Description of Functions 07/2004 Edition

sinumerik

SINUMERIK 840D/840Di/810D
ISO Dialects for SINUMERIK

SIEMENS

SIEMENS

SINUMERIK 840D/840Di/810D

ISO Dialects for SINUMERIK

Description of Functions

Valid for

Control	*Software version*
SINUMERIK 840D powerline	7
SINUMERIK 840DE powerline	7
SINUMERIK 840Di	3
SINUMERIK 840DiE (export version)	3
SINUMERIK 810D powerline	7
SINUMERIK 810DE powerline	7

07.2004 Edition

SINUMERIK® Documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the “Remarks” column.

Status code in the “Remarks” column:

A . . . . . New documentation.
B . . . . . Unrevised reprint with new order no.
C . . . . . Revised edition with new status.
If factual changes have been made on the page in relation to the same software version, this is indicated by a new edition coding in the header on that page.

Edition	**Order No.**	**Remarks**
08.99	6FC5297-5AE10-0BP0	**A**
04.00	6FC5297-5AE10-0BP1	**C**
10.00	6FC5297-6AE10-0BP0	**C**
09.01	6FC5297-6AE10-0BP1	**C**
12.01	6FC5297-6AE10-0BP2	**C**
11.02	6FC5297-6AE10-0BP3	**C**
07.04	6FC5297-6AE10-0BP4	**C**

This book is part of the documentation on CD-ROM **(DOCONCD)**

Edition	**Order No.**	**Remarks**
09.04	6FC5 298-7CA00-0BG1	**C**

Trademarks
SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIROTEC®, SINUMERIK® and SIMODRIVE® are trademarks of Siemens. Other product names used in this documentation may be trademarks which, if used by third parties, could infringe the rights of their owners.

Further information is available on the Internet under:
http://www.siemens.com/motioncontrol

This publication was produced with Interleaf V7.

Other functions not described in this documentation might be executable in the control. However, no claim can be made regarding the availability of these functions when the equipment is first supplied or for service cases.

We have checked that the contents of this document correspond to the hardware and software described. Nonetheless, differences might exist and therefore we cannot guarantee that they are completely identical. The information contained in this document is, however, reviewed regularly and any necessary changes will be included in the next edition. We welcome suggestions for improvement.

Subject to changes without prior notice.

Order No. 6FC5-297-6AE10-0BP4
Printed in Germany

Siemens Aktiengesellschaft

Preface

Structure of the documentation

The SINUMERIK documentation is structured in three levels:

- General documentation
- User documentation
- Manufacturer/service documentation.

For detailed information on further publications on SINUMERIK 840D/840Di/ 810D, as well as on publications applicable to all SINUMERIK control systems, please contact your regional Siemens branch office.

Reader group

This documentation is intended for use by manufacturers of machine tools with SINUMERIK 840D or SINUMERIK 810D and SIMODRIVE 611D.

Hotline

If you have any questions about the control, please contact the hotline:

A&D Technical Support
Phone.: ++49-180-5050-222
Fax: ++49-180-5050-223
Email: adsupport@ad.siemens.com

Please send any questions about the documentation (suggestions for improvement, corrections) to the following fax number or email address:

Fax: ++49-9131 / 98-2176
Email: motioncontrol.docu@erlf.siemens.de

Fax form: see reply form at the end of the manual.

Internet address

http://www.siemens.com/motioncontrol

SINUMERIK 840D powerline

With effect from 09.2001 the

- SINUMERIK 840D powerline and
- SINUMERIK 840DE powerline

have been given improved performance. See the hardware description below for the list of the available **powerline** modules:

References: /PHD/, Configuring Manual SINUMERIK 840D

SINUMERIK 810D powerline

With effect from 12.2001 the

- SINUMERIK 810D powerline and
- SINUMERIK 810DE powerline

have been given improved performance. See the hardware description below for the list of the available **powerline** modules:

References: /PHC/, Configuring Manual SINUMERIK 810D

Target readers

- Configuring engineers
- Electricians and start-up specialists
- Service and operating personnel

The purpose of this manual

The information in this manual makes it possible to import and use parts programs from external CNC systems.

Indexes and references

For your better orientation, this manual offers a list of contents and the following appendices:

- References
- Index
- Index of commands

Warning notes

The following warning notes with graded degrees of importance are used in this documentation:

Danger
Indicates an imminently hazardous situation which, if not avoided, **will** result in death or serious injury or in substantial property damage.

