limited depending on the version and can be found in the documentation [SYSP].

This is where you will also find details of the limits regarding the maximum number of M and H functions in each CNC block as well as the maximum number of simultaneously active 'late' synchronisations per channel and per axis.

The M functions reserved in compliance with DIN 66025 pose a further restriction (see the next Section: "Default M functions [▶ 14]").

2.2 Default M functions

Channel M functions

In compliance with DIN 66025, the following M functions are assigned fixed defaults as CNC commands. Unless specified accordingly in the channel parameter list, they are neither output nor synchronised.

M function	Meaning
M00	Programmed stop
M01	Optional stop
M02, M30	Main program end
M17, M29	Subroutine end

The parameter P-CHAN-00041 must be assigned to the channel- or axis-specific output.

Spindle M functions

As defined in DIN 66025, the following M functions are fixed defaults for spindle control. A synchronisation method is not specified as for other M functions but for each spindle under the special parameters in the channel parameter list [CHAN] (except M40 - M45).

M function	Meaning
M03	Endless clockwise spindle rotation
M04	Endless counter-clockwise spindle rotation
M05	Stop spindle
M19	Positioning spindle
M40 - M45	Exception: Gear speed selection for the main spindle

Default M functions

Spindle output

```
spindel[0].m3_synch        0x00020002   PLC_INFO, MVS_SVS
spindel[0].m4_synch        0x00020004   PLC_INFO, MVS_SNS
```

Channel output

```
m_synch[3] 0x00000002  Channel output
```

2.3 Parameterising the synchronisation methods

Synchronisation mode

The synchronisation mode of each corresponding M function is defined in the table entitled P-CHAN-00041 (m_synch).

The field index "MNr" corresponds to the M function number. The value specifies the synchronisation mode of the M function, i.e. when an output to the PLC and a check for the presence of the PLC acknowledgement take place.

A motion is not executed or is stopped at the latest towards the end of the block if no acknowledgement is received from the PLC.

The corresponding table P-CHAN-00027 (h_synch) is used for H functions.

Parameter

P-CHAN-00041	Synchronisation method of the M function with the number MNr.XX, e.g. M03
P-CHAN-00027	Synchronisation method of the H function with the number HNr.XX

The synchronisation mode of each M/H function can also be changed in the NC program by corresponding variables [PROG]:

V.G.M_FCT[MNr].SYNCH or

V.G.H_FCT[HNr].SYNCH

Synchronisation methods

Symbol	Value	Meaning
NO_SYNCH	0x00000000	No output of the M/H function to the PLC
M**O**S	0x00000001	Output of M/H function to PLC **without** synchronisation
M**V**S_S**V**S	0x00000002	Output of M/H function to PLC **before** motion block, synchronisation **before** motion block
M**V**S_S**N**S	0x00000004	Output of M/H function to PLC before motion block, synchronisation after motion block
M**N**S_S**N**S	0x00000008	Output of M/H function to PLC after motion block, synchronisation after motion block
M**N**E_S**N**S	0x00000020	Output of M/H function to PLC **after** measurement event and removal of distance to go, synchronisation **after** motion block (for edge banding option only)
BWD_SYNCH	0x00400000	Synchronisation of M function during **backward** motion with MVS_SVS
FWD_SYNCH	0x00800000	Synchronisation of M function during 'Simulated forward motion' with corresponding synchronisation mode
FAW_SYNCH	0x10000000	Decoder stop (**F**lush **a**nd **W**ait): 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.
Cross-block synchronisation (CAUTION: only allowed for M functions.)		
M**V**S_**SLM**	0x00004000	Late synchronisation, M function output at block start. Synchronisation at transition to motion block with machining feed rate (G01/G02/G03) ('late' synchronisation implicit, **S**ync. **l**ate **m**ovement)
M**V**S_**SLP**	0x00008000	Late synchronisation, M function output at block start. Synchronisation with NC command #EXPL SYN ('late' synchronisation explicit, **S**ync. **l**ate **p**rogram)

Synchronisation output brought forward		
ME**P**_S**V**S	0x01000000	M function output to PLC with specified **distance**, synchronisation **before** next block
ME**T**_S**V**S	0x02000000	M/H function output to PLC with specified **distance**, synchronisation **before** next block
M**O**S_TS	0x00040000	**CAUTION: Only allowed for M functions.** Output of M function to PLC before motion block **without** synchronisation, time offset as parameter for high-precision time output in PLC (see detailed description of MOS_TS in Section "Pre-output of M functions")
ME**P**_M**O**S	0x00100000	Output of M function with specified **path**, **without** synchronisation. M function must be fetched from PLC!
ME**T**_M**O**S	0x00200000	Output of M function with specified **time**, **without** synchronisation. M function must be fetched from PLC!

M or H output **Synchronisation**	**Before NC BLOCK**	**After NC block**	**After event**	**Brought forward according to specified distance/time**	**none**
Before motion	MVS_SVS			MEP_SVS MET_SVS	
After motion	MVS_SNS	MNS_SNS	MNE_SNS		
Implicit at next transition to G01/G02/ G03	MVS_SLM				
Programmed synchronisation	MVS_SLP				
none	MOS MOS_TS			MEP_MOS MET_MOS	NO_SYNCH

The reaction of the synchronisation modes MVS_SVS, MVS_SNS and MNS_SNS is identical if M or H functions are programmed without a motion in the block.

Defining M/H functions and parameterising the synchronisation methods

```
# ******************************************************
# Definition of M functions and synchronisation types
# ======================================================
m_synch[0]                          0x00000002     MVS_SVS
m_synch[1]                          0x00000001     MOS
m_synch[2]                          0x00000002     MVS_SVS
m_synch[8]                          0x00000008     MNS_SNS
m_synch[9]                          0x00000000     NO_SYNCH
#
```

```
# ******************************************************
# Definition of H functions and synchronisation types
# ======================================================
h_synch[0]                          0x00000001     MOS
h_synch[1]                          0x00000002     MVS_SVS
h_synch[2]                          0x00000004     MVS_SNS
h_synch[3]                          0x00000008     MNS_SNS
h_synch[4]                          0x00000002     MVS_SVS
#
```

2.3.1 Examples of the synchronisation methods in combination with motions

To simplify matters, DIN syntax with channel-specific output is used in the following examples. Parameterisation takes place with the known parameter P-CHAN-00041 (m_synch[..]).

2.3.2 Example with MVS_SVS

Motion is not enabled until the M function is acknowledged by the PLC.

Initialisation in the channel parameter list

```
m_synch[..]    0x2
```

MVS_SVS

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

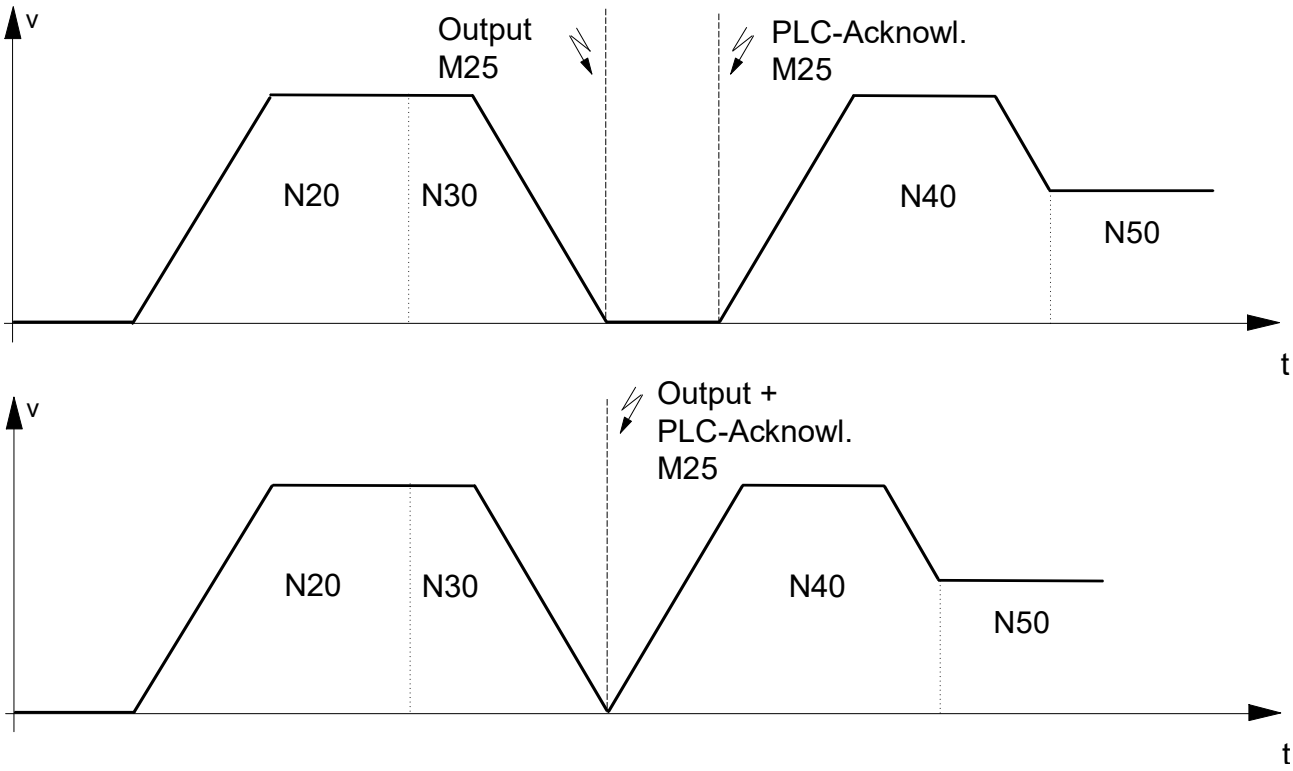


Fig. 2: Synchronisation type MVS_SVS

When block N40 is executed, M25 is output and the program waits for the PLC acknowledgement before start of the motion in N40.

2.3.3 Example with MVS_SNS

The following CNC block is not executed until after the PLC acknowledges the M function.

Initialisation in the channel parameter list

```
m_synch[..]     0x4
```

MVS_SNS

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

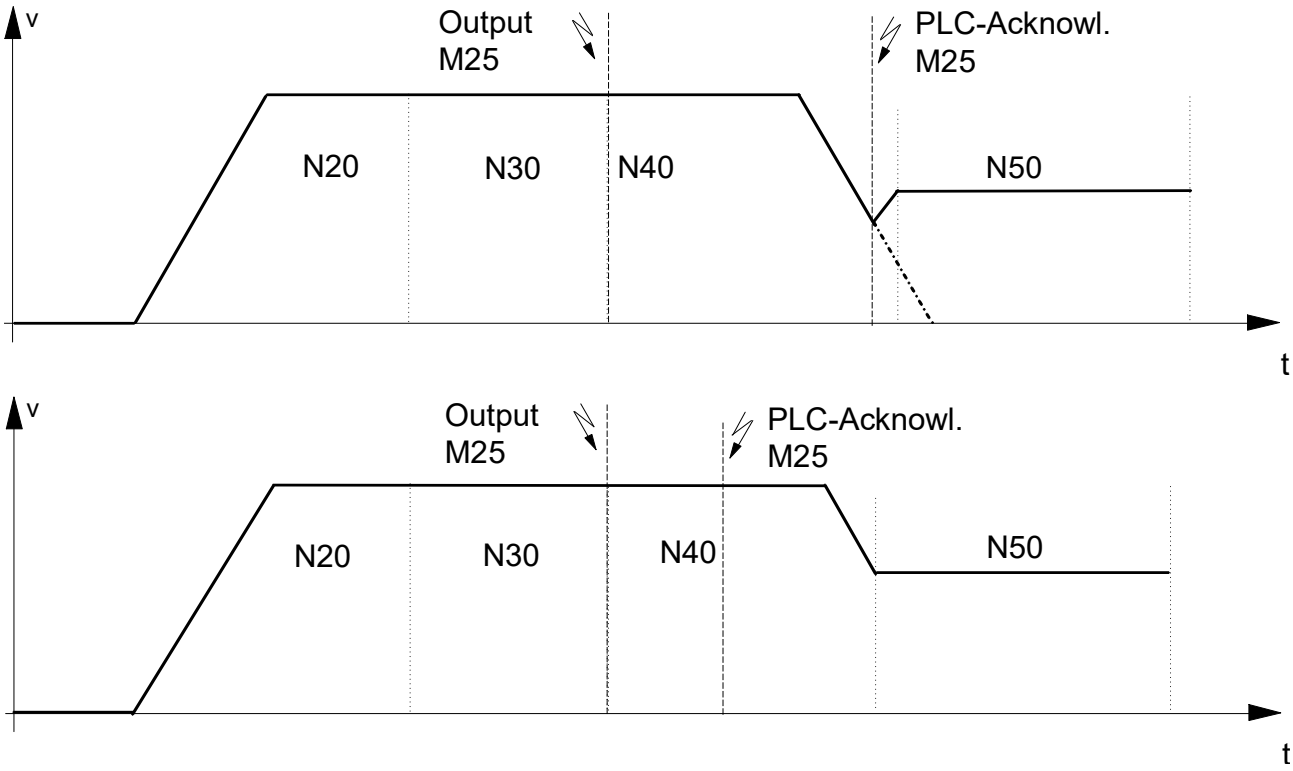


Fig. 3: Synchronisation type MVS_SNS

When the N40 block is executed, M25 is output and the motion is continued. If the acknowledgement of M25 is not received in good time, the program stops at the end of N40.

2.3.4 Example with MNS_SNS

Stopping at block end until the M function is acknowledged by the PLC.

Initialisation in the channel parameter list

```
m_synch[..]    0x8
```

MNS_SNS

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

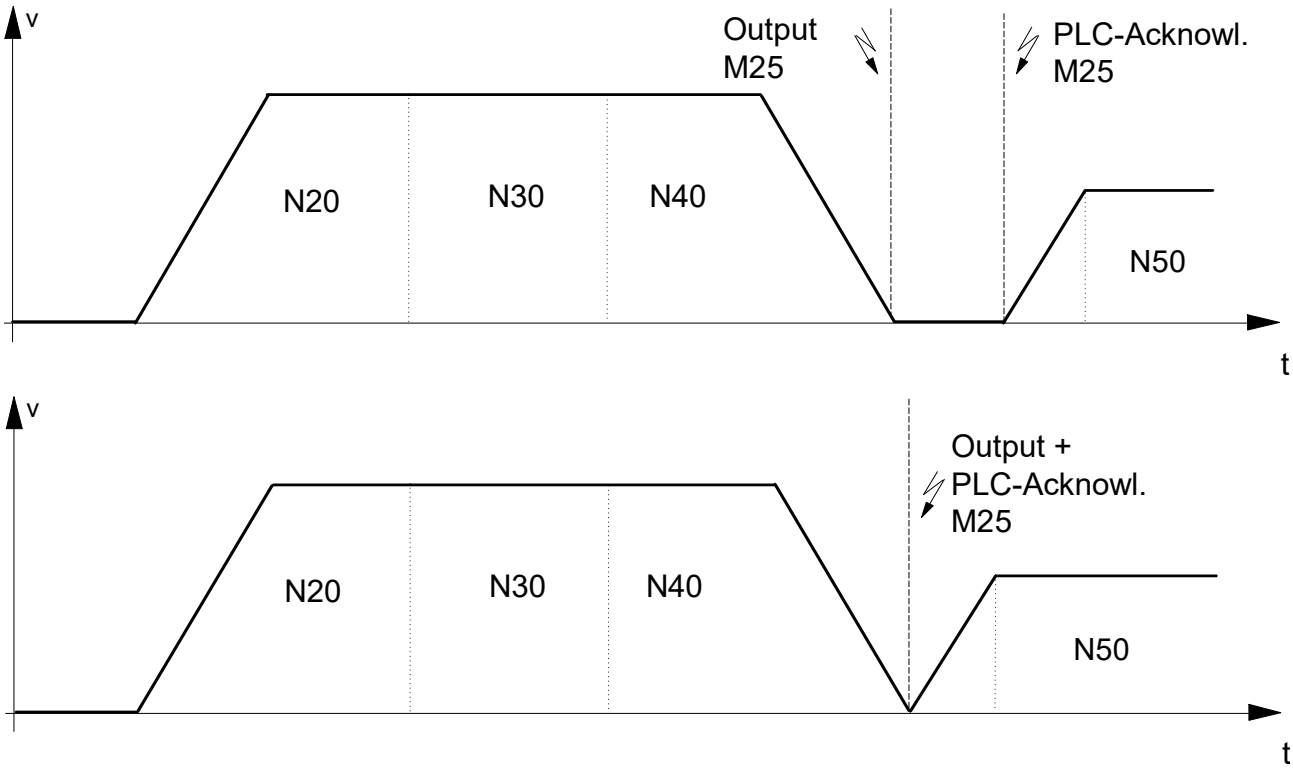


Fig. 4: Synchronisation type MNS_SNS

After the motion in N40, the program stops in any case and waits for the acknowledgement by the PLC after output of M25.

2.3.5 Example with MNE_SNS

The M function is output by a measurement event and takes place after removal of the specified distance to go. Motion is not enabled beyond the end of the block until after acknowledgement of the M function by the PLC.

Initialisation in the channel parameter list

```
m_synch[..]   0x20
```

MNE_SNS

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

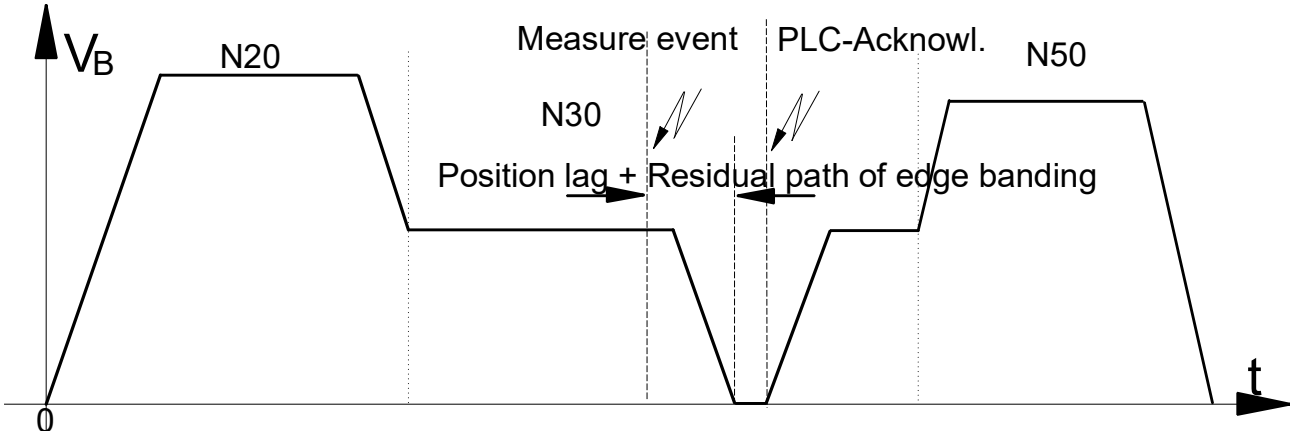


Fig. 5: Synchronisation method MNE_SNS

M33 is output takes place after the measurement event in the N30 block and after removal of the distance to go defined by the measuring method. The program then waits for the PLC to acknowledge M33.