Warning
Indicates a potentially hazardous situation which, if not avoided, **could** result in death or serious injury or in substantial property damage.

Caution
Used with the safety alert symbol indicates a potentially hazardous situation which, if not avoided, **may** result in minor or moderate injury or in property damage.

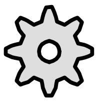
Caution
Used without safety alert symbol indicates a potentially hazardous situation which, if not avoided, **may** result in property damage.

Notice
Used without the safety alert symbol indicates a potential situation which, if not avoided, **may** result in an undesirable result or state.

Further information

Important
Important indicates an important or especially relevant item of information.

Note
The "note" symbol is displayed in this document to draw your attention to information relevant to the subject in hand.



Machine manufacturer

The symbol shown is found in this documentation whenever the machine manufacturer can influence or amend the feature described. Please note the machine manufacturer's specifications.

Trademarks

IBM is a registered trademark of the International Business Corporation.
MS DOS and WINDOWS^{TM} are registered tradmarks of the Microsoft Corporation.

Contents

Notes

1 Brief Description

Introduction

Parts programs can be read in from external CNC systems, and can then be edited and executed.

This manual describes the startup measures and procedures necessary to run NC programs created on an external CNC system. Functional differences are also explained.

Note

For a detailed description of the external programming functions, please refer to the original documentation of the external CNC system.

Terms used

The following terms are defined for this manual:

- ISO Dialect M is similar to the G code of the “Fanuc16 Milling” control
- ISO Dialect T is similar to the G code of the “Fanuc16 Turning” control System B
- ISO Dialect Original is equivalent to the original Fanuc16 control

■

Notes

Programming 2

2.1 Activation of functions

Machine data 18800 $MN_EXTERN_LANGUAGE is used to activate the external language. The language type, ISO Dialect-M or T is selected with machine data 10880 $MN_EXTERN_CNC_SYSTEM.
The external language can be activated separately for each channel. For example, channel 1 can operate in ISO mode but channel 2 is active in Siemens mode.

Switchover

The following two G commands from Group 47 are used to switch between Siemens mode and ISO Dialect mode:

- G290 Siemens NC programming language active
- G291 ISO Dialect NC programming language active

The active tool, tool offsets and zero offsets remain active here (see Subsection 2.2.4 and Section 2.4).

Siemens mode

The following conditions apply when Siemens mode is active:

- Siemens G commands are interpreted on the control by default.
- It is not possible to extend the Siemens programming system with ISO Dialect functions because some of the G functions have different meanings.
- Downloadable MD files can be used to switch the control to ISO Dialect mode. In this case, the user sees the ISO Dialect mode by default.

ISO Dialect mode

The following conditions apply when ISO Dialect mode is active:

- Only ISO Dialect G codes can be programmed, not Siemens G codes.
- It is not possible to use a mixture of ISO Dialect code and Siemens code in the same NC block.
- It is not possible to switch between ISO Dialect M and ISO Dialect T via G command
- If further Siemens functions are to be used, it is necessary to switch to Siemens mode first (exception: program branches and subprogram calls, see Subsection 2.3.2)

Power ON/Reset

Table 10-1 shows the possible combinations of machine data $MN_EXTERN_CNC_SYSTEM and $MC_GCODE_RESET_VALUE. This specifies the Power ON/Reset response.

Table 2-1 Activation of functions

After Power ON/Reset...	$MC_GCODE_RESET_VALUES[46] =	$MN_EXTERN_CNC_SYSTEM =
Siemens mode active, switch-over to ISO Dialect M possible	**1** G290 Siemens mode	**1** ISO Dialect M
Siemens mode active, switch-over to ISO Dialect T possible	**1** G290 Siemens mode	**2** ISO Dialect T
ISO Dialect M active, switchover to Siemens mode possible	**2** G291 ISO Dialect mode	**1** ISO Dialect M
ISO Dialect T active, switchover to Siemens mode possible	**2** G291 ISO Dialect mode	**2** ISO Dialect T

Modal G commands

Modal G commands which have the same function in both systems (Siemens and ISO Dialect) are treated as follows.
When these G codes are programmed in one language, the equivalent G code in the other language is determined and activated. The following G codes are affected.

Data management

ISO programs can be both read into and output from the MMC 103 in punchtape format.
ISO programs which have been read in are stored in the NC data management system as main programs in the default path:
_N_WKS_DIR/_N_SHOPMILL_WPD.

You can change the entry by editing the file DINO.INI in the USER directory. You will find further information in the publication

References: /IAM/, IM3: MMC Installation and Startup Guide, Section 3.1.

2.1.1 Switchover from ISO mode to Siemens mode

G290/291

G commands 290/291 can be used from the parts program to change mode.
On switchover, the display of current G codes also changes.

G65/66

Non-modal and modal macro:
The programmed subprogram is called. Switchover to Siemens mode only takes place when the PROC instruction is used in the first line of the subprogram.
If a program of this type is terminated with M17 or RET, when the subprogram returns, the mode is switched back to ISO mode.

Siemens subprogram call in ISO mode

Modal and non-modal subprogram calls, e.g.

```
     N100 CALL "SHAFT"
or
     N100 MCALL SHAFT
or
     N100 SHAFT
```

Modal and non-modal subprogram calls with parameter passing

```
     N100 MCALL SHAFT("ABC", 33.5)      or
     N100 SHAFT("ABC", 33.5)
```

Subprogram calls with path name

```
    N100 CALL "/_N_SPF_DIR/SHAFT
or
    N100 MCALL /_N_SPF_DIR/SHAFT
or
    N100 PCALL /_N_SPF_DIR/SHAFT
```

Siemens mode is selected implicitly on subprogram calls, and the system is switched back to ISO Dialect mode at the end of the subprogram.

Modal, non-modal cycles

If a modal or non-modal cycle is programmed in ISO mode, a shell cycle will be called.
This call results in switchover to Siemens mode.

2.2 G commands

The G codes of ISO Dialect T refer to G code system B (see also 4.1.5).

The active G codes in ISO mode can be read using system variable $P_EXTGG[...]. The numbers alongside the G code specify the respective value in $P_EXTGG[...]. Machine data 20154 EXTERN_GCODE_RESET_VALUES[n]: 0, ..., 30 is used to specify the G codes that are effective on start-up when the NC channel is not operating in Siemens mode.

Table 2-2 The default setting is indicated by 1)

ISO Dialect T		ISO Dialect M		Description
Group 1				
G00 1)	1	G00 1)	1	Rapid traverse
G01	2	G01	2	Linear motion
G02	3	G02	3	Circle/helix, clockwise
		G02.2	6	Involute, clockwise
G03	4	G03	4	Circle/helix, counterclockwise
		G03.2	7	Involute, counterclockwise
G33	5	G33	5	Thread cutting with constant lead
G34	9			Thread cutting with variable lead
G77	6			Longitudinal turning cycle
G78	7			Thread cutting cycle
G79	8			Face turning cycle
Group 2				
		G17 1)	1	XY plane
		G18	2	ZX plane
		G19	3	YZ plane
G96	1			Constant cutting rate ON
G97 1)	2			Constant cutting rate OFF
Group 3				
G90 1)	1	G90 1)	1	Absolute programming
G91	2	G91	2	Incremental programming
Group 4				
		G22	1	Working area limitation, protection zone 3 ON
		G23 1)	2	Working area limitation, protection zone 3 OFF
G68	1			Double turret/slide on
G69 1)	2			Double turret/slide off

Table 2-2 The default setting is indicated by 1)

ISO Dialect T	ISO Dialect M	Description
Group 5		
	G93 3	Inverse-time feedrate (rpm)
G94 1	G94 1) 1	Feed in [mm/min, inch/min]
G95 1) 2	G95 2	Revolutional feedrate in [mm/rev, inch/rev]
Group 6		
G20 1) 1	G20 1) (G70) 1	Input system inch
G21 2	G21 (G71) 2	Input system metric
Group 7		
G40 1) 1	G40 1) 1	Deselect cutter radius compensation
G41 2	G41 2	Compensation to left of contour
G42 3	G42 3	Compensation to right of contour
Group 8		
	G43 1	Tool length compensation positive ON
	G44 2	Tool length compensation negative ON
	G49 1) 3	Tool length compensation OFF
Group 9		
G22 1		Working area limitation, protection zone 3 ON
G23 1) 2		Working area limitation, protection zone 3 OFF
	G73 1	Deep hole drilling cycle with chipbreaking
	G74 2	Counterclockwise tapping cycle
	G76 3	Fine drilling cycle
	G80 1) 4	Cycle OFF
	G81 5	Counterbore drilling cycle
	G82 6	Countersink drilling cycle
	G83 7	Deep hole drilling cycle with swarf removal
	G84 8	Clockwise tapping cycle
	G85 9	Drilling cycle
	G86 10	Drilling cycle, retraction with G00
	G87 11	Reverse countersinking
	G89 13	Drilling cycle, retraction with machining feed
Group 10		
G80 1) 1		Drilling cycle OFF
G83 2		Face deep hole drilling
G84 3		Face tapping
G85 4		End face drilling cycle