2.3.6 Example of MVS_SLM

Cross-block implicit synchronisation at the transition to the next feed motion block (G01, G02, G03). Motion is not enabled after the end of this motion block until the PLC acknowledges the M function.

Initialisation in the channel parameter list

```
m_synch[..]    0x4000
```

MVS_SLM

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

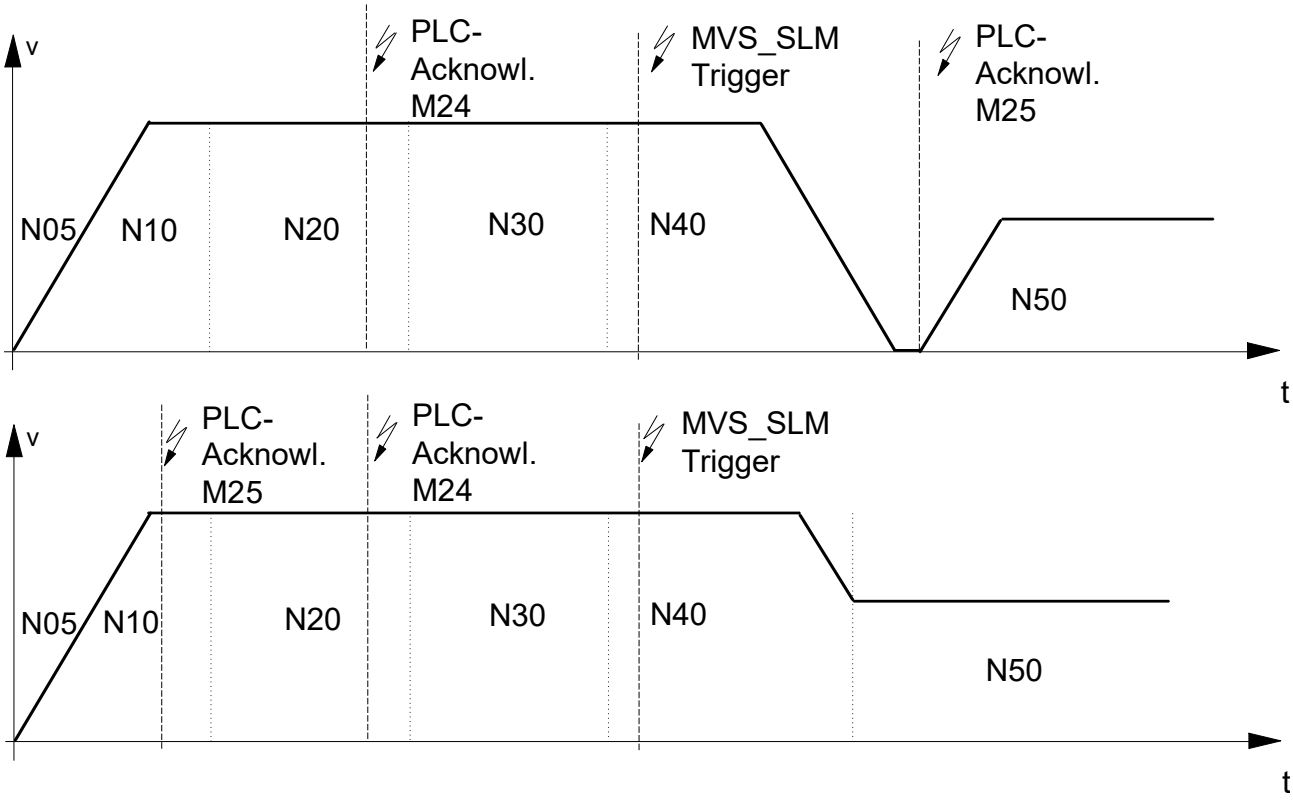


Fig. 6: Synchronisation type MVS_SLM

If the MVS_SLM M function is programmed in the feed motion block, synchronisation takes place before this motion starts in conformity with MVS_SVS.

Example: N10 **G01** F100 **M24**

2.3.7 Example of MVS_SLP

Programmed synchronisation (#EXPL SYN)

Initialisation in the channel parameter list

```
m_synch[..]    0x8000
```

MVS_SLP

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

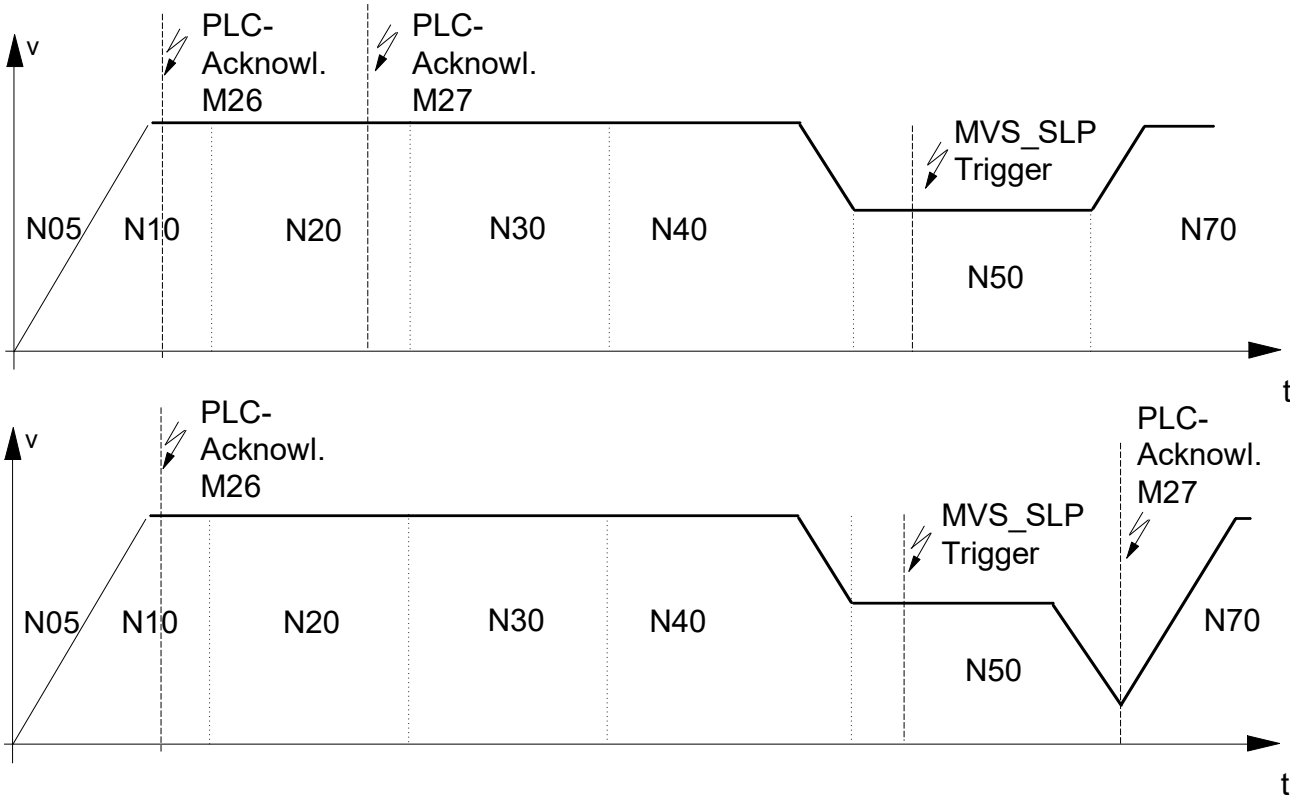


Fig. 7: Synchronisation type MVS_SLP

2.3.8 Response of 'late' synchronisation (MVS_SLM, MVS_SLP) at program end

If there is no feed motion block after programming

- an MVS_SLM M function
- or no #EXPL SYN left in the current CNC program after programming an M function with synchronisation method MVS_SLP,

the M function remains active after the CNC program ends.

Synchronisation then takes place in a subsequent CNC program when the trigger condition is reached.

“Late synchronisation”

```
%PRG1
N05 M26 G00 X25       (M26, Synchronisationstyp MVS_SLM)
N10 M27               (M27, synchronisation type MVS_SLP)
M30                   (M26, M27 still active at program end)
                      (enabled)

%PRG2
N10 G01 F100 X10     <--triggering M26 before
                        execution of the motion
N20 #EXPL SYN           Triggering M27 before
                        execution of the next block
M30
```

If, when a CNC program is started, there are still pending M functions from the previous CNC program with a 'late' synchronisation method, synchronisation can always be forced here.

To enable this, the channel parameter P-CHAN-00033 (late_sync_ready) is assigned the value 1.

Initialisation in the channel parameter list

```
late_sync_ready          1
```

2.3.9 Example with synchronisation output brought forward MEP_SVS and MET_SVS

With these M/H functions the CNC calculates the required output time across the default path parameter (MEP_SVS) or time parameter (MET_SVS). In this case, look-ahead profile planning takes place internally. The basic calculation model can be modified using P-CHAN-00209.

These synchronisation methods determine the pre-output of the M function in accordance with the path or time specified. Motion is not enabled until the M function is acknowledged by the PLC.

The parameter P-CHAN-00212 is used to activate the provision of the remaining time/distance to go up to the synchronisation point.

The parameter P-CHAN-00274 is used to increment the number of blocks for the look-ahead (output position preview).

Initialisation in the channel parameter list:

`m_synch[..]` 0x01000000 (synchronisation MEP_SVS)

`m_synch[..]` 0x02000000 (synchronisation MET_SVS)

Synchronisation output brought forward

```
N10 G01 X10 G90 F5000
N20     X20
N30     X30
N40     X40
N50 M96              (M96 MEP_SVS m_pre_outp = 250000,)
                     (oder MET_SVS m_pre_outp = 300000us)
N55 X80
N60 X0
M30
```

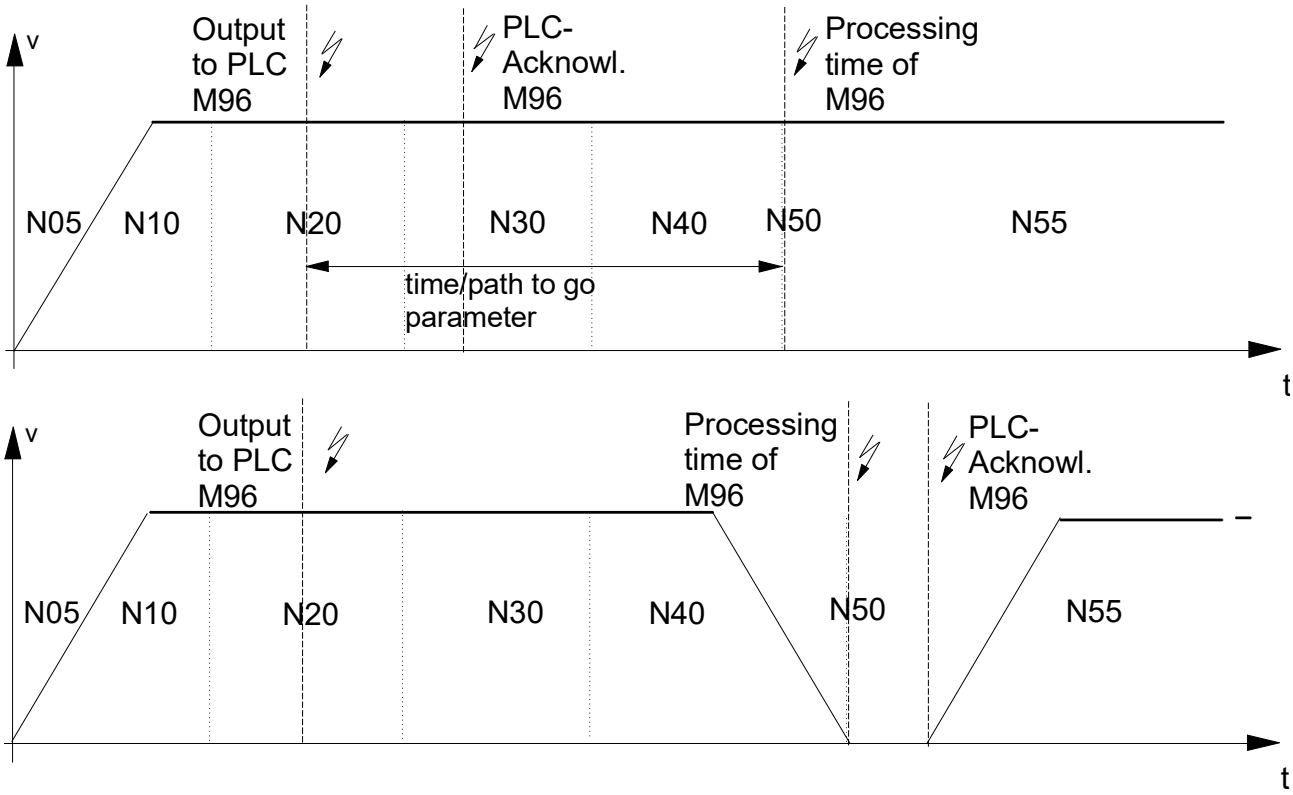


Fig. 8: Synchronisation types MET_SVS, MEP_SVS

Operating principle of P-CHAN-00212:

Activates the provision of distance to go/remaining time up to the synchronisation point for access via ADS

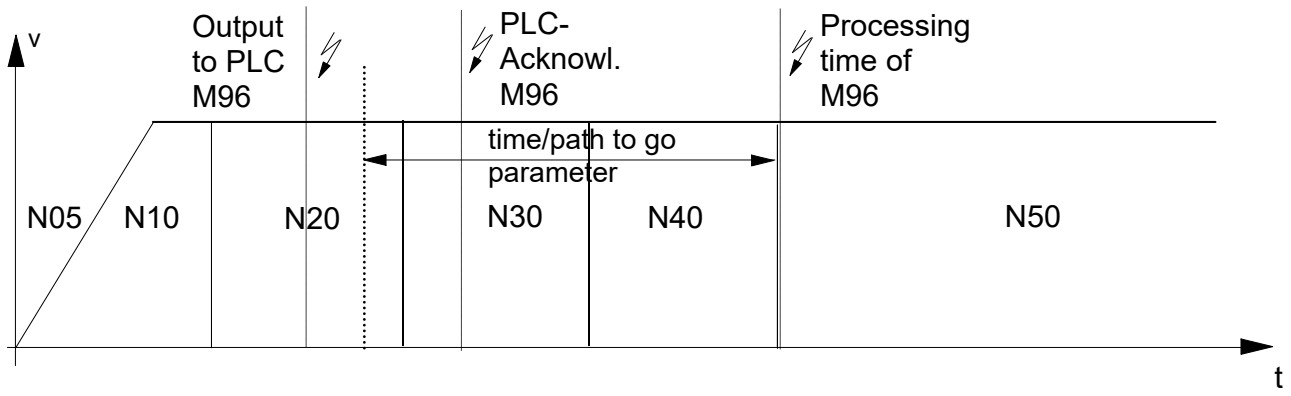


Fig. 9: Activate residual path/time calculation with M/H code look ahead

2.3.10 Defining the lead distance or time lead

The channel parameters P-CHAN-00070 (m_pre_outp[i]) and P-CHAN-00107 (h_pre_outp[i]) are used in conjunction with the synchronisation methods **MET_SVS** and **MEP_SVS**. The parameters define the value for output before execution of the M/H functions in the path interpolator.

- The time lead is specified with the MET_SVS method.
- The lead distance is specified with the MEP_SVS method.

Example for initialisation in the channel parameter list for 2 M functions:

- The user-specific M function M96 is to be output to the PLC 10 millimetres before the synchronisation position is reached in the block sequence.
- The user-specific M function M97 is to be output to the PLC 40 millimetres before the synchronisation time in the block sequence is reached.

```
# Definition of M functions and synchronisation types
# ===================================================
:
m_synch[96]                        0x01000000     MEP_SVS
m_synch[97]                        0x02000000     MET_SVS
#
# Pre-output time/distance setting with MET_SVS, MEP_SVS
# ===================================================
m_pre_outp[96]                       100000     in 0.1 µm
m_pre_outp[97]                       40000      in µs
```

The lead distance/time lead of an M/H function can also be modified in the NC program by corresponding variables (e.g. V.G.M_FCT[MNo].PRE_OUTP_PATH, see [PROG]).

NOTICE

Please note for MET_SVS codes: The pre-output time is planned due to the required synchronisation with path movements based on a feed profile with end velocity 0.

As a result, there may be deviations between the planned and the actual motion times up to the M code (block limit). See the figure below:

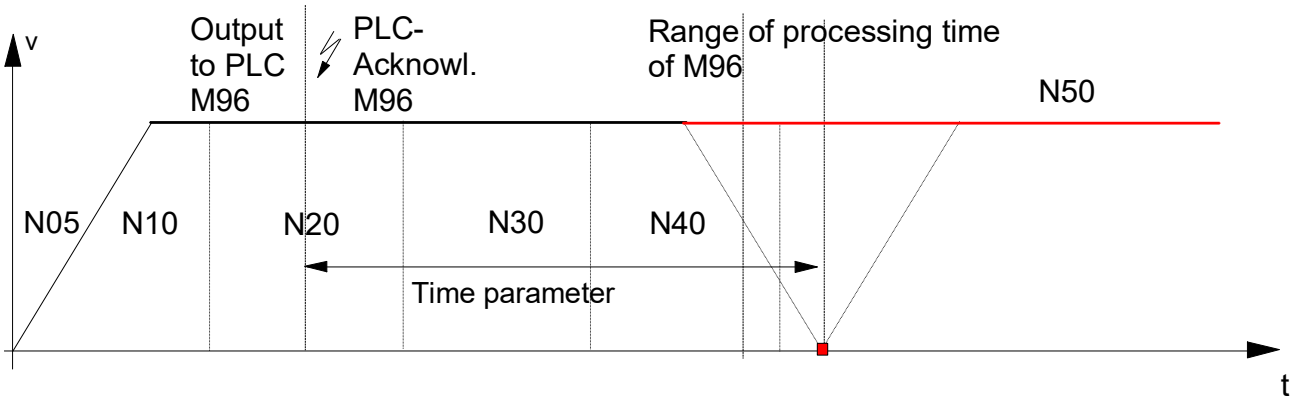


Fig. 10: Planned output time for MET_SVS

2.3.11 Example with MEP_MOS and MET_MOS

With these M/H functions the CNC calculates the required output time across the default path parameter or time parameter. In this case, look-ahead profile planning takes place internally. The basic calculation model can be modified using P-CHAN-00209.

The parameter P-CHAN-00274 is used to increment the number of blocks for the look-ahead (output position preview).

Pre-output of the M function in accordance with the specified path or time. Without synchronisation, M/H functions must be read by the PLC like all MOS.