Table 2-2 The default setting is indicated by 1)

ISO Dialect T		ISO Dialect M		Description
G87	5			Side deep hole drilling
G88	6			Side tapping
G89	7			Side drilling
		G98 1)	1	Return to starting point for fixed cycles
		G99	2	Return to point R for fixed cycles
Group 11				
G98 1)	1			Return to starting point for drilling cycles
G99	2			Return to point R for drilling cycles
		G50 1)	1	Scaling OFF
		G51	2	Scaling ON
Group 12				
G66	1	G66	1	Modal macro call
G67 1)	2	G67 1)	2	Delete modal macro call
Group 13				
		G96	1	Constant cutting rate ON
		G97 1)	2	Constant cutting rate OFF
Group 14				
G54 1)	1	G54 1)	1	Select zero offset
G55	2	G55	2	Select zero offset
G56	3	G56	3	Select zero offset
G57	4	G57	4	Select zero offset
G58	5	G58	5	Select zero offset
G59	6	G59	6	Select zero offset
G54 P{1...48}	1	G54 P{1...48}	1	Extended zero offsets
		G54.1	7	Extended zero offset
G54 P0	1	G54 P0	1	"External ZO extOffset"
Group 15				
		G61	1	Exact stop modal
		G62	4	Automatic corner override
		G63	2	Tapping mode
		G64 1)	3	Continuous-path mode
Group 16				
G17	1			XY plane
G18 1)	2			ZX plane
G19	3			YZ plane

Table 2-2 The default setting is indicated by 1)

ISO Dialect T		ISO Dialect M		Description
		G68	1	Rotation ON 2D 3D
		G69 1)	2	Rotation OFF
Group 17				
		G15 1)	1	Polar coordinates OFF
		G16	2	Polar coordinates ON
Group 18 (non-modal)				
G04	1	G04	1	Dwell time in [s] or spindle revolutions
G05	20	G05	18	High-speed cycle cutting
G05.1	22	G05.1	20	High speed cycle –> call CYCLE305
G07.1	18	G07.1	16	Cylindrical interpolation
		G08	12	Feedforward control ON/OFF
		G09	2	Exact stop
G10	2	G10	3	Write zero offset/tool offset
G10.6	19	G10.6	17	Rapid lift ON/OFF (T) Retraction from contour (POLF) (M)
		G11	4	Terminate parameter input
G27	16	G27	13	Referencing check (available soon)
G28	3	G28	5	Approach 1st reference point
G30	4	G30	6	Approach 2nd/3rd/4th reference point
G30.1	21	G30.1	19	Floating reference position
G31	5	G31	7	Measurement with touch-trigger probe
G52	6	G52	8	Programmable zero offset
G53	17	G53	9	Approach position in machine coordinate system
G65	7	G65	10	Call macro
G70	8			Finishing cycle
G71	9			Stock removal cycle longitudinal axis
G72	10			Stock removal cycle transverse axis
		G72.1	14	Contour repetition with rotation
		G72.2	15	Contour repetition, linear
G73	11			Repeat contour
G74	12			Deep hole drilling and recessing in longitudinal axis (Z)
G75	13			Deep hole drilling and recessing in facing axis (X)
G76	14			Multiple thread cutting cycle
G92	15	G92	11	Preset actual value memory, spindle speed limitation
G92.1	23	G92.1	21	Reset actual value, reset WCS

Table 2-2 The default setting is indicated by 1)

ISO Dialect T	ISO Dialect M	Description
Group 20		
G50.2 1) 1		Polygon turning OFF
G51.2 2		Polygon turning ON
Group 21		
G13.1 1) 1		TRANSMIT OFF
G12.1 2		TRANSMIT ON
Group 22		
	G50.1 1	Mirroring on programmed axis OFF
	G51.1 2	Mirroring on programmed axis ON
Group 25		
	G13.1 1	Polar coordinates, interpolation
	G12.1 2	Polar coordinates, interpolation
Group 31		
G290 1) 1	G290 1) 1	Select Siemens mode
G291 2	G291 2	Select ISO Dialect mode

Table 2-3 G commands are functionally identical in Siemens mode and in ISO Dialect mode

G commands in Siemens mode	Corresponding G commands in ISO Dialect T	Corresponding G commands in ISO Dialect M
Group 1: G00, G01, G02, G03, G33	Group 1: G00, G01, G02, G03, G33	Group 1: G00, G01, G02, G03, G33
Group 6: G17, G18, G19	Group 16: G17, G18, G19	Group 2: G17, G18, G19
Group 7: G40, G41, G42	Group 7: G40, G41, G42	Group 7: G40, G41, G42
Group 8: G54 to G554	Group 14: G54 to G59 G54 P1 to P48	Group 14: G54 to G59, G54 P1 to P48
Group 10: G60, G64		Group 15: G60, G64
Group 13: G700, G710	Group 6: G20, G21	Group 6: G20, G21
Group 14: G90, G91	Group 3: G90, G91	Group 3: G90, G91
Group 15: G94 G95 G96 G961 G97 G971	Group 5: G94 Group 2: G97 Group 5: G95 Group 2: G97 Group 5: G95 Group 2: G96 Group 5: G94 Group 2: G96 Group 5: G95 Group 2: G97 Group 5: G94 Group 2: G97	Group 5: G94 Group 13: G97 Group 5: G95 Group 13: G97 Group 5: G95 Group 13: G96 Group 5: G94 Group 13: G96 Group 5: G95 Group 13: G97 Group 5: G94 Group 13: G97

Note

If individual G codes of the groups in Table 2-3 cannot be mapped, the default setting in machine data
20154: $MC_EXTERN_GCODE_RESET_VALUES and/or
20152: $MC_GCODE_RESET_VALUES
is activated.

Example: `ISO mode`

```
N5   G00 X100. Y100.
N10  G90          ;Activate G90 in ISO mode Group 3
                  ;In Siemens mode Group 14
N15  G290         ;Switch over to Siemens, G90 is active
N20  G91          ;Activate G91 in ISO mode Group 3
                  ;In Siemens mode Group 14
N25  G291         ;Switch over to ISO mode
N30  G291         ;G91 is active
```

2.2.1 G code display

In the G code display, the G codes for the currently active language are displayed. G290/G291 also causes the G code display to switch over.

Example:

The Siemens standard cycles are called up using some of the ISO Dialect mode G functions (e.g. G28). DISPLOF is programmed at the start of the cycle, with the result that the ISO Dialect G commands remain active for the display.

```
PROC CYCLE328 SAVE DISPLOF
N10 ...
...
N99 RET
```

Sequence:

- External main program calls Siemens shell cycle.
  Siemens mode is selected implicitly on the shell cycle call.
- DISPLOF freezes the block display at the call block;
  the G code display remains in external mode. This display is refreshed while the Siemens cycle is running.

2.2.2 Display of non-modal G codes

As of SW 6.4 the external non-modal G codes (group 18) will no longer be reset on block change if these G codes call up subprograms. The G codes remain visible on the display until the next jump out of this subprogram.

Switching to external language mode in the subprogram and programming another G code from group 18 overwrites the previous value and the new value is retained until the return jump.

Example:

Main program	Display group 18
`N05 G00 X0 Y0`	empty
`N08 G27 X10` –> calls Cycle328	empty
`N09 M0`	empty
`N40 M30`	empty

Subprogram Cycle328	Display group 18
`N100 G290`	G27
`N102 X=$C_X`	G27
`N103 M0`	G27
`N104 G291`	G27
`N105 G30 X10 Y12 Z13`	G30
`N120 M99`	G30

2.2.3 G code output to PLC (as from SW 6.4)

The behavior of G group transfer to PLC is described in machine data $MC_GCODE_GROUPS_TO_PLC_MODE.

The previous behavior was for the G group to be the array index of a 64 byte field (DBB 208 – DBB 271). That way, up to the 64th G group can be reached. Only the G groups of the standard or external language can be displayed.

The new behavior is for the data storage in the PLC to be up to 8 bytes (DBB 208 – DBB 215), i.e. up to 8 G groups can be output.