Initialising in the channel parameter list

```
# Definition of M functions and types of synchronisation
# ======================================================
:
m_synch[96] 0x00100000 MEP_MOS
m_synch[97] 0x00200000 MET_MOS
#
```

```
# Pre-output time/distance setting with MET_MOS, MEP_MOS
# ======================================================
m_pre_outp[96] 100000 in 0.1 um
m_pre_outp[97] 40000 in us
```

MEP_MOS and MET_MOS

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

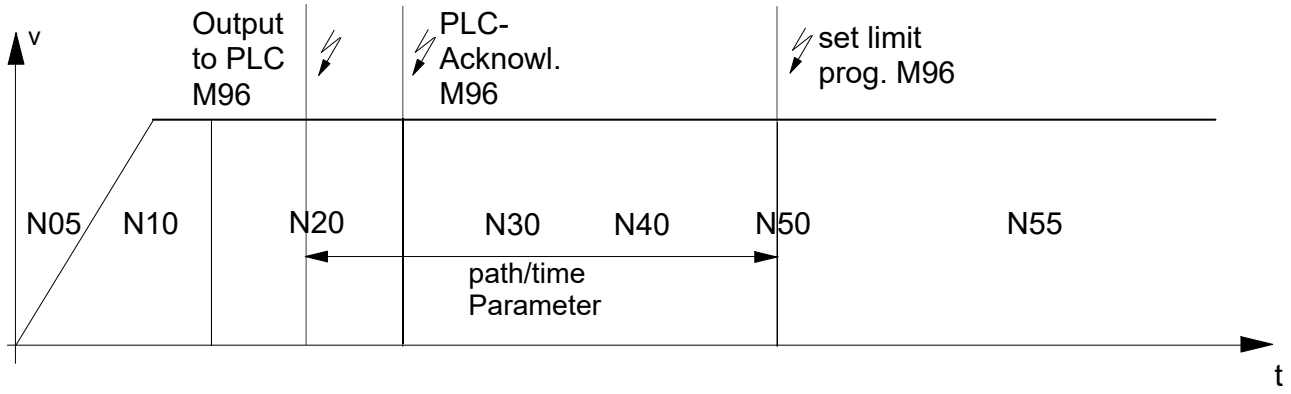


Fig. 11: Synchronisation types MET_MOS and MEP_MOS

2.3.12 Example with MOS_TS

Output of the M function before the block, no synchronisation. With this method the CNC provides a time offset referred to the block transition point.

Due to sampling at cycle time t_{cycl}, the output point of an M function at interpolator level lies within the time of the sampling rate but this is offset in time to the block transition point by maximum one sampling cycle.

The exact output time can be calculated and executed in the PLC by using the tracked time offset of the M function.

See also [HLI], Section “Data of the M function/H function“ for CNC versions up to Build V2.11.2800 or “Data of the M function/H function“ for CNC versions as of Build V2.11.2800.

Initialisation in the channel parameter list:

`m_synch[..]` 0x00040000 (MOS_TS)

MOS_TS

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

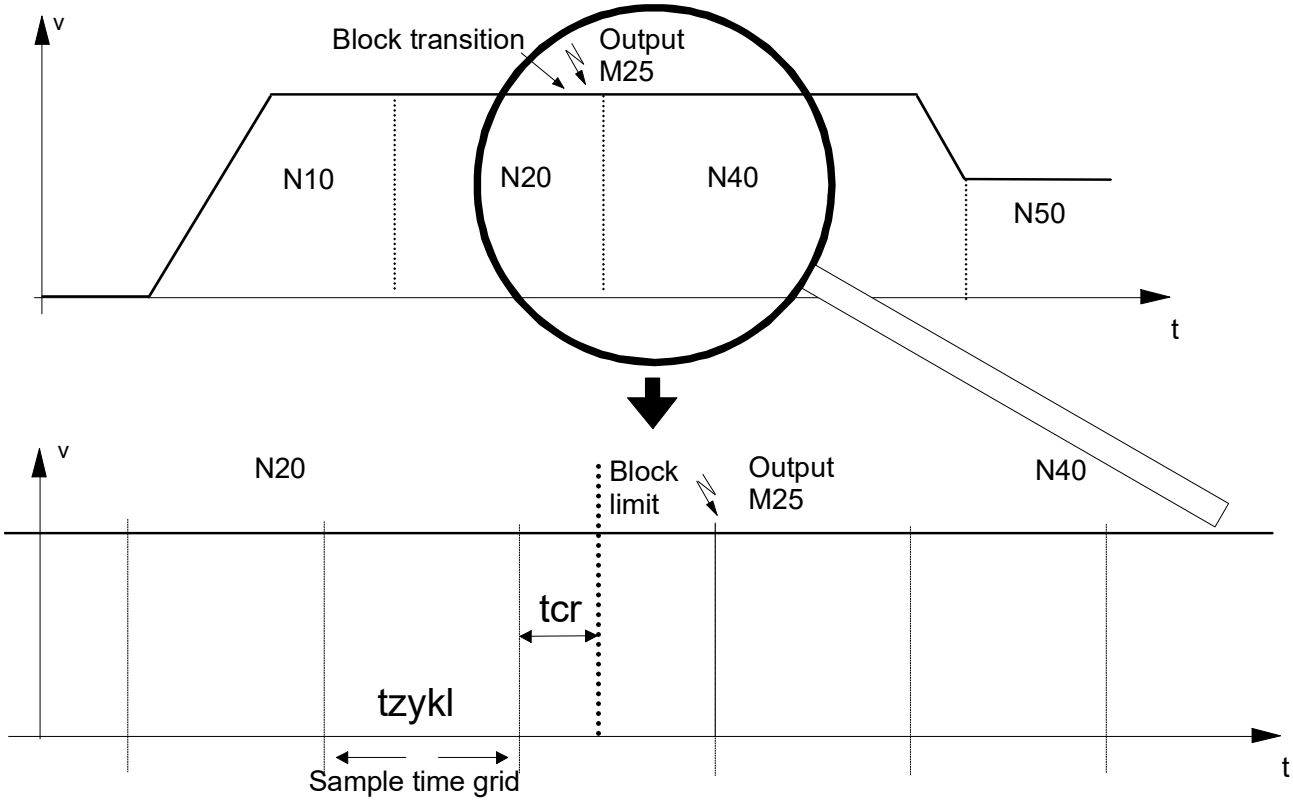


Fig. 12: Synchronisation method MOS_TS

M function output

The M function is programmed at the block transition between N20 and N40. Due to the time-based sampling the sample point and therefore the output of the M function lags slightly behind the transition of the block limit.

- As a result, the M function is output with a maximum delay of one sampling interval.
- The offset of the programmed theoretical output time of the M function up to the last sampling cycle is output as deceleration value t_{cr}.

NOTICE

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

The function for the exact output instant is typically executed in the PLC by using a special high-resolution timer hardware.

NOTICE

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

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

Exact path-synchronous triggering of I/Os

During a cutting operation, a laser (100 µm beam width) is to be switched on/off at the exact position. The tolerance is within 10 µs or within ½ beam width (=50 µm). The time-related resolution of the CNC interpolation cycle (typically 1 ms) is not sufficient for this. The problem can be solved by using high-resolution timer hardware and algorithms in the PLC.

The dead time in the CNC to drive system chain must be greater than the dead time in the system chain to the laser system.

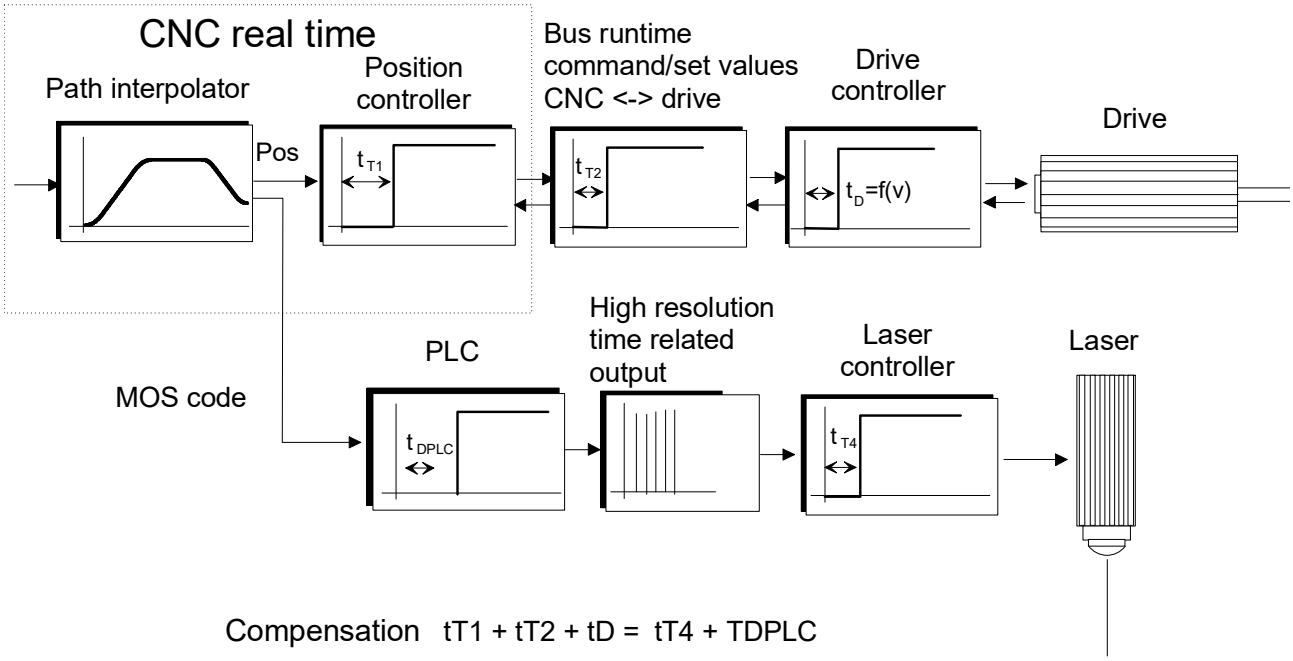


Fig. 13: Overview of dead times in system chains

This case is based on a laser system with a typical dead time of 800 µs. With the CNC - drive positioning chain, the typical dead time is 5 sample cycles ((5 x 1 ms = 5 ms).

The positioning control system in the drive operates with no position lag (active feedforward control). However if necessary, position lag can be estimated.

The PLC takes into consideration the dead time of the positioning chain to control the laser with time T_{DPLC}. As a result, the PLC delays the M code of the path interpolator supplied by the CNC by n cycles. The exact switching signal for the laser in the succeeding cycle is calculated in the PLC based on the interpolation data and the dead time and it is generated with a high basic clock-pulse rate (e.g. 1 µs) via a hardware terminal.

```
.....
N10 G01 G90 X100 F10000
N15 M5
N20              X101
N30.....
```

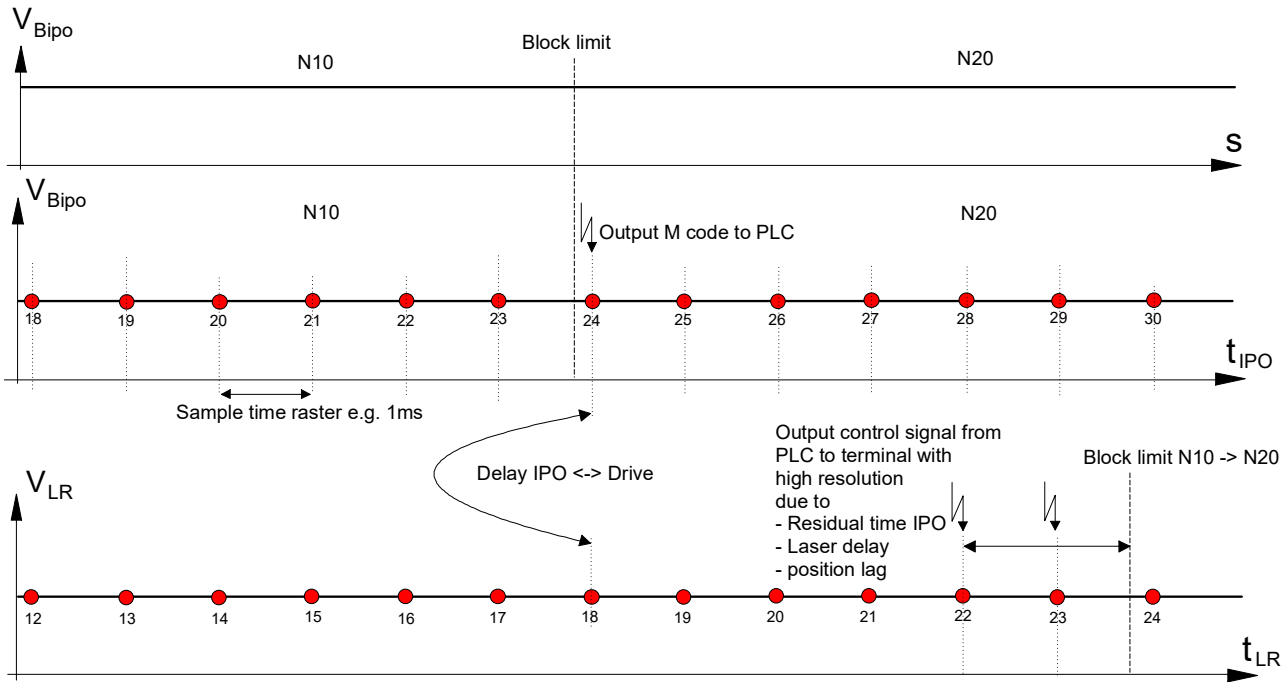


Fig. 14: Diagram of dead time compensation

3 Channel and axis-/-specific M/H functions

3.1 General programming

Channel-specific

Programming is done in the usual fashion in conformity with DIN66025. The M/H function is output at the channel-specific part of the HLI.

For all M/H functions to be used, the synchronisation entry must be under the parameters P-CHAN-00041 (m_synch[mnr]) and P-CHAN-00027 (h_synch[mnr]) in the channel parameter list. This does not apply to default M functions.

Channel-specific programming

```
N20 X75
N30 X50
N40 X25 M8
N50 X0 H1
```

Axis-specific (programming)

An extended CNC syntax is used to output an M or H function to an axis.

If the output is always to take place at a specific axis, one option is to dispense with extended syntax and assign the M/H function to this axis by default.

The chosen axis must be in the channel. This is done by assigning the output axis configuration in the channel parameter list accordingly or by programming axis exchange commands during execution in the CNC program [PROG].

Axis-specific (parameterisation)

Configuration for default output at the axis-specific part of the HLI when programming with DIN 66025 syntax.

```
m_default_outp_ax_name[7]          Z
```

Axis-specific programming

```
N20 X75
N30 X50
N40 X25 M7      Output of M7 to the Z axis
N50 X[M7]       Output of M7 to the X axis
```

Explicit (extended syntax)

Programming in axis-specific syntax [PROG] for output to the axis-specific part of the HLI. In the case of explicit programming, an assignment from the channel parameter list is ignored.

Programming axis-specific syntax

```
N20 X75
N30 X50
N40 X25 X[MM7]        Output of MM7 to the X axis
N50 X0
```

Mixed programming

Output of M or H functions to several axes and the channel range in the same NC block.

Mixed programming

```
N10 X[M7] S[M7] M7    Simultaneous output of M7 to the
                      X axis, the main spindle S and the
                      channel range
N20 X[H1] S[H1] H1    Corresponding example for the H function
```

3.2 M/H functions with additional information

Programming

With M/H functions, an optional additional value can be programmed directly in the NC program as a negative or positive integer or as a general mathematical expression. This value is supplied to the PLC via the technology interface together with the M/H function.

Example: M50 **= 123** or H44 **= -256**

The internal M functions M00, M01, M02, M17, M29 and M30 and all user-specific M/H functions can be programmed with an additional value both in channel-specific and in axis-specific programming syntax.

Restrictions

The spindle M functions M03, M04, M05, M19 and M40-M45 and M6 (if P-CHAN-00118 is set) may only be programmed **without** additional value.

M/H functions with additional information

```
#M functions with additional value
N10 M52=-345
N20 M12=123 (with channel parameter m_default_outp_ax_name[12] Z)
N30 M10=321 (with channel parameter m_default_outp_ax_name[10] S)
N35 P1=567     P2=345
N40 X[M54=P1]
N50 S[REV 1000 M03 M63=-789]
N60 M12=123    M10=321 M52=-345 X[M54=567] S[REV 1000 M03 M63=-789]
N70 M63=-789   M52=-P2 M54=567
N80 X[M52=-345 M54=567] Y[M63=-789] S[M05 M63=789 M54=-567] M54 M63

#H functions with additional value
N110 H5=-345
N120 H6=123 (with channel parameter h_default_outp_ax_name[6] Z)
```

```
N130 H9=321 (with channel parameter h_default_outp_ax_name[9] S)
N135 P3=567    P4=-345
N140 X[H7=P3]
N150 S[REV 1500 M04 H8=-789]
N160 H6=123    H9=321 H5=-345 X[H7=567] S[REV 1500 M04 H8=-789]
N170 H8=-789   H5=P4  H7=567
N180 X[H8=-789 H4 H5=-345] Y[H7=567] S[M05 H5=345 H7=567] H3 H8

#Mixed M/H functions with additional value
N200 X[M52=-345 H4 H8=-789 M54=567 H5=345] H3=333 M54=444 H7=567 M63

#M/H functions with additional value in axis-specific function (INDP)
N05 X[INDP G90 G01 FEED=2000 POS=555 M54=151 H8=-181]

N999 M30=111
```

3.3 Production time calculation

When planning production processes, it is important to know what production times are required for the various jobs. This also applies to the time that a machine needs to produce a part.

To simulate production time, the run times in µs that the user determined empirically must be specified in the table P-CHAN-00040 or P-CHAN-00026 (***_prozess_zeit[i]**) in the channel parameter list.

The field index "i" specifies the number of the M/H function [CHAN].

Parameter

P-CHAN-00040	Execution time for the M function with the number i
P-CHAN-00026	Execution time for the H function with the number i

Production time calculation 1

A time of 0 or 1.3 s are required to execute the M07 and M08 functions:

```
m_prozess_zeit[7]               500000
m_prozess_zeit[8]               1300000
```

With pre-assigned spindle M functions (and also for the S function), the execution time is specified in the spindle parameters [CHAN].

Parameter

P-CHAN-00042	Execution time for the spindle M function M19
P-CHAN-00044	Execution time for the spindle M function M3
P-CHAN-00046	Execution time for the spindle M function M4
P-CHAN-00048	Execution time for the spindle M function M5
P-CHAN-00080	Execution time for the spindle S function

Production time calculation 2

A time of 0.5 or 1.5 s is required to execute the M03 and M19 functions:

```
spindel[0].m3_prozess_zeit      500000
spindel[0].m19_prozess_zeit     1500000
```

3.4 Pre-output of M functions (MicroJoint)

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

Requirements to use the MicroJoints function:

The pre-output function must be enabled in each channel in P-CHAN-00600 [▶ 75], or in the start-up list in P-STUP-00060.

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

The MicroJoint function only supports the pre-output of channel-specific M functions; axis-specific M functions are not supported.

With the MicroJoint function, M functions can be pre-output by specifying a path. No timed pre-output is possible. This can only be implemented with M functions of synchronisation type MET_SVS.

When the MicroJoints function is inactive, a path-based pre-output of M/H functions is only possible with the synchronisation type MEP_SVS.

Activating and enabling the function

Pre-output of an M/H function is executed if:

a pre-output path is specified in P-CHAN-00070 or P-CHAN-00107

Path-related pre-output of M functions

A pre-output can automatically output an M function in advance at a specific point along the path.

For example, in the case of M functions with a time stamp MOS_TS, this can be used for advanced deactivation of a laser to briefly interrupt the cutting process. This leaves so-called MicroJoints.

Output of the advanced M function is not tied to the originally programmed block limits. The motion block is opened automatically by the CNC at the corresponding positions and the M function is inserted.

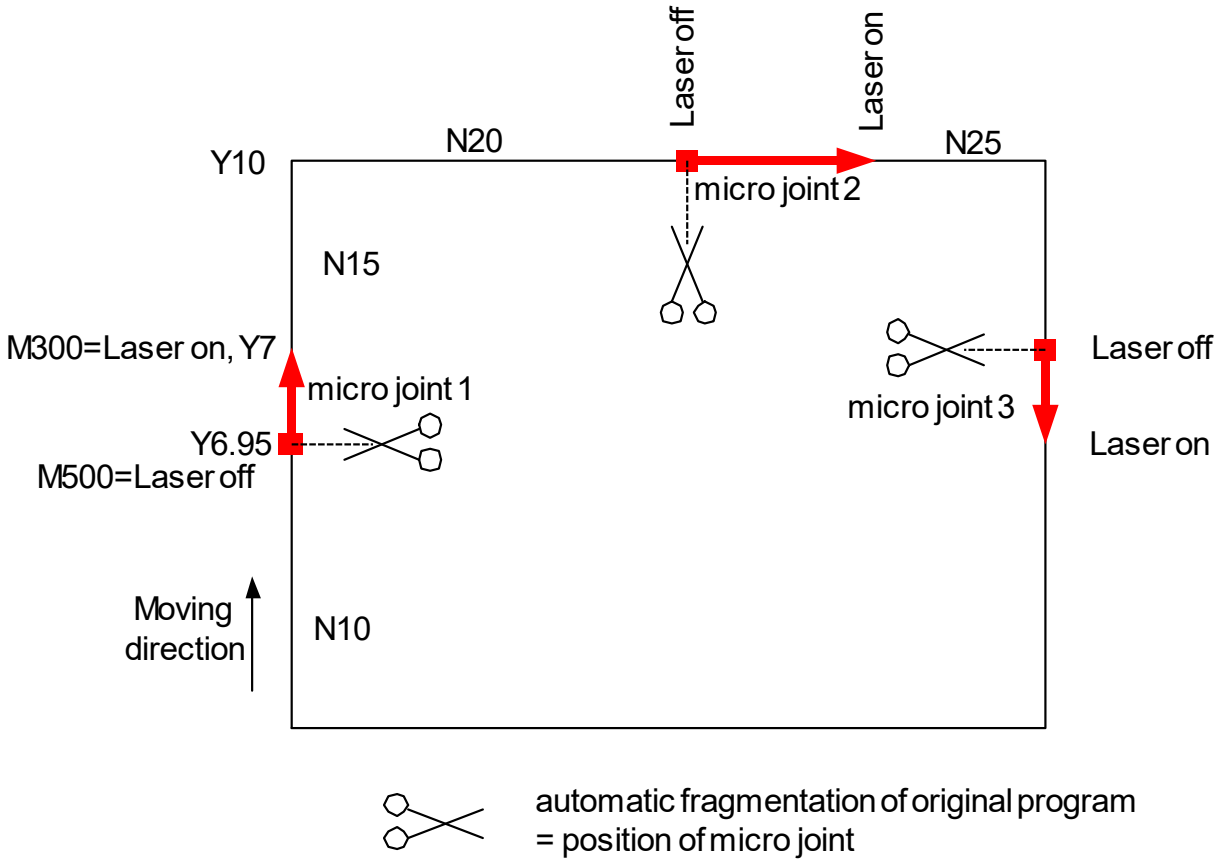


Fig. 15: Programmed MicroJoints in the part

Pre-output of M functions

```
; M300 - Laser on, M500 - Laser off
N05 V.G.M_FCT[500].PRE_OUTP_PATH = 0.05
N10 G00 G90 X0 Y0
N15 L Laser_on.sub
N20 G01 F5000
N25      Y7
N30 M500 M300           ;Microjoint 1
```

```
N35      Y10
N40 X14
N45 M500 M300           ;Microjoint 2
N50 X20
N55 L Laser_off.sub
N99 M30
```

Equivalent example with explicit programming

```
; M300 - Laser on, M500 - Laser off
N05 G00 G90 X0 Y0
N10 L Laser_on.sub
N15 G01 F5000
N20      Y6.95
N25 M500                  ;Microjoint 1
N30      Y7
N35 M300
N40      Y10
N45 X13.95
N50 M500                  ;Microjoint 2
N55 X14
N60 M300
N65 X20
…
N70 L Laser_off.sub
N99 M30
```

M/H functions for pre-output

In addition to the actual use of pre-output with high-resolution MOS_TS, output is basically also possible for other M or H functions.

The following synchronisation methods of the M and H functions are evaluated at pre-output:

MOS, MVS_SVS, MVS_SNS, MNS_SNS, MOS_TS

reference position of the pre-output

If the M/H function is programmed together with a motion, then:

the path of the pre-output is determined for its output time relative to the block.

MOS, MOS_TS, MVS_SVS and MVS_SNS are determined relative to the block start position

MNS_SNS is positioned relative to the block end position.

Due to pre-output, however, it is basically no longer required to separate the output and synchronisation points.

In other word, if the M/H function is synchronised (MVS_SVS, MVS_SNS, MNS_SNS), output and synchronisation take place at the same point. This corresponds to programming the M/H function in a separate NC line.

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

Parametrisation using lists

M functions with pre-output are parameterised in the channel list by P-CHAN-00041 (m_synch[..]) and P-CHAN-00070 (m_pre_outp[..]):

```
m_synch[100]     MOS_TS
m_pre_outp[100]  500                    ;in 0.1 µm
```

H functions with pre-output are parameterised in the channel list by P-CHAN-00027 (h_synch[..]) and P-CHAN-00107 (h_pre_outp[..]):

```
h_synch[50]      MVS_SVS
h_pre_outp[50]   400                    ;in 0.1 µm
```

Parametrisation by programming

As a alternative to the parametrisation of M/H functions, the synchronisation method and the path also can be specified directly in NC program.

```
V.G.M_FCT[11].SYNCH = 1
V.G.M_FCT[11].PRE_OUTP_PATH = 14         ;in [mm]

V.G.H_FCT[200].SYNCH = 4
V.G.H_FCT[200].PRE_OUTP_PATH = 40        ;in [mm]
```

Synchronisation types as macro

```
%MicroJoint
; Synchronisation types as macro
"MOS" = "1"
"MVS_SVS" = "2"
"MVS_SNS" = "4"
"MNS_SNS" = "8"
"MOS_TS" = "262144" ;0x40000

V.G.M_FCT[11].SYNCH = "MOS_TS"
V.G.M_FCT[11].PRE_OUTP_PATH = 11   ;in [mm]
V.G.M_FCT[13].SYNCH = "MNS_SNS"
V.G.M_FCT[13].PRE_OUTP_PATH = 23   ;in [mm]

V.G.H_FCT[12].PRE_OUTP_PATH = 12   ;in [mm]
V.G.H_FCT[12].SYNCH = "MVS_SVS"

N01 X0 G01 F500
N10 X100
N20 X200    M11 H12 M13
N30 X300
M30
```

3.4.1 Limitations, special cases

Limitation of the look-ahead range

The described look-ahead range is limited due to resource limitation and the requirement for the NC program to run up as soon as possible after start.

The default limits the maximum number of blocks considered for pre-output to 10 blocks; this can be set in P-CHAN-00603 (or in P-STUP-00061). Depending on the block length, this results in a maximum joint width.

If an M/H function is brought forward beyond the look-ahead range, the M/H function is only brought forward automatically as far as the look-ahead range permits.

Limitation of the look-ahead range

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

```
N03 V.G.M_FCT[100].PRE_OUTP_PATH = 28.6 ;in mm
N20 G91 Y1
N21 Y1  ; -> planned MicroJoint at Y1.4 mm
N22 Y1
N23 Y1
N24 Y1
N25 Y1
…
N37 Y1
N38 Y1
N39 Y1   ; -> real MicroJoint caused by block number limitation
N40 Y1
N41 Y1
N42 Y1
N43 Y1
N44 Y1
N45 Y1
N46 Y1
N47 Y1
N48 Y1
N49 Y1
N50 M100 M26
N99 M30
```

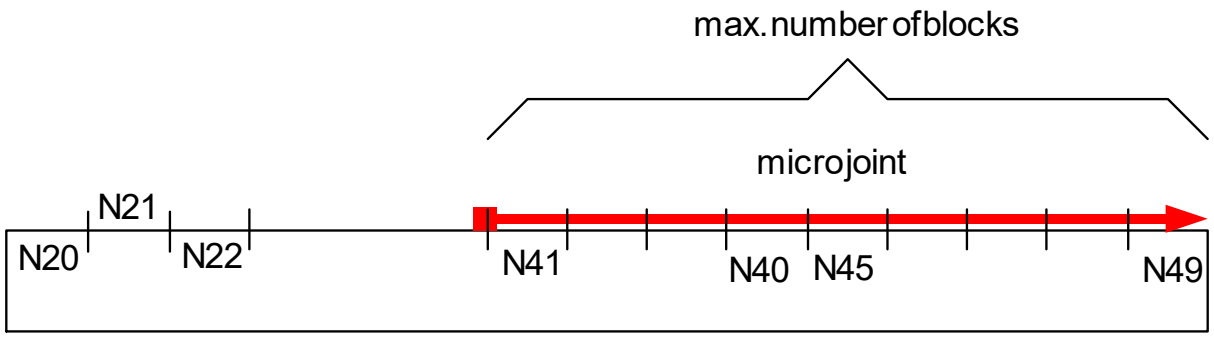


Fig. 16: Limitation of the pre-output path of the M function to 10 blocks

Explicit cancellation of the look-ahead range, #FLUSH, Read synchronous V.E. variable

The look-ahead range of M functions is reset by flushing the channel (#FLUSH or #FLUSH WAIT). In other words, the pre-output of M functions cannot be reversed beyond the #FLUSH point.

An implicit #FLUSH WAIT, i.e. the channel is flushed, can also be executed when a synchronous V.E variable (see [EXTV]) is read. A pre-output via the read access of a synchronous V.E. variable is therefore not possible either.

If an M/H function is brought forward beyond the look-ahead range, the M/H function is only brought forward automatically as far as the look-ahead range permits.

Explicit cancellation of the look-ahead range, #FLUSH, Read synchronous V.E. variable

```
%microjoint6
N01 G00 G90 X0 Y0
N02 G01 F10000

N10 V.G.M_FCT[100].PRE_OUTP_PATH = 28.6 ; in mm
N20 G91 Y1
N21 Y1 ; -> planned MicroJoint at Y1.4 mm
N22 Y1
N23 Y1
…
N38 Y1
N39 Y1
N40 Y1
N41 Y1
N42 Y1
N43 Y1
N44 Y1
```

```
N400#FLUSH   ; -> MicroJoint inserted at Y24
N45 Y1
N46 Y1
N47 Y1
N48 Y1
N49 Y1
N50 M100 M26
N99 M30
```

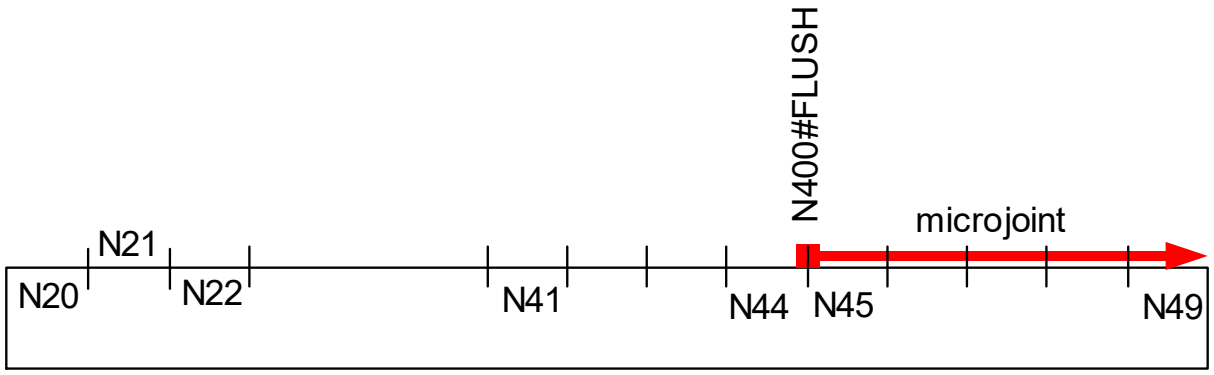


Fig. 17: Explicit limitation of the lead distance of the M function

"Overlapping" path-related pre-output

After a path-related pre-output of the M function is detected, all previously stored motion blocks are output. This corresponds to explicit flushing of the channel (see #FLUSH), thus avoiding delayed processing of the motion blocks.

As a result, it is not possible to overlap the path range of several M functions.

If pre-outputs of M/H functions overlap, the individual M/H functions are only brought forward automatically up to the originally programmed position.

"Overlapping" path-related pre-output

```
%microjoint5
(* M100 - Laser off, M26 - Laser on *)
N01 G00 G90 X0 Y0
N02 G01 F10000
N03 V.G.M_FCT[101].PRE_OUTP_PATH = 5   ;in mm
N04 V.G.M_FCT[102].PRE_OUTP_PATH = 23
N05 V.G.M_FCT[103].PRE_OUTP_PATH = 31
N20 X10
N30 M101 M26
N40 X30
N50 M102 M26
N60 X40
N70 M103 M26
N80 M30
```

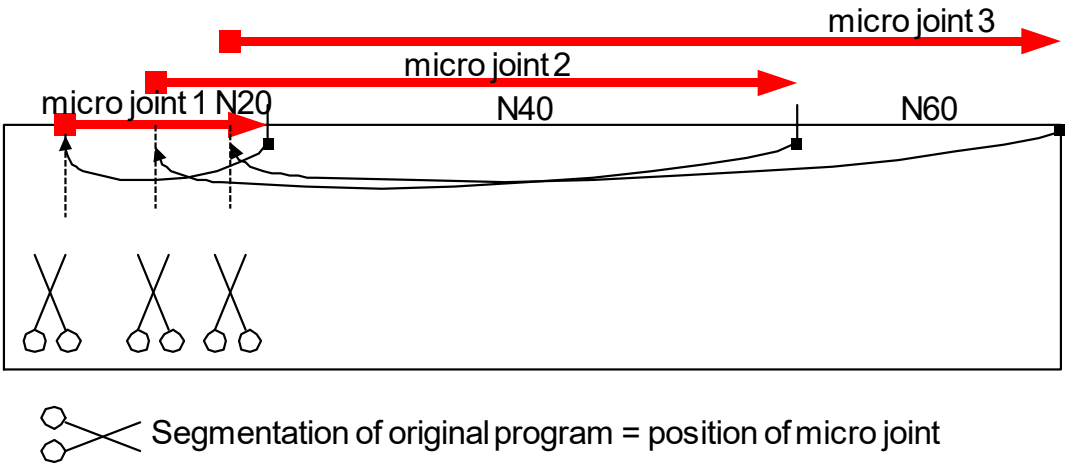


Fig. 18: Theoretical overlapping of MicroJoints in the part

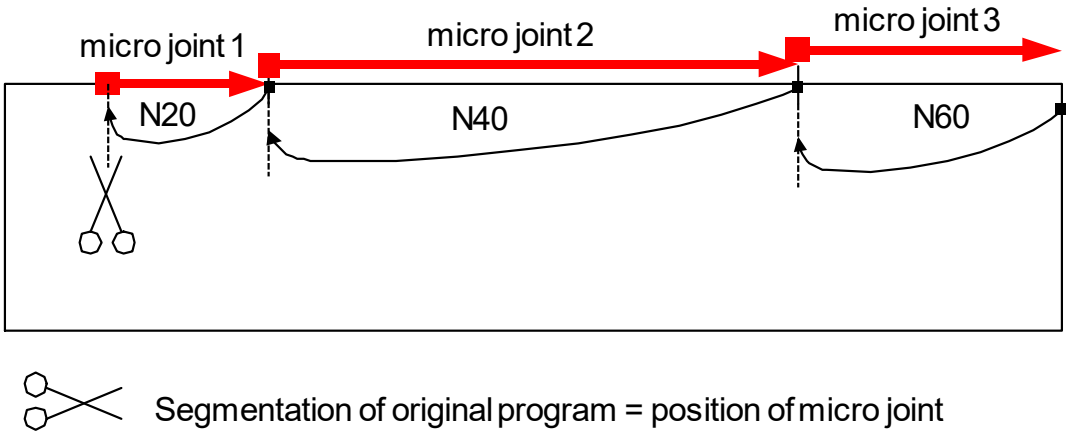


Fig. 19: Limitation of overlapping of MicroJoints in the part

Combination of MNS_SNS with and without pre-output path

It is not permitted to execute the simultaneous programming of MNS_SNS M functions with and without pre-output path in the same NC block including an axis motion.

Combination of MNS_SNS with and without pre-output path

```
%microjoint9
N01 G01 G90 X0 Y0 F10000

N02 V.G.M_FCT[100].PRE_OUTP_PATH = 35.6; in mm

N04 V.G.M_FCT[100].SYNCH = 8 ;MNS_SNS
N04 V.G.M_FCT[200].SYNCH = 8 ;MNS_SNS

N20 X10
N40 X30 M100 M200
N60 X40

N99 M30
```

Pre-output between output and synchronisation points of an MVS_SNS

It is not possible to insert an M/H function of synchronisation type MVS_SNS between output and synchronisation points. The pre-output of the M/H function is then only brought forward up to synchronisation point of the MCS_SNS at the most.

Pre-output of another M/H function of the MVS_SNS type

The example below shows an attemptyto move M100 between the output and synchronisation points of the M200 (an MVS_SNS with motion). In this case, M100 is not output as required at X9 but at the end of N10, i.e. at X10.

```
N01 G01 G90 X0 Y0 F1000
(Definitions of M functions)
N02 V.G.M_FCT[100].PRE_OUTP_PATH = 21.0
N03 V.G.M_FCT[100].SYNCH = 2  ; MVS_SVS
N04 V.G.M_FCT[200].SYNCH = 4  ; MVS_SNS

N10 X10 M200
N20 X30
N30 X40 M100
N40 X0 Y0
N50 M30
```

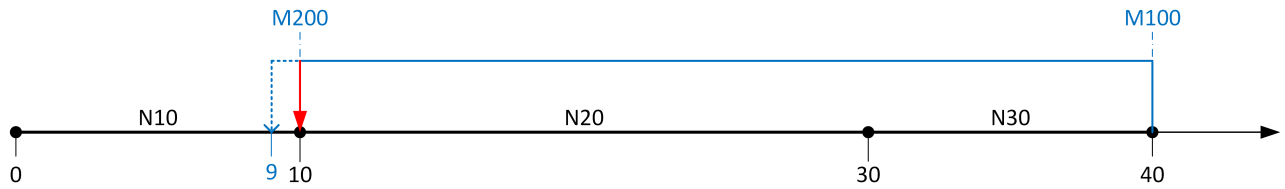


Fig. 20: Attempted pre-output with another M/H function of the MVS_SNS type

3.4.2 Tolerance parameters for the permitted deviation between real and planned pre-output positions of an M/H function

The Limitations and special cases [▶ 39] described in the section can result in an M/H function with pre-output not being output at the required position, but possibly at a significantly less advanced position.