This method has the array index of machine data

22515: $MC_GCODE_GROUPS_TO_PLC[ ] or
22512: $MC_EXTERN_GCODE_GROUPS_TO_PLC[ ]

equal to the array index of the data storage in the PLC (DBB 208 – DBB215).
The G code group from MD $MC_GCODE_GROUPS_TO_PLC[ ] is output in DBB 208.

The advantage is that Siemens mode and ISO mode G codes can be output simultaneously.

Because only the G code of one language can be output in a DBB2xx, each index (0 –7) can only be set on one of the two machine data, and the value 0 must be entered in the other MD. Errors are signaled with Alarm 4045.

Example

$MC_GCODE_GROUPS_TO_PLC[0]=3
$MC_GCODE_GROUPS_TO_PLC[1]=0
$MC_GCODE_GROUPS_TO_PLC[2]=0
$MC_GCODE_GROUPS_TO_PLC[3]=0
$MC_GCODE_GROUPS_TO_PLC[4]=1
$MC_GCODE_GROUPS_TO_PLC[5]=2
$MC_GCODE_GROUPS_TO_PLC[6]=0
$MC_GCODE_GROUPS_TO_PLC[7]=0

$MC_EXTERN_GCODE_GROUPS_TO_PLC[0]=0
$MC_EXTERN_GCODE_GROUPS_TO_PLC[1]=3
$MC_EXTERN_GCODE_GROUPS_TO_PLC[2]=18
$MC_EXTERN_GCODE_GROUPS_TO_PLC[3]=1
$MC_EXTERN_GCODE_GROUPS_TO_PLC[4]=0
$MC_EXTERN_GCODE_GROUPS_TO_PLC[5]=0
$MC_EXTERN_GCODE_GROUPS_TO_PLC[6]=6
$MC_EXTERN_GCODE_GROUPS_TO_PLC[7]=31

The following G codes are then available on the PLC

DBB 208 = group 03 Siemens
DBB 209 = group 03 ISO dialect
DBB 210 = group 18 ISO dialect
DBB 211 = group 01 ISO dialect
DBB 212 = group 01 Siemens
DBB 213 = group 02 Siemens
DBB 214 = group 06 ISO dialect
DBB 215 = group 31 ISO dialect

Example of faulty configuration:

$MC_GCODE_GROUPS_TO_PLC[0]=3
$MC_GCODE_GROUPS_TO_PLC[1]=0
$MC_GCODE_GROUPS_TO_PLC[2]=0

$MC_EXTERN_GCODE_GROUPS_TO_PLC[0]=3 –>
Alarm 4045, channel K1 conflict between machine data
{S$MC_GCODE_GROUPS_TO_PLC} and machine data
{S$MC_EXTERN_GCODE_GROUPS_TO_PLC}
$MC_EXTERN_GCODE_GROUPS_TO_PLC[1]=0
$MC_EXTERN_GCODE_GROUPS_TO_PLC[2]=18

The method enables simultaneous display of G codes of standard mode and ISO dialect mode.

2.2.4 Zero offset

The zero offsets (ZO) of Siemens mode are shown in Fig. 2-1.