However, the channel parameter P-CHAN-00760 (pre_output_tolerance) is used to check how far the pre-output of an M/H function may deviate from the planned position.

If the deviation exceeds the specified tolerance value, the program is aborted and a corresponding error message is output. Otherwise, the M/H function is automatically without a warning output and offset by this deviation from the expected position.

3.4.3 Explicit feed programming for MicroJoints (#CHANNEL SET)

Feed at / behind a MicroJoint

For technical process reasons, it may be necessary to limit the path velocity for a MicroJoint (in particular with an M function MOS which requires no acknowledgement). In addition, the path after the advanced M function (MicroJoint path) is completely traversed to the end at a change in velocity.

This can be defined by the following feed settings in the NC command #CHANNEL SET (see figure below “Feed definition with MicroJoints”).

#CHANNEL SET [M_PRE_OUTPUT [**E=..**] [**F=..**] [**VECTOR_LIMIT_OFF**] **]**		non-modal
E=..	Block end velocity E of the previous MicroJoint (start of MicroJoint)	
F=..	Feed velocity within the MicroJoint (path between the position of the advanced M function and the originally programmed position of the M function)	
VECTOR_LIMIT_OFF	Deselecting a possible dynamic limitation. If one of the previously programmed dynamic influences is active via #VECTOR LIMIT (VEL, ACC, DEC), it is suppressed within the MicroJoint range.	

Explicit feed programming for MicroJoints

```
%microjoint16
N01 G00 G90 X0 Y0
N02 G01 F100

N05 #CHANNEL SET [M_PRE_OUTPUT E=20 F=5000]

N10 V.G.M_FCT[100].PRE_OUTP_PATH = 8; in mm
N20 G91 Y1
…
N40 Y10
N50 M100 M26
N99 M30
```

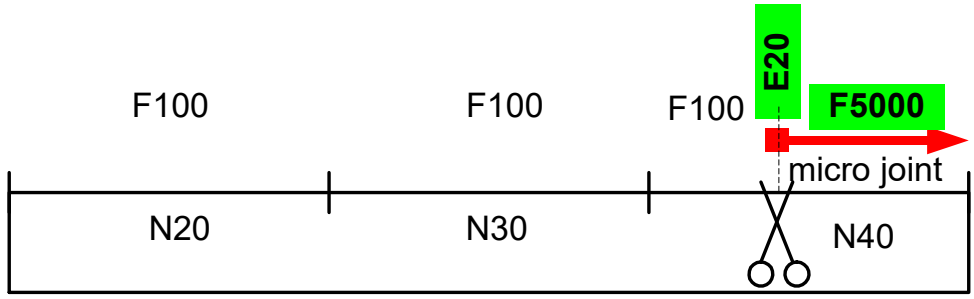


Fig. 21: Feed definition with MicroJoints

If the F or E word is not specified, the feed for the advanced M function and subsequent motion blocks is not changed.

MicroJoint feed across multiple blocks

When the pre-output of the M function is advanced across multiple blocks, the feed of all MicroJoint motion blocks is also changed to the specified value.

A possibly explicitly programmed feed is replaced by the specific MicroJoint feed.

MicroJoint feed across multiple blocks

```
%microjoint17
```

```
N01 G01 G90 X0 Y0 F100

N05 #CHANNEL SET [M_PRE_OUTPUT E=20 F=5000]
N10 V.G.M_FCT[100].PRE_OUTP_PATH = 15; in mm
…
N40 G91 Y10 F7500
N50 M100 M26
N99 M30
```

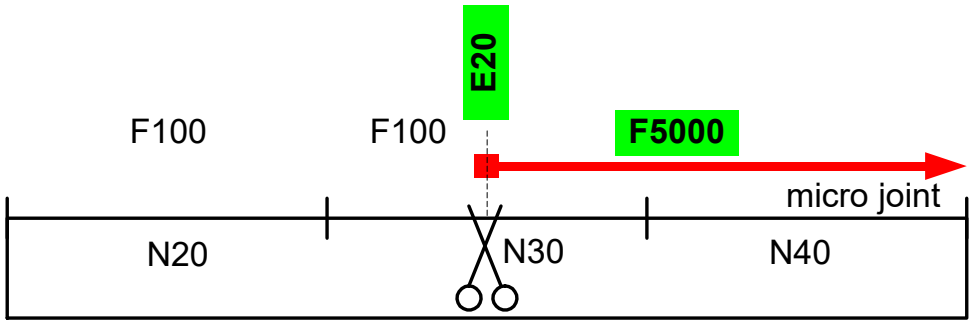


Fig. 22: Feed with block global MicroJoints

Specifying a MicroJoint specific feed replaces the possible explicitly programmed feed of the NC block.

See example above:

F7500 in N40 is replaced by F5000.

Increased feed with M11, decreased feed with M12

```
%microjoint16
V.G.M_FCT[11].SYNCH = "MOS"
V.G.M_FCT[11].PRE_OUTP_PATH = 125
V.G.M_FCT[12].SYNCH = "MOS"
V.G.M_FCT[12].PRE_OUTP_PATH = 325


N300 #CHANNEL SET [M_PRE_OUTPUT E=250 F=1500]
N01 X-222 G01 F1000

N10         X10
N20         X100
N30         X200 M11 (125mm)

N32 #VECTOR LIMIT ON[VEL=500]

N35 #CHANNEL SET [M_PRE_OUTPUT E=150 F=750]

N40         X300
N41         X310
N42         X320
N43         X330
N44         X340
N45         X350
N46         X360
N47         X370
N48         X380
N49         X390
N50         X500
N60         M12 (325mm)
N70         X600
N80         X700
M30
```

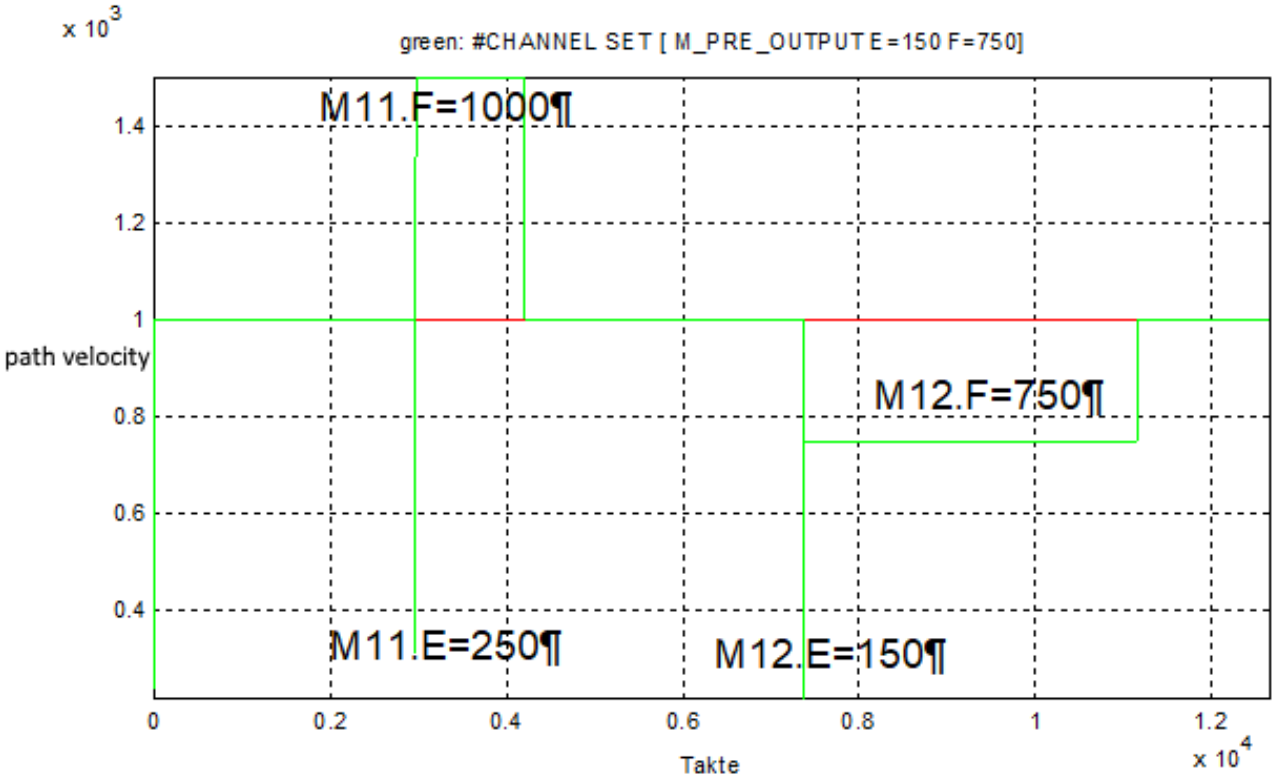


Fig. 23: Increased feed with M11, decreased feed with M12

Suppress VECTOR_LIMIT during MicroJoint

```
%microjoint18
V.G.M_FCT[11].SYNCH = "MOS"
V.G.M_FCT[11].PRE_OUTP_PATH = 125
V.G.M_FCT[12].SYNCH = "MOS"
V.G.M_FCT[12].PRE_OUTP_PATH = 325


N300 #CHANNEL SET [M_PRE_OUTPUT E=250 F=1500]
N01 X-222 G01 F1000

N10         X10
N20         X100
N30         X200 M11 (125mm)

N32 #VECTOR LIMIT ON[VEL=500]

N35 #CHANNEL SET [M_PRE_OUTPUT E=150 F=750 VECTOR_LIMIT_OFF]

N40         X300
N41         X310
N42         X320
N43         X330
N44         X340
N45         X350
N46         X360
N47         X370
N48         X380
N49         X390
N50         X500
N60         M12 (325mm)
N70         X600
N80         X700
M30
```

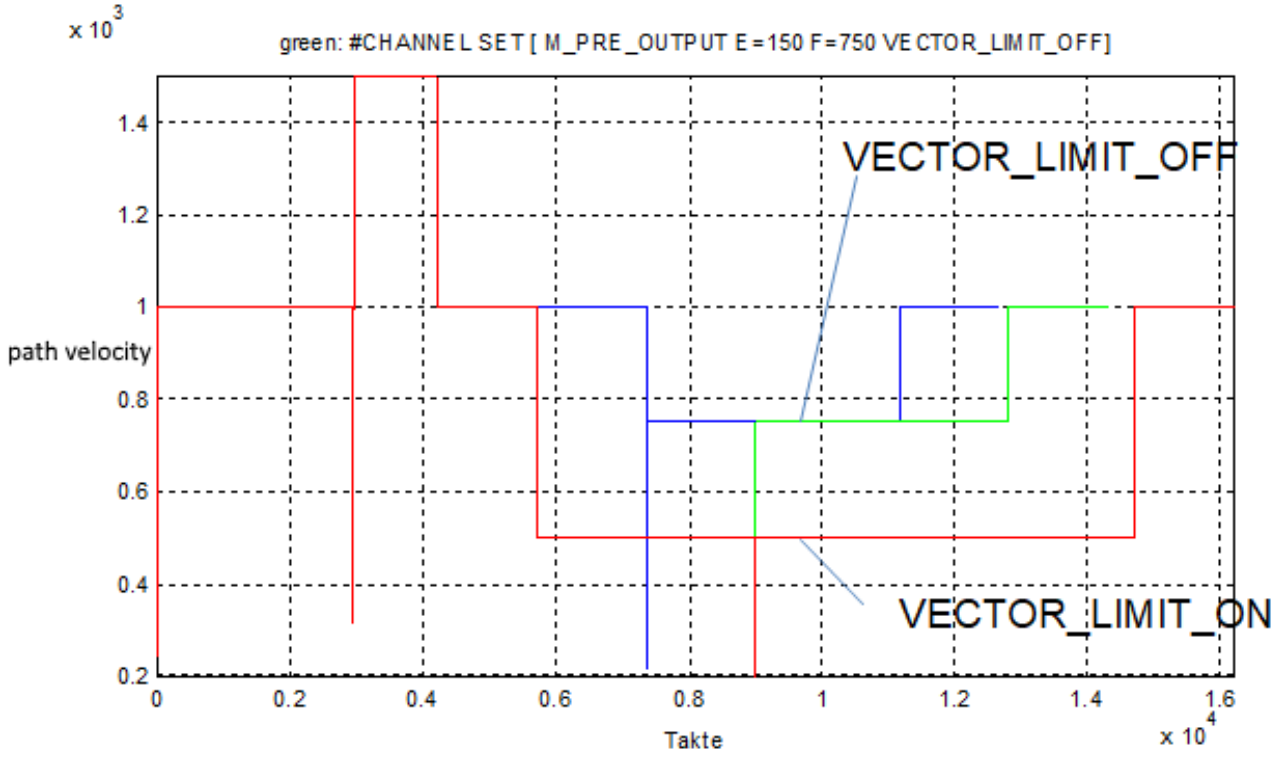


Fig. 24: Suppress VECTOR_LIMIT during MicroJoint

3.5 HMI display

Displays active M/H functions

Active M/H functions can be accessed via CNC objects, for example, and displayed on the user interface.

There is also an axis-specific view here in addition to a channel-specific view.

Active M/H functions, channel-specific	
Description	Displays currently active M/H functions, i.e. waiting for acknowledgement at the PLC interface.
Type	Structure: UNS32 Number UNS32 active M/H[12]
Value range	[0; MAX_UNS32]
HMI elements	mc_active_m_functions_r, mc_active_h_functions_r
Access	Read
$IndexOffset_{ADS}$	0x3d, 0x91 ($IndexGroup_{ADS}$ = 0x000201<ii> where <ii> = channel, [1; max])

Active M/H function, axis-specific	
Description	Displays the active axis-specific M/H functions, i.e. waiting for acknowledgement at the PLC interface.
Type	Structure: UNS32 Number UNS32 active M/H[12]
Value range	[0; MAX_UNS32]
HMI elements	ac_<I>_active_m_functions_r, ac_<I>_active_h_functions_r
Access	Read
$IndexOffset_{ADS}$	0x13, 0x17 ($IndexGroup_{ADS}$ = 0x000202<ii> where <ii> = axis index, [1; max])

AmsAdsViewer: Reads channel-specific H functions

After programming a manual block H41, H42, H51, these three H functions are active at the PLC interface since it is not yet acknowledged.

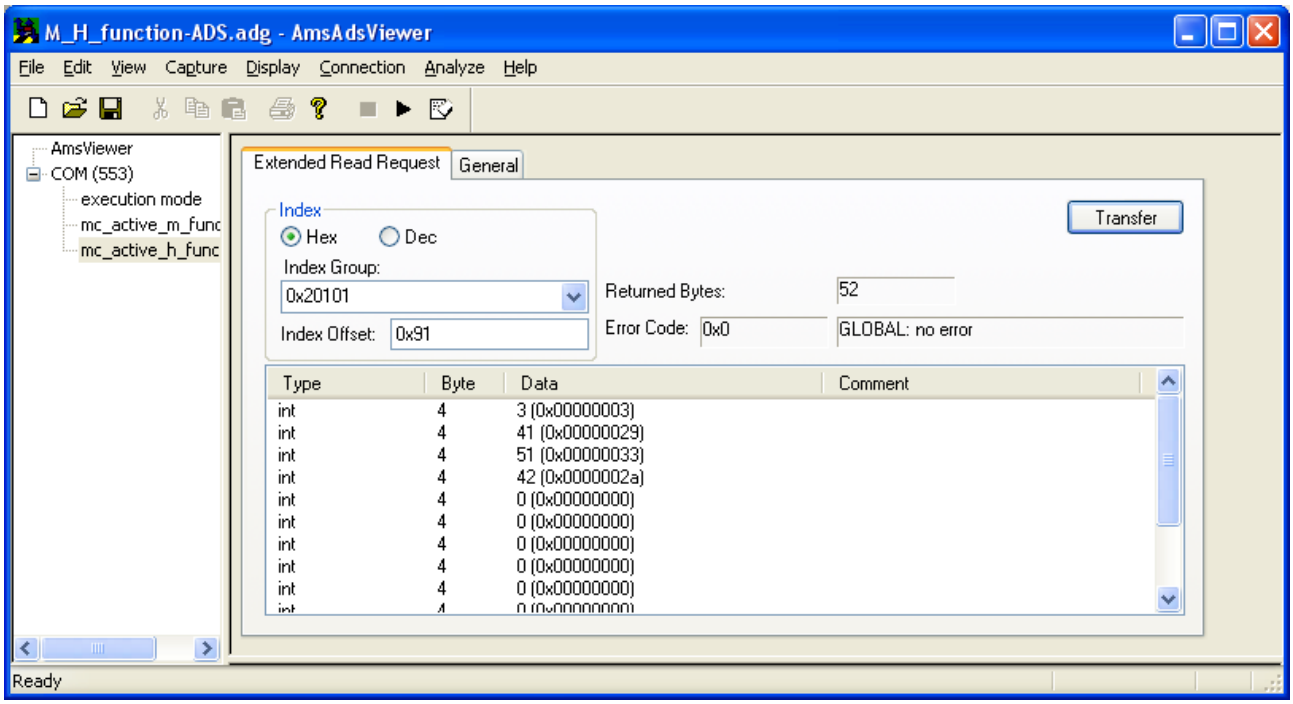


Fig. 25: AmsAdsViewer- Reads channel-specific H functions

AmsAdsViewer: Reading axis-specific H functions

After programming a manual block X[H50], the axis-specific H function is active at the PLC interface since it is not yet acknowledged.

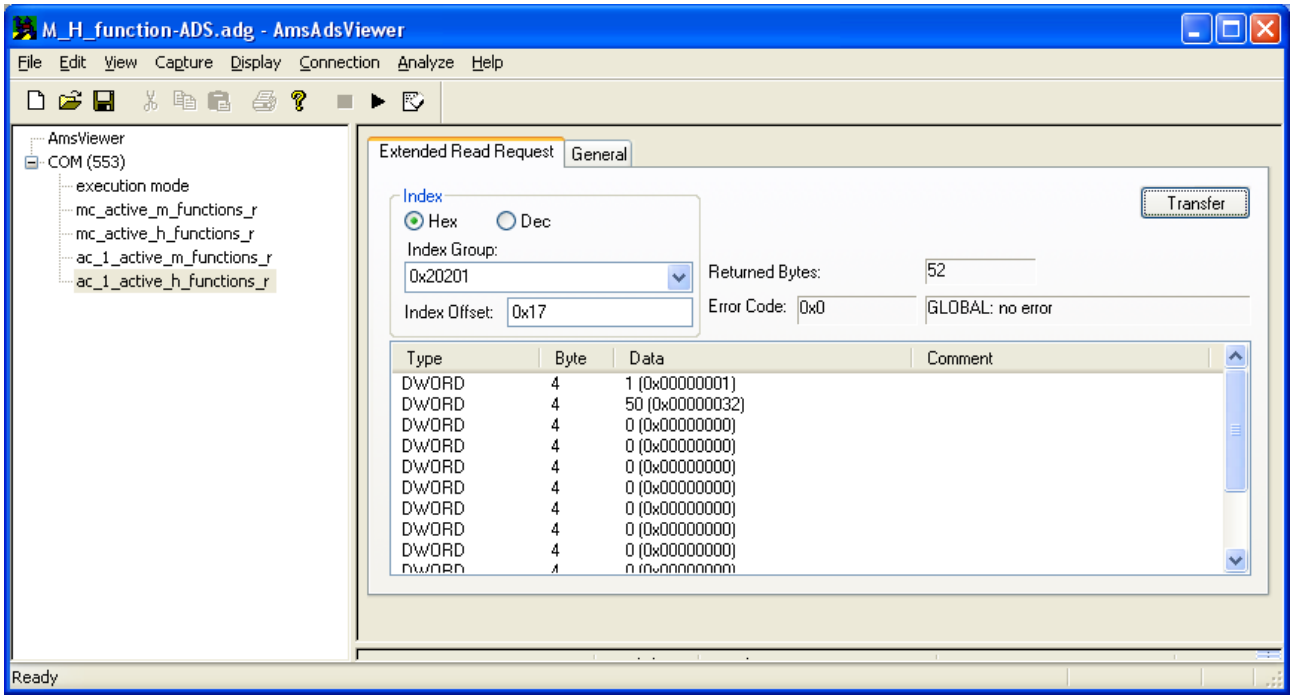


Fig. 26: AmsAdsViewer- Reads axis-specific H functions

4 Spindle M functions

In DIN 66025, the M functions M3, M4, M5, M19 and M40-M45 are reserved for spindle control on Class 1 and 2 machines. These M functions can be synchronised with an executing path motion. With M3, M4 and M5, synchronisation is always internal (speed reached) and, as an additional option, it is executed by the PLC (->PLC_INFO).

The use of M functions M3, M4, M5 and M19 on machines without spindles can be enabled with the channel parameter P-CHAN-00098 (spindle_m_fct_free).

The M functions M40-M45 are also freely available when gear changing is deactivated (P-CHAN-00052 (main_spdl_gear_change)).

In addition, as described in Section Example 2b: Programming an axis-specific M function in DIN syntax [▶ 60] and Example 1a: CNC spindle, internally synchronised [▶ 52], all freely available M functions can be output to a spindle for a specific axis.

Spindles are controlled by a CNC interpolator (CNC spindle) or by the PLC (PLC spindle).

For details of the use of spindles and the various spindle types, see [FCT-S1].

Parameter

P-CHAN-00045	Synchronisation method of the spindle function M3
P-CHAN-00047	Synchronisation method of the spindle function M4
P-CHAN-00049	Synchronisation method of the spindle function M5
P-CHAN-00043	Synchronisation method of the spindle function M19
P-CHAN-00069	Identifier of a PLC spindle with output of M functions via the channel range of the HLI
P-CHAN-00098	Enabling M3, M4, M5, M19 for any use

Synchronisation methods

The spindle M functions can be assigned the known synchronisation methods NO_SYNCH, MOS, MVS_SVS, MVS_SNS, MNS_SNS, MNE_SNS, MVS_SLM and MVS_SLP.

PLC_INFO

It only makes sense to use the PLC_INFO bit with NC spindles (position-controlled spindles). In addition to the synchronisation type, the PLC_INFO bit can be set for every spindle M function. This determines whether a spindle M function is output to the PLC or not and whether it must be acknowledged by the PLC.

If the PLC_INFO bit is not set, no output to PLC is sent to the PLC and internal synchronisation only takes place based on window monitoring for the position or speed.

For PLC spindles (speed-controlled spindles) please note the following:

In general, with every spindle M function, an output of the M function is also sent to the PLC automatically. It is therefore not necessary to additionally set the PLC_INFO bit.

4.1 Examples of spindle M functions

4.1.1 Example 1a: CNC spindle, internally synchronised

CNC spindle, internally synchronised

Initialisation in the axis parameter list

```
kenngr.achtstyp            0x00000004      Spindle
```

Initialising in the channel parameter list

```
spindel[0].m3_synch        0x00000002    MVS_SVS
spindel[0].m4_synch        0x00000004    MVS_SNS
spindel[0].m5_synch        0x00000004    MVS_SNS
spindel[0].m19_synch       0x00000004    MVS_SNS
```

Main spindle

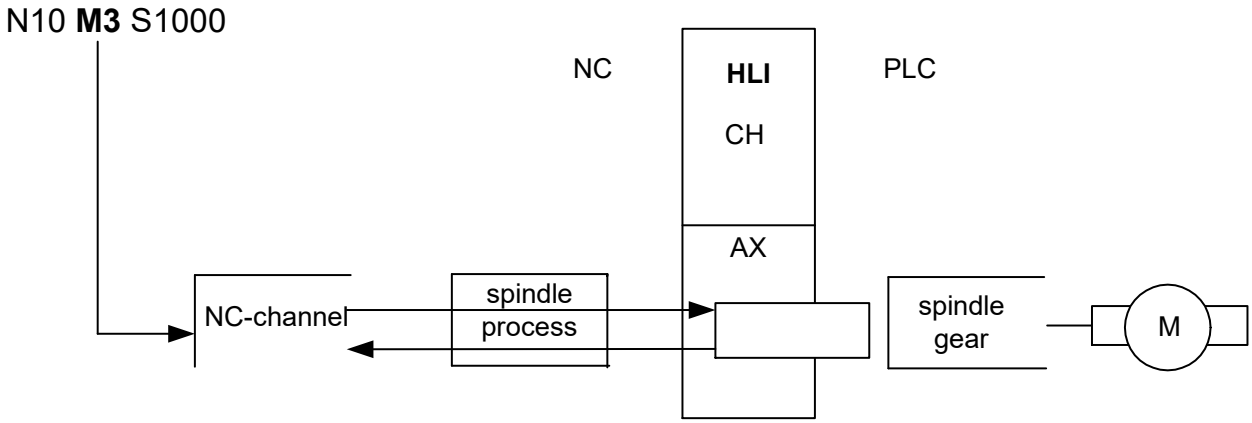


Fig. 27: CNC spindle, internally synchronised

The feed motion X100 is started after the spindle confirms execution of the M3 command. The M function is not output to the PLC.

The same example in axis-specific syntax for a further spindle

```
N10 S2[M3 REV1000] G01 F100 X100
```

4.1.2 Example 1b: CNC spindle, internal and synchronised by the PLC

CNC spindle, internal and synchronised by the PLC

To achieve output of M functions to the PLC, the PLC_INFO bit is set in the synchronisation method. The spindle then always executes the corresponding command immediately (asynchronously). However, corresponding to the synchronisation method, the path motion is also synchronised by the PLC acknowledgement.

Initialisation in the axis parameter list

```
kenngr.achstyp            0x00000004      Spindle
```

Initialisation in the channel parameter list

```
spindel[0].m3_synch       0x00020002    MVS_SVS, PLC_INFO
spindel[0].m4_synch       0x00020004    MVS_SNS, PLC_INFO
spindel[0].m5_synch       0x00000004    MVS_SNS
spindel[0].m19_synch      0x00000004    MVS_SNS
```

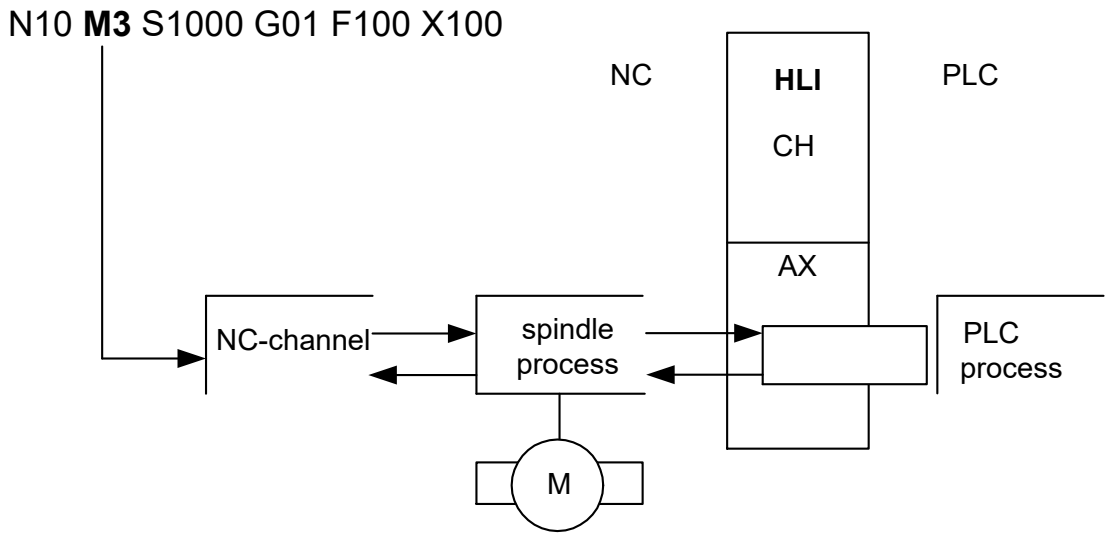


Fig. 28: CNC spindle, internal and synchronised by the PLC

The feed motion X100 is started after both the spindle and the PLC confirm execution of the M3 command.

4.1.3 Example 2a: PLC spindle

PLC spindle

For PLC spindles, the M functions are always output to the PLC. The PLC_INFO bit need not be set here.For details of configuration of a PLC spindle, see [FCT-S1].

Initialisation in the axis parameter list

```
kenngr.achstyp            0x00000004  Spindle
achs_mode                  0x00040000 externally controlled
                                       Spindle
antr_typ                   0x00000004  Simulation
```

Initialisation in the channel parameter list

```
spindel[0].m3_synch        0x00000002  MVS_SVS
spindel[0].m4_synch        0x00000004  MVS_SNS
spindel[0].m5_synch        0x00000004  MVS_SNS
spindel[0].m19_synch       0x00000004  MVS_SNS
```

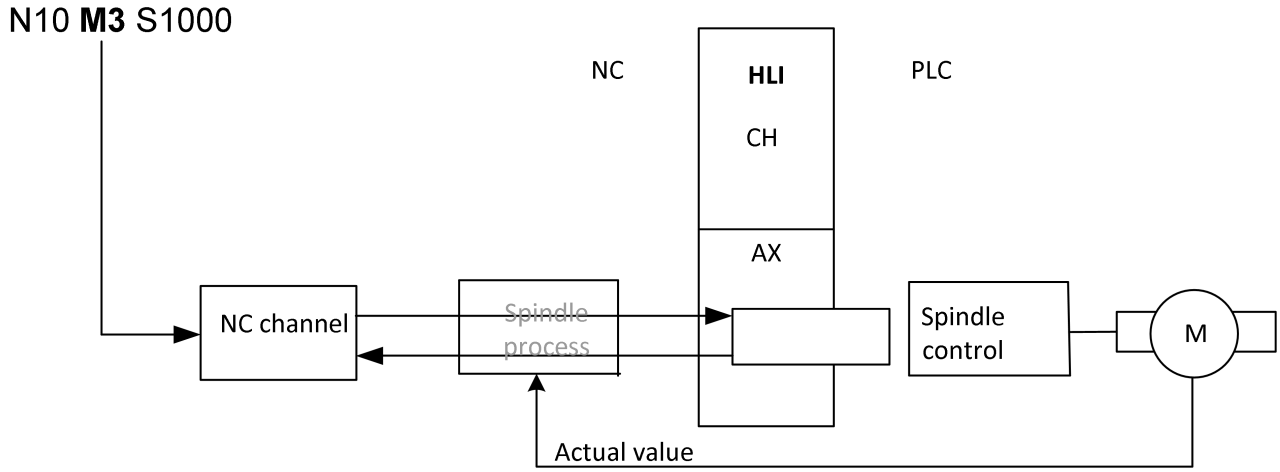


Fig. 29: PLC spindle

4.1.4 Example 2b: PLC spindle with output of M functions via the channel range

PLC spindle with output of M functions via the channel range

A PLC spindle can also be supplied with M functions via the channel-specific range of the HLI. In this case, the channel parameter P-CHAN-00069 (plc_control) must be assigned the value 1. The PLC_INFO need not be set for the output of M functions. In this case, the spindle is not configured as a CNC axis.

Initialisation in the channel parameter list

```
spindel[0].plc_control    1
spindel[0].m4_synch       0x00000004    MVS_SNS
spindel[0].m5_synch       0x00000004    MVS_SNS
spindel[0].m19_synch      0x00000004    MVS_SNS
```

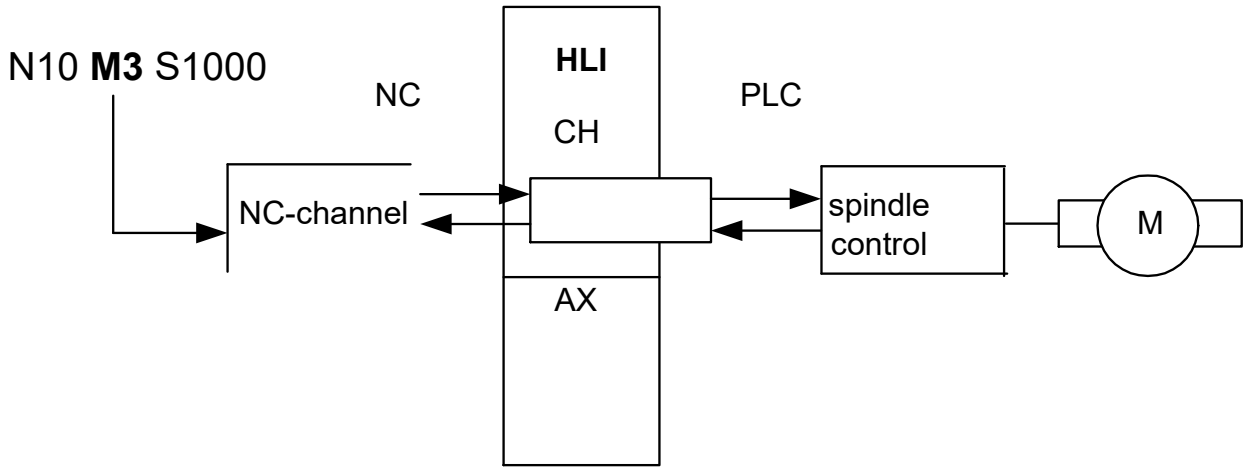


Fig. 30: PLC spindle with output of M functions via the channel range

4.2 Gear speed selection/parameter set changeover

The parameter P-CHAN-00052 (main_spindle_gear _change) in the channel parameter list activates the M functions M40-M45 to select the gear stages of the main spindle. The M functions M40-M45 can be freely used if gear changes are disabled.

M function	Meaning
M40-M45	Gear speed selection for the main spindle

When gear changes P-CHAN-00052 (main_spindle_gear_change) are activated, the M functions M40-M45 must be assigned the synchronisation method MVS_SVS by the parameter P-CHAN-00041 (m_synch[..]).

Output is always in the spindle-specific range. The PLC_INFO bit is not used here.

Possible:

- Output to other axes by axis-specific programming (e.g. X[M40]). The gear change function is then not active.

Not possible:

- Output to a spindle by axis-specific programming
- A default output to axes or spindles (channel parameters).
- The M functions M40-M45 are defined by specifying the prescribed synchronisation method.

Gear speed selection/parameter set changeover

Initialisation in the channel parameter list:

```
m_synch[40]                0x00000002     MVS_SVS
m_synch[41]                0x00000002     MVS_SVS
m_synch[42]                0x00000002     MVS_SVS
m_synch[43]                0x00000002     MVS_SVS
m_synch[44]                0x00000002     MVS_SVS
m_synch[45]                0x00000002     MVS_SVS
```

Activating gear changing:

```
main_spindle_gear_change   1
```

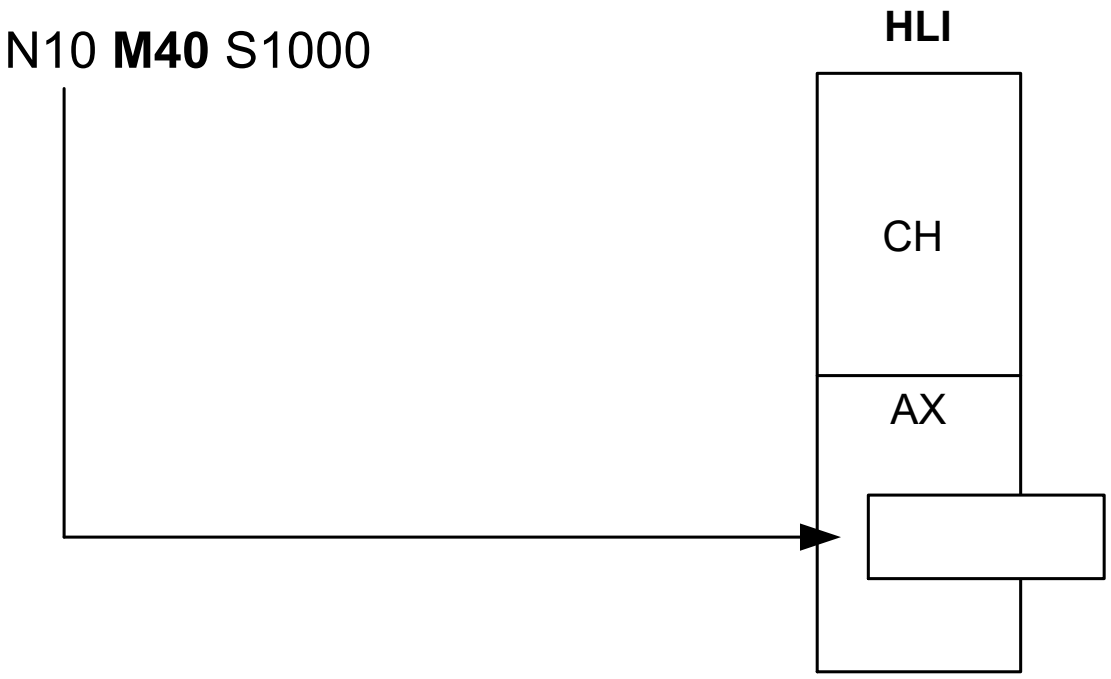


Fig. 31: Activating gear changing

5 Programming

5.1 Programming using the example of M functions, channel- and axis-specific

5.1.1 Example 1: Programming a channel-specific M function

Channel-specific M function

The M function M25 programmed in the block is output via the channel-specific range of the HLI.