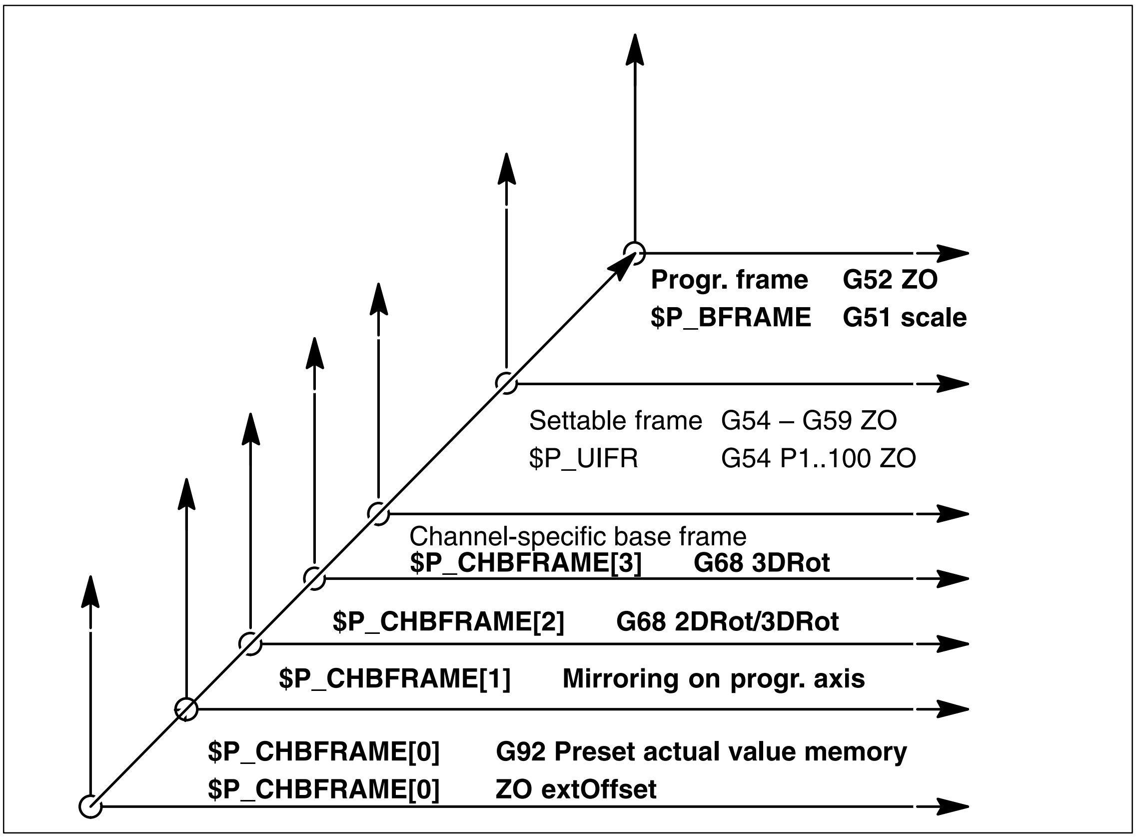


Fig. 2-1 Instantaneous mapping of the ISO functions onto the Siemens frames

The zero offsets that are available in ISO mode are mapped onto the existing Siemens frames. **No separate frames exist for ISO mode.** Active zero offsets are incorporated in both language modes.
Changes in ISO mode have an immediate effect in Siemens mode and vice-versa.

Zero offsets exist in both ISO Dialect T and ISO Dialect M:

- G52 is a programmable, additive ZO that remains active until the end of the program or a reset
- G54 to G59 are settable zero offsets
- G54 P1...P100 are additional settable zero offsets
- G54 P0 is an “external ZO” extOffset

G54.1

G54.1 Pxx is provided as an alternative notation to G54 Pxx. The functionality is identical. With G54.1 the P address must always be programmed in the block. If P is not programmed, alarm 12080 (syntax error) is produced.

Display of extended zero offset G54 Pxx

Previously, it was not possible to program G54.1 P.. in ISO dialect T. G code group 14 in ISO dialect T has now been extended with G code G54.1 and G54.1 is now displayed by default if P is programmed.

Previously, when programming G54 Pxx or G54.1 Pxx, G54.1 was displayed in the G code display in ISO dialect M.

MD $MC_EXTERN_FUNCTION_MASK bit 11 can now be used to activate the display of the programmed P after the point in the G code display.

Programmed	Bit 11 = 1	Bit 11 = 0
G54 P1	Display G54P1	G54.1
G54 P28	Display G54P28	G54.1
G54.1 P28	Display G54P28	G54.1
G54 P48	Display G54P48	G54.1
G54.1 P48	Display G54P48	G54.1

2.2.5 Writing a zero offset with G10

G10 can be used from the parts program to write the zero offsets.

```
G10 L2 P1...P6 X.. Y..     ; G54.. G59
G10 L20 P1...P100          ; Additional, settable ZO
G10 L2 P0                  External ZO extOffset
```

These zero offsets are mapped onto the same frames as the zero offsets that already exist in ISO Dialect M.

The G10 command is extended for **ISO dialect T**:

Writing of system data
G10 Pxx X Y Z ;writing of tool offset data

Depending on machine data $MC_EXTERN_FUNCTION_MASK, bit1, G10 Pxx is used to write either tool geometry or tool wear.