Initialisation in the channel parameter list:

```
m_synch[25]   0x0002
```

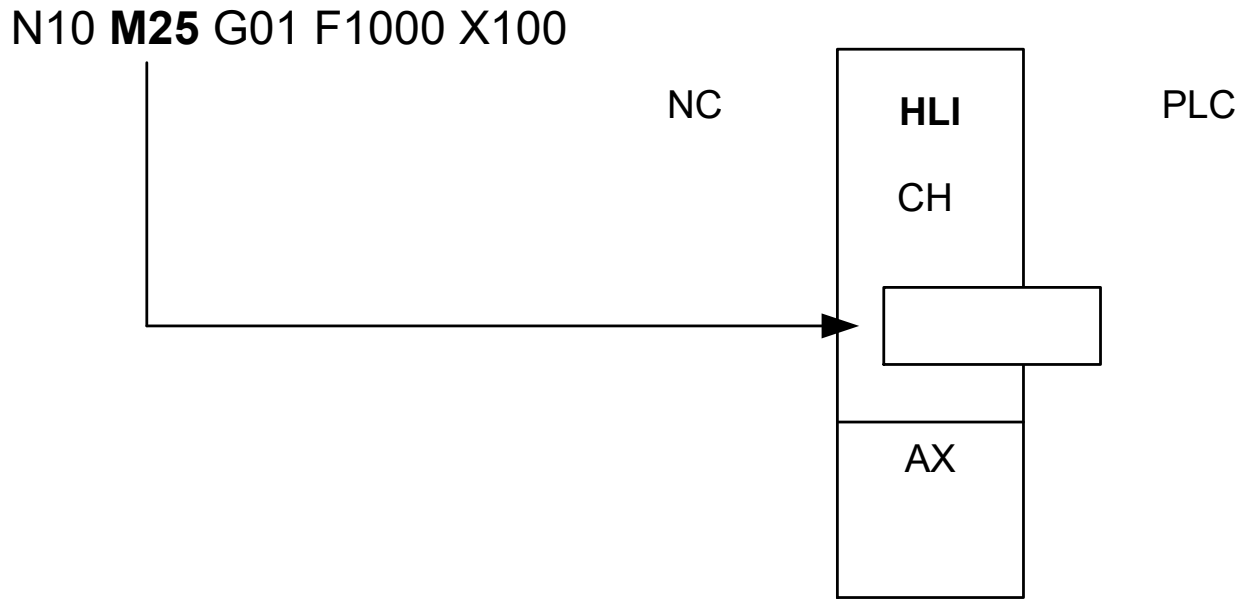


Fig. 32: Programming a channel-specific M function

5.1.2 Example 2a: Programming an axis-specific M function in extended syntax

Axis-specific M function in extended syntax

The M function M25 assigned in the block by programming the X axis is output via the axis-specific range of the HLI.

Initialisation in the channel parameter list:

```
m_synch[25]   0x0002
```

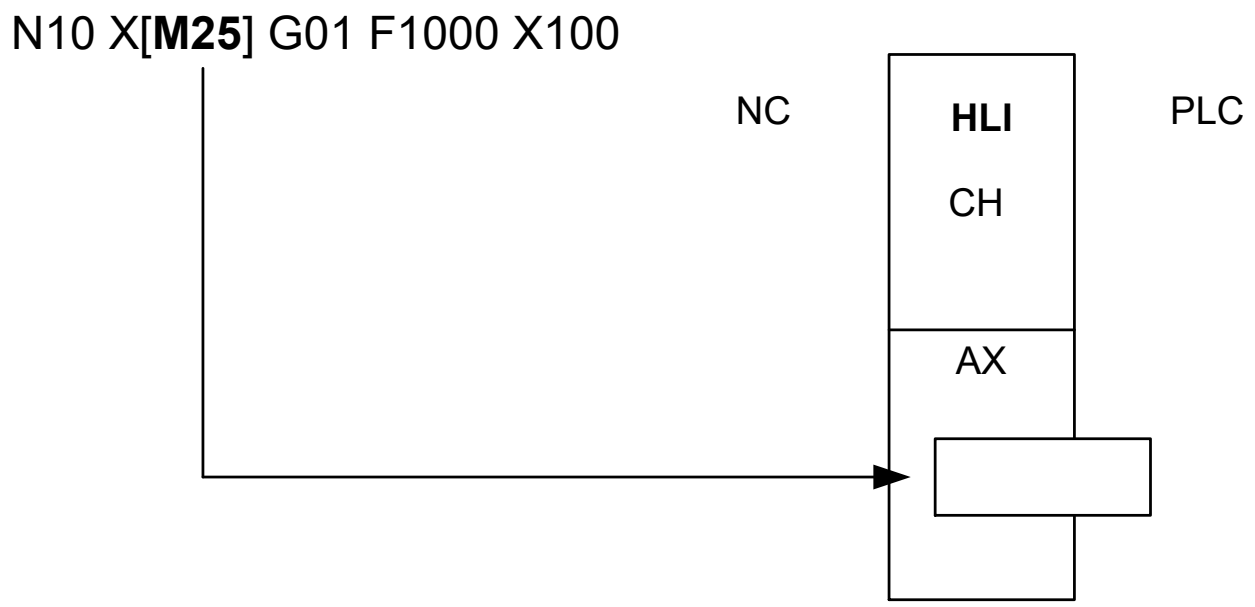


Fig. 33: Programming a channel-specific M function

5.1.3 Example 2b: Programming an axis-specific M function in DIN syntax

Compatibility mode

Axis-specific M and H functions require a compatibility mode. This permits the axis-specific output in existing CNC programs without changing the syntax:

The channel parameters

P-CHAN-00039 (m_default_outp_ax_name[MNr]) or

P-CHAN-00025 (h_default_outp_ax_name[HNr])

can define the axis-specific output of an M or H function as default. A channel-specific output then no longer takes place.

Parameter

P-CHAN-00039	Designation of the axis to which the M function with the number MNo is output
P-CHAN-00025	Designation of the axis to which the H function with the number HNo is output

Axis-specific M function in DIN syntax

The M function M25 assigned in the block by configuring the X axis is output via the axis-specific range of the HLI.

Initialisation in the channel parameter list:

```
m_synch[25]                    0x0002
m_default_outp_ax_name[25]     X
```

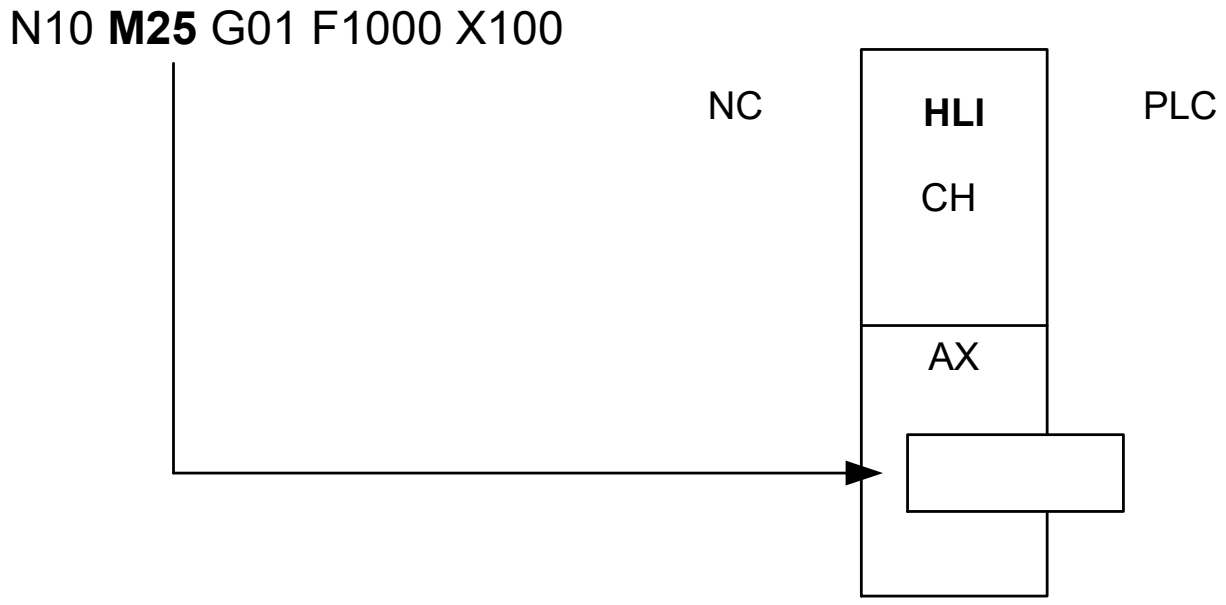


Fig. 34: Programming an axis-spec. M function in DIN syntax

5.2 Other examples

EXAMPLE 1

Optionally, it should be possible to output the M function to the channel or the X axis. Synchronisation is not required.

Optional output of M83

Channel parameter list:

```
m_synch[83]                          0x1

N10 M83 X[M83]
N20 M83
N30 X[M83]
```

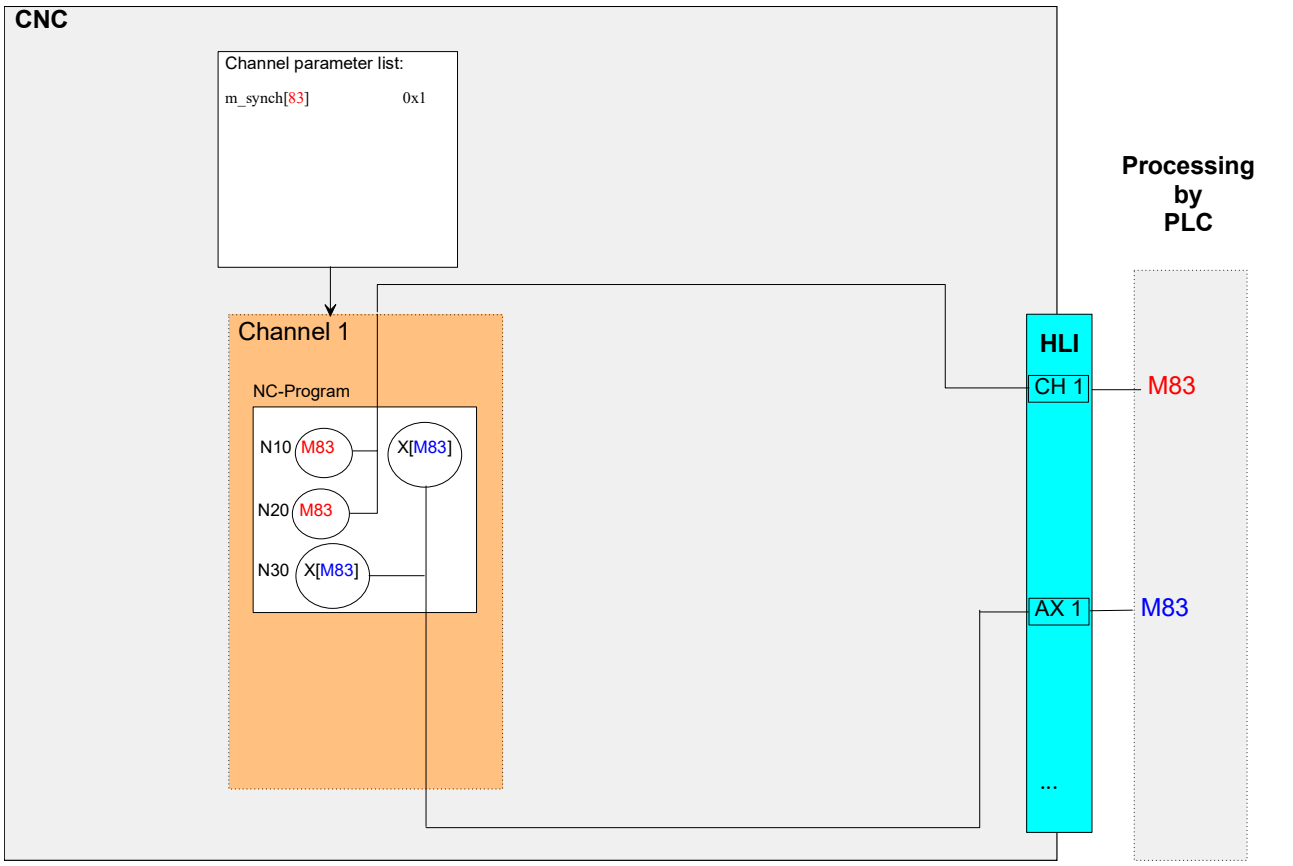


Fig. 35: Output of M83

EXAMPLE 2

In future, the H function H3 should always be output to the spindle axis S without the need for a change to CNC programs. Synchronisation is required before a motion is commenced. To simulate production time, an execution time of 1s is assumed.

H3 always output to the spindle axis S

Channel parameter list:

```
h_synch[3]                              0x2 MVS_SVS
h_default_outp_ax_name[3]               S
h_prozess_zeit[3]                       1000000

N10 X10 H3
```

EXAMPLE 3

The default M function M1 should be synchronised before a motion is commenced. To simulate production time, an execution time of 0.8 s must be assumed.

Synchronisation of default M1 function

Channel parameter list:

```
m_synch[1] 0x2 MVS_SVS
m_prozess_zeit[1] 800000

N10 M1 X100
```

6 Parameter

6.1 Overview

ID	Parameter	Description
P-CHAN-00025	h_default_output_ax_name[i]	Axis-specific output of H functions
P-CHAN-00026	h_prozess_zeit[i]	Process time of H functions
P-CHAN-00027	h_synch[i]	Synchronisation methods for H functions
P-CHAN-00033	late_sync_ready	Handling open 'late' synchronisation processes at end of CNC program
P-CHAN-00039	m_default_output_ax_name[i]	Axis-specific output of M functions
P-CHAN-00040	m_prozess_zeit[i]	Process time of M functions
P-CHAN-00041	m_synch[i]	Synchronisation types for M functions
P-CHAN-00042	m19_prozess_zeit	Process time of M19 functions
P-CHAN-00043	m19_synch	Synchronisation method M19
P-CHAN-00044	m3_prozess_zeit	Process time of the M03 function
P-CHAN-00045	m3_synch	Synchronisation mode M3
P-CHAN-00046	m4_prozess_zeit	Process time of the M04 function
P-CHAN-00047	m4_synch	Synchronisation mode M4
P-CHAN-00048	m5_prozess_zeit	Process time of the M05 function
P-CHAN-00049	m5_synch	Synchronisation mode M5
P-CHAN-00052	main_spindle_gear_change	Spindle gear change/parameter set changeover
P-CHAN-00069	plc_control	Identifier of a PLC spindle with output of M functions via the channel range of the HLI
P-CHAN-00070	m_pre_outp[i]	Path or time-related pre-output of M functions
P-CHAN-00080	s_prozess_zeit	Process time of the S function
P-CHAN-00098	spindle_m_fct_free	Free use of spindle M functions M3, M4, M5, M19
P-CHAN-00107	h_pre_outp[i]	Path or time-related pre-output of H functions
P-CHAN-00209	m_h_pre_outp_time_calc_mode	Profile calculation model for MET_MOS, MET_SVS
P-CHAN-00212	m_h_pre_outp_calc_value_to_go	Distance/time to go with MET_SVS, MEP_SVS
P-CHAN-00274	m_h_pre_outp_nbr_block	Number of NC blocks at M/H code look ahead
P-CHAN-00600	configuration.path_preparation.function	Defining functionalities for path preparation

6.2 Description

P-CHAN-00039	**Axis-specific M functions**
Description	User-specific M functions programmed in DIN syntax are processed and executed channel-specific. If the user forces axis-specific processing for specific M functions, it is possible to configure them using this parameter so that they have an axis-specific effect. An axis name can be assigned to each M function on which it should act. Both path axes and spindle axes are permissible.
Parameter	m_default_outp_ax_name[i] where i = 0 ... 999 (maximum number of M functions, application-specific)

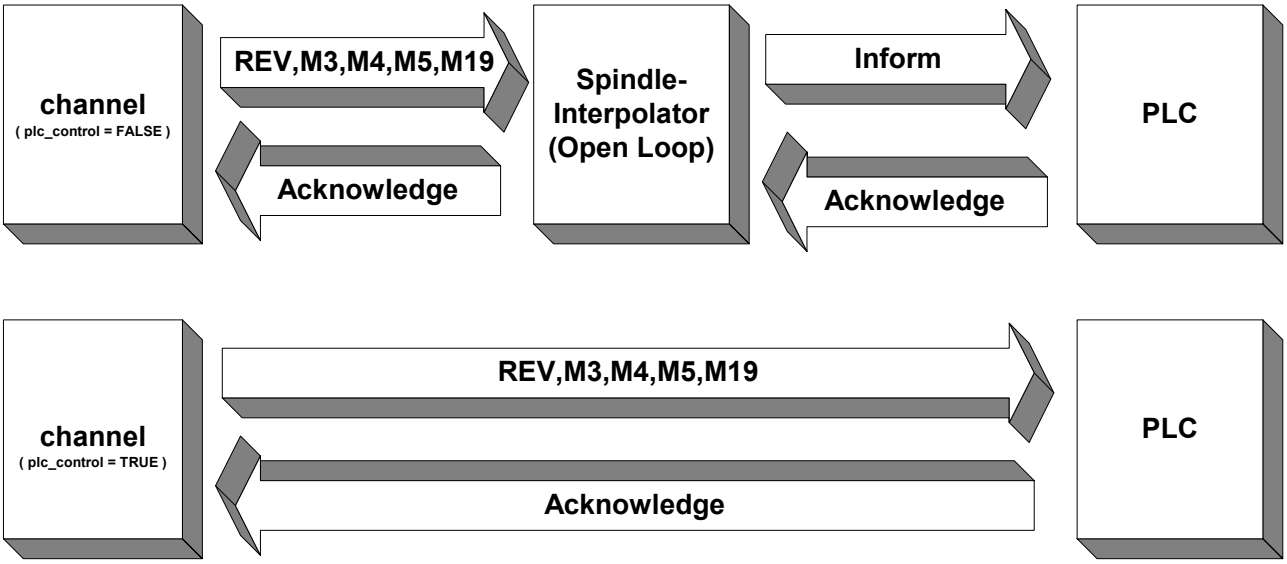
Data type	STRING
Data range	Maximum 16 characters (length of axis name, application-specific)
Dimension	----
Default value	*
Remarks	Parameterisation example: The user-specific M function M10 is to act on the Z axis when programmed in the DIN syntax. The user-specific M function M11 is act on the S2 spindle when programmed in the DIN syntax *m_default_outp_ax_name[10] Z* *m_default_outp_ax_name[11] S2* * Note: The default value of variables is a blank string.

P-CHAN-00025	**Axis-specific H functions**
Description	User-specific H functions programmed in DIN syntax are processed and executed channel-specific. If the user wishes to force axis-specific handling for specific H functions, it is possible to configure them using this parameter so that they have an axis-specific effect. Each H function can be assigned an axis name on which it is to act. Both path axes and spindle axes are permissible.
Parameter	h_default_outp_ax_name[i] where i = 0... 999 (maximum number of H functions, application-specific)
Data type	STRING
Data range	Maximum 16 characters (length of axis name, application-specific)
Dimension	----
Default value	*
Remarks	Parameterisation example: The user-specific H function H10 is to be programmed in DIN syntax to act on the Z axis. The user-specific H function H11 is to be programmed in DIN syntax to act on the S2 spindle axis. *h_default_outp_ax_name[10] Z* *h_default_outp_ax_name[11] S2* * Note: The default value of variables is a blank string.

P-CHAN-00052	**Enable mechanical gear change of main spindle**
Description	This parameter enables or disables gear changes for the main spindle.
Parameter	main_spindle_gear_change
Data type	BOOLEAN
Data range	0: Spindle gear changes disabled for the main spindle 1: Spindle gear changes enabled for the main spindle
Dimension	----
Default value	0
Remarks	The M functions to select the gear speeds of the main spindle M40–45 are activated by the parameter P-CHAN-00052 in the channel parameter list. The M functions M40–45 can be freely used if gear changes are disabled.

P-CHAN-00069	**Spindle control by PLC via channel specific interface**

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



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

Synchronisation definitions

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 [▶ 75] 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-00027	**Synchronisation type of H functions**
Description	The synchronisation type of the corresponding H function is defined in the array ‘h_synch[i]’. Here, the field index ‘i’ defines the number of the H function. This value indicates the synchronisation type of the H function, i.e. when a check is made for the presence of a PLC acknowledgement. A motion is not executed, or is stopped at the latest towards the end of the block, if no acknowledgement has arrived from the PLC. The synchronisation type is defined as a string constant or a hexadecimal value.
Parameter	h_synch[i] where i = 0 999 (maximum number of H functions, application-specific)
Data type	STRING
Data range	See figure below
Dimension	----
Default value	NOT_VALID

Remarks	H functions are consumption information and they must be fetched (read) from the PLC. This also applies to H functions of the type MOS, MEP_MOS and MET_MOS. Otherwise, this results in a blocked interface to the HLI in the CNC and as a consequence to an unexpected processing stop. **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-00107 (m_pre_outp[i]) must be assigned to 0. Otherwise, a license error is generated in case of microwebs if this function is not licensed or not enabled (see P-CHAN-00600 [▶ 75] Alternatively: P-STUP-00060) is inactive. Example h_synch[12] MVS_SVS 0x00000002 Note: Programming a UNS32 variable is permissible for downward compatibility reasons. Example: m_synch[12] 0x00000002

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

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

Remarks	Parameterisation example: For a (position-controlled) spindle 'S1' the spindle-specific M function M04 is assigned the synchronisation type MVS_SNS. The PLC is also informed. *spindel[0].bezeichnung S1* *spindle[0].log_achs_no 6* *spindel[0].s_synch MOS 0x00000001* spindel[0].m3_synch PLC_INFO | MVS_SVS 0x00020002 **spindel[0].m4_synch PLC_INFO | MVS_SNS 0x00020004** *spindle[0].m5_synch PLC_INFO | MVS_SVS 0x00020002* *spindle[0].m19_synch MNS_SNS 0x00000008* Note: Programming a UNS32 variable is permissible for downward compatibility reasons. Example: spindel[0].m4_synch 0x00020004

P-CHAN-00049	**Synchronisation type for M05**
Description	When the M05 function is used, the synchronisation type must be defined for the spindles used. The synchronisation type is defined as a string constant or a hexadecimal value.
Parameter	spindel[i].m5_synch
Data type	STRING
Data range	See Spindle-specific synchronisation types
Dimension	----
Default value	NO_SYNCH
Remarks	Parameterisation example: For a (position-controlled) spindle 'S1', the spindle-specific M function M05 is assigned the synchronisation type MVS_SVS. The PLC is also informed. *spindel[0].bezeichnung S1* *spindle[0].log_achs_no 6* *spindel[0].s_synch MOS 0x00000001* spindel[0].m3_synch PLC_INFO | MVS_SVS 0x00020002 *spindle[0].m4_synch PLC_INFO | MVS_SNS 0x00020004* **spindel[0].m5_synch PLC_INFO | MVS_SVS 0x00020002** *spindle[0].m19_synch MNS_SNS 0x00000008* Note: Programming a UNS32 variable is permissible for downward compatibility reasons. Example: spindel[0].m5_synch 0x00020002

P-CHAN-00043	**Synchronisation type for M19**
Description	When the M19 function is used, the synchronisation type must be defined for the spindles used. The synchronisation type is defined as a string constant or a hexadecimal value.
Parameter	spindel[i].m19_synch
Data type	STRING
Data range	See Spindle-specific synchronisation types
Dimension	----
Default value	NO_SYNCH
Remarks	Parameterisation example: For a (position-controlled) spindle 'S1' the spindle-specific M function M19 is assigned the synchronisation type MNS_SNS. The PLC is also informed. *spindel[0].bezeichnung S1* *spindle[0].log_achs_no 6* *spindel[0].s_synch MOS 0x00000001* *spindle[0].m3_synch PLC_INFO | MVS_SVS 0x00020002* *spindle[0].m4_synch PLC_INFO | MVS_SNS 0x00020004* *spindle[0].m5_synch PLC_INFO | MVS_SVS 0x00020002* **spindel[0].m19_synch MNS_SNS 0x00000008** Note: Programming a UNS32 variable is permissible for downward compatibility reasons. Example: spindel[0].m19_synch 0x00000008

P-CHAN-00070	Path or time-related pre-output of M functions
Description	This parameter is used in connection with M functions • of synchronisation types MET_SVS, MET_MOS and MEP_SVS, MEP_MOS. The lead time is specified for MET_SVS, MET_MOS and the lead distance for the MEP_SVS or MEP_MOS type. • of synchronisation types MOS, MVS_SVS, MVS_SNS, MNS_SNS, MOS_TS and the 'microjoints' function which requires a separate license (see [FCT-C1 [▶ 44]]). Path-related parameters make sense here. The field index 'i' defines the number of associated M functions. The value of m_pre_outp[i] defines the path and time-related output point before actual processing of the M function on the path.
Parameter	m_pre_outp[i] where i = 0 999 (maximum number of M functions, application-specific)
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	0.1 µm or µs
Default value	0
Remarks	The pre-output value can also be defined in the NC program [PROG//section V.G. variables]. **Caution:** If one of these synchronisation types is later changed into one which requires no pre-output value, P-CHAN-00070 (m_pre_outp[i]) must be assigned to 0. Otherwise, a license error is generated in case of microjoints if this function is not licensed or not enabled (see P-CHAN-00600 [▶ 75] alternatively P-STUP-00060). **Parameterisation example:** The user-specific M functions M96 and M98 must be output to the SPS 10 mm before reaching the synchronisation position in the block sequence. The user-specific M functions M97 and M99 to SPS must be output 40 milliseconds before reaching the time of synchronisation in the block sequence. *# Definition of M functions and synchronisation types* *# ==========* *m_synch[96] 0x01000000 MEP_SVS* *m_synch[97] 0x02000000 MET_SVS* *m_synch[98] 0x00100000 MEP_MOS* *m_synch[99] 0x00200000 MET_MOS* *#* *# Definition of pre-output path, pre-output time* *#==========* *m_pre_outp[96] 100000 in 0.1µm* *m_pre_outp[97] 40000 in µs* *m_pre_outp[98] 100000 in 0.1µm* *m_pre_outp[99] 40000 in µs*

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

Description	This parameter is used in connection with H functions • of synchronisation types MET_SVS, MET_MOS and MEP_SVS, MEP_MOS. The lead time is specified for MET_SVS, MET_MOS and the lead distance for the MEP_SVS or MEP_MOS type. • of synchronisation types MOS, MVS_SVS, MVS_SNS, MNS_SNS, MOS_TS and the 'microwebs' function which requires a separate license (see [FCT-C1 [▶ 44]]). Path-related parameters make sense here. The field index 'i' defines the number of the associated H function. The value of h_pre_outp[i] defines the path and time-related output point before actual processing of the H function on the path.
Parameter	h_pre_outp[i] where i = 0 999 (maximum number of H functions, application-specific)
Data type	UNS32
Data range	0 ... MAX(UNS32)
Dimension	0.1µm or µs
Default value	0
Remarks	The pre-output value can also be defined in the NC program [PROG//Chapter V.G. variables]. **Caution:** If the synchronisation type of an H function is later changed into one which requires no pre-output value, h_pre_outp[i] must be assigned to 0. Otherwise, a license error is generated in case of microwebs if this function is not licensed or not enabled (see P-STUP-00060). **Parameterisation example:** User-specific H functions H96 and H98 must be output to the PLC 10 mm before reaching the synchronisation position in the block sequence. The output of the user specific H functions H97 and H99 to the PLC must be executed 40 milliseconds before reaching the time of synchronisation in the block sequence. *# Definition of H functions and synchronisation types* *# ==========* *h_synch[96] 0x01000000 MEP_SVS* *h_synch[97] 0x02000000 MET_SVS* *h_synch[98] 0x00100000 MEP_MOS* *h_synch[99] 0x00200000 MET_MOS* *#* *# Definition of pre-output path, pre-output time* *#==========* *h_pre_outp[96] 100000 in 0.1µm* *h_pre_outp[97] 40000 in µs* *h_pre_outp[98] 100000 in 0.1µm* *h_pre_outp[99] 40000 in µs*

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

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

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

P-CHAN-00274	**Number of NC blocks at M/H code look ahead**
Description	This parameter increases the number of NC blocks for look ahead for M/H synchronisation types MEP_SVS and MET_SVS. By default the number of blocks is 50. Due to the additionally required block buffer, the parameter P-CHAN-00653 and possibly P-CHAN-00650 (or alternatively start-up parameter P-STUP-00071 and possibly P-STUP-00070must be adapted for values > 70. The increased number of blocks causes an increased cycle time in the real-time part of CNC.
Parameter	m_h_pre_outp_nbr_block
Data type	UNS32
Data range	50 ≤ P-CHAN.00274 ≤ 200
Dimension	----
Default value	50

Remarks	Configuration example:
	• Channel parameter list:
	`m_h_pre_outp_nbr_block 100 #100 blocks M code look ahead`
	`# P-CHAN-00653 - Large Look-Ahead buffer`
	`configuration.interpolator.parameter 100`
	`# P-CHAN-00655 - Activate customer-specific`
	`# setting for Look-Ahead`
	`configuration.interpolator.fct_enable[0] FCT_LOOK_AHEAD_CUSTOM`
	• Alternatively - in the start-up parameter list, modify instead of P-CHAN-00653 and P-CHAN-00650
	`configuration.channel[0].interpolator.parameter 100`
	`configuration.channel[0].interpolator.function FCT_LOOK_AHEAD_CUSTOM`

P-CHAN-00033	**Default setting for 'Late synchronization at program end' at program start**
Description	This parameter is used in conjunction with synchronisation types MVS_SLM and MVS_SLP (Late Sync). It defines the reaction to open Late Sync M functions at program end. This may occur if there is no G01 block in the NC program until program end (with MVS_SLM) or if no explicit synchronisation was programmed (with MVS_SLP).
Parameter	prog_start.late_sync_ready
Data type	BOOLEAN
Data range	0: Open Late Sync M functions at the end of the program are still active the next time the program is started. This means that one or several Late Sync M functions of the first NC program are only triggered by a Late Sync event (G01 block or #EXPL SYN) of the second NC program. 1: At program end, the program waits until all open Late Sync M functions are acknowledged by the PLC.
Dimension	----
Default value	0
Remarks	Parameterisation example: At program end, the program waits until all open Late Sync M functions are acknowledged by the PLC. *prog_start.late_sync_ready 1*

Production time calculation

P-CHAN-00040	**Timeout / process times of M functions for machining time calculation**
Description	The timeout times of M functions are set in the array ‘m_prozess_zeit[i]’. The field index ‘p’ defines the number of the M function. If the calculation of machining time is activated, the process times of the M functions are specified in this array.
Parameter	m_prozess_zeit[i] where i = 0 999 (maximum number of M functions, application-specific)
Data type	UNS32
Data range	0 ≤ m_prozess_zeit ≤ MAX(UNS32)
Dimension	µs
Default value	0
Remarks	This element is currently only used to calculate machining time. Parameterisation example: The timeout and process time of M function 'M15' to specified as 0.5 s. *m_prozess_zeit[15] 500000*

P-CHAN-00026	**Timeout / process times of H functions for machining time calculation**
Description	The timeout times of H functions are specified in the array ‘h_prozess_zeit[i]’. The field index ‘i’ defines the number of the H function. If the calculation of machining time is activated, the process times of H functions are specified in this array.

Parameter	h_prozess_zeit[i] here i = 0... 999 (maximum number of H functions, application-specific)
Data type	UNS32
Data range	0 < P-CHAN-00026 < MAX(UNS32)
Dimension	[µs]
Default value	0
Remarks	This element is currently only used to calculate machining time. Parameterisation example: The following example specifies the timeout and process time of H function 'H1' to 20 ms. *h_prozess_zeit[15] 20000*

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

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

P-CHAN-00046	**Timeout and process time of M04 to calculate machining time**
Description	This parameter specifies the timeout and the process time of the M function M04 when the machining time calculation is activated.
Parameter	spindel[i].m4_prozess_zeit
Data type	UNS32

Data range	0 < m4_prozess_zeit < MAX(UNS32)
Dimension	µs
Default value	0

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

P-CHAN-00048	**Timeout and process time of M05 to calculate machining time**
Description	This parameter specifies the timeout and the process time of the M function M05 when the machining time calculation is activated.
Parameter	spindel[i].m5_prozess_zeit
Data type	UNS32
Data range	0 < m5_prozess_zeit < MAX(UNS32)
Dimension	µs
Default value	0

Remarks	Currently, the parameter is only used to calculate machining time.
	Parameterisation example: For spindle 'S1' the process time for M05 is defined as 1 s.
	`spindel[0].bezeichnung S1`
	`spindel[0].log_achs_nr 6`
	`:`
	`spindel[0].s_prozess_zeit 1000000`
	`spindel[0].m3_prozess_zeit 1000000`
	`spindel[0].m4_prozess_zeit 1000000`
	`spindel[0].m5_prozess_zeit 1500000`
	`spindel[0].m19_prozess_zeit 2000000`

P-CHAN-00042	**Timeout and process time of M19 to calculate machining time**
Description	This parameter specifies the timeout and the process time of the M function M19 when the machining time calculation is activated.
Parameter	spindel[i].m19_prozess_zeit
Data type	UNS32
Data range	0 < m19_prozess_zeit < MAX(UNS32)
Dimension	µs
Default value	0

Remarks	Currently, the parameter is only used to calculate machining time.
	Parameterisation example: The process time for M19 is set to 2s for the spindle 'S1'.
	`spindel[0].bezeichnung S1`
	`spindel[0].log_achs_nr 6`
	`:`
	`spindel[0].s_prozess_zeit 1000000`
	`spindel[0].m3_prozess_zeit 1000000`
	`spindel[0].m4_prozess_zeit 1000000`
	`spindel[0].m5_prozess_zeit 1500000`
	`spindel[0].m19_prozess_zeit 2000000`

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

Data range	See Path preparation function table
Dimension	----
Default value	FCT_DEFAULT
Remarks	Parameter is available as of the following Builds: V2.11.2040.04 ; V2.11.2810.02 ; V3.1.3079.17 ; V3.1.3107.10 Functions can be defined in P-CHAN-00605 and P-CHAN-00606 depending on the machining mode.

6.3 Method to create an M or H function in the channel parameter list

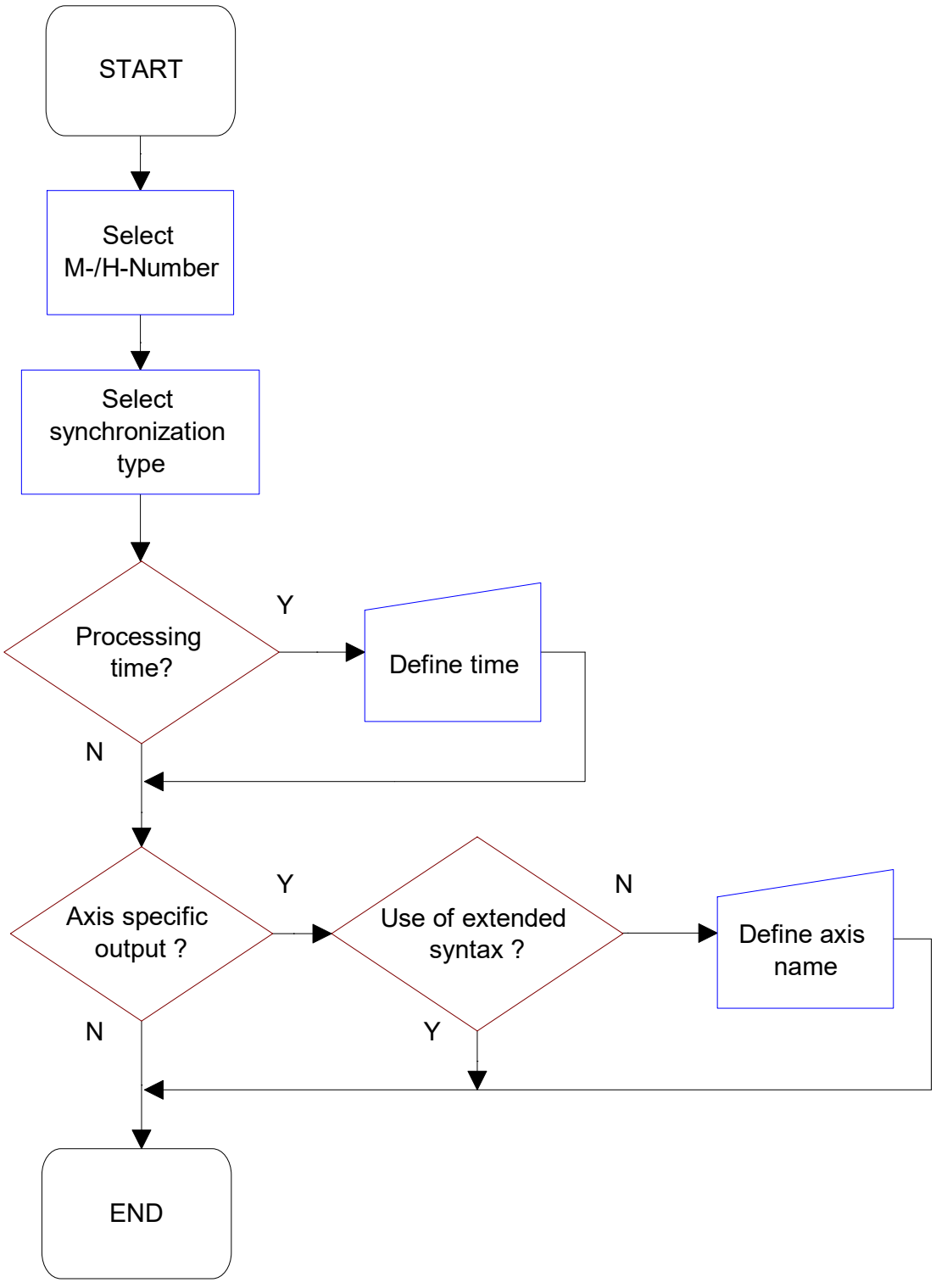


Fig. 36: Flow diagram to create an M/H function

7 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963-157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963-460
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

Index

P

P-CHAN-00025 64
P-CHAN-00026 73
P-CHAN-00027 67
P-CHAN-00033 73
P-CHAN-00039 63
P-CHAN-00040 73
P-CHAN-00041 65
P-CHAN-00042 75
P-CHAN-00043 69
P-CHAN-00044 74
P-CHAN-00045 68
P-CHAN-00046 74
P-CHAN-00047 68
P-CHAN-00048 75
P-CHAN-00049 69
P-CHAN-00052 64
P-CHAN-00069 64
P-CHAN-00070 70
P-CHAN-00080 74
P-CHAN-00098 65
P-CHAN-00107 70
P-CHAN-00209 71
P-CHAN-00212 72
P-CHAN-00274 72
P-CHAN-00600 75

More Information:
www.beckhoff.com/TF5200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com