$MC_EXTERN_FUNCTION_MASK, bit1 = 0:
P > 100 write geometry values
P < 100 write wear values

$MC_EXTERN_FUNCTION_MASK, bit 1=1:
P > 10000 write geometry values
P < 10000 write wear values

2.2.6 Decimal point programming

There are two notations for the interpretation of programming values without a decimal point in ISO Dialect mode:

- **Pocket calculator type notation**
  Values without decimal points are interpreted as mm, inch or degrees.
- **Standard notation**
  Values without decimal points are multiplied by a conversion factor.

The setting is defined by MD 10884, see Chapter 4 “Startup”.

There are two different conversion factors, **IS-B** and **IS-C.** This evaluation refers to addresses X Y Z U V W A B C I J K Q R and F.

Example of linear axis in mm:

```
X 100.5    corresponds to value with decimal point: 100.5 mm
X 1000     pocket calculator type notation: 1000 mm
           standard notation:  IS-B: 1000* 0.001= 1 mm
                               IS-C: 1000* 0.0001 = 0.1 mm
```

ISO dialect Milling

Table 2-4 Different conversion factors for IS-B and IS-C

Address	Unit	IS-B	IS-C
Linear axis	mm inch	0.001 0.0001	0.0001 0.00001
Rotary axis	deg	0.001	0.0001
F feed G94 (mm/inch per min.)	mm inch	1 0.01	1 0.01
F feed G95 (mm/inch per min.)	mm inch	0.01 0.0001	0.01 0.0001
F thread pitch	mm inch	0.01 0.0001	0.01 0.0001
C chamfer	mm inch	0.001 0.0001	0.0001 0.00001
R radius, G10 toolcorr	mm inch	0.001 0.0001	0.0001 0.00001
Q	mm inch	0.001 0.0001	0.0001 0.00001
I, J, K interpolation parameters	mm inch	0.001 0.0001	0.0001 0.00001
G04 X or U	s	0.001	0.001
A contour angle	deg	0.001	0.0001
G74, G84 thread drilling cycles $MC_EXTERN_FUNCTION_MASK Bit8 = 0 F feedrate like G94, G95 Bit8 = 1 F thread pitch			

ISO dialect Turning

Table 2-5 Different conversion factors for IS-B and IS-C

Address	Unit	IS-B	IS-C
Linear axis	mm inch	0.001 0.0001	0.0001 0.00001
Rotary axis	deg	0.001	0.0001
F feed G94 (mm/inch per min.)	mm inch	1 0.01	1 0.01
F feed G95 (mm/inch per rev) $MC_EXTERN_FUNCTION_MASK Bit8 = 0 Bit8 = 1	 mm inch mm inch	 0.01 0.0001 0.0001 0.000001	 0.01 0.0001 0.0001 0.000001
F thread pitch	mm inch	0.0001 0.000001	0.0001 0.000001
C chamfer	mm inch	0.001 0.0001	0.0001 0.00001
R radius, G10 toolcorr	mm inch	0.001 0.0001	0.0001 0.00001

Table 2-5 Different conversion factors for IS-B and IS-C

Address	Unit	IS-B	IS-C
I, J, K interpolation parameters	mm inch	0.001 0.0001	0.0001 0.00001
G04 X or U		0.001	0.001
A contour angle		0.001	0.0001
G76, G78 thread drilling cycles $MC_EXTERN_FUNCTION_MASK Bit8 = 0 F feedrate like G94, G95 Bit8 = 1 F thread pitch			
G84, G88 thread drilling cycles $MC_EXTERN_FUNCTION_MASK			
Bit9 = 0 G95 F	mm inch	0.01 0.0001	0.01 0.0001
Bit8 = 1 G95 F	mm inch	0.0001 0.000001	0.0001 0.000001

2.2.7 Rapid lift with G10.6

G10.6 <AxisPosition> is used to activate a retraction position for the rapid lifting of a tool (e.g., in the event of a tool break). The retraction motion itself is started with a digital signal. The second NC fast input is used as the start signal.
Machine data $MN_EXTERN_INTERRUPT_NUM_RETRAC is used to select a different fast input (1 – 8).

In Siemens mode, the activation of the retraction motion comprises a number of part program commands.

```
N10 G10.6 X19.5 Y33.3
```

generates internally in the NCK

```
N10 SETINT (2) PRIO=1 CYCLE3106 LIFTFAST     ; Activate interrupt input
N30 LFPOS                                    ; Select lift mode
N40 POLF[X]=19.5  POLF[Y]=33.3               ; Program lift positions
                                             ; for x19.5 and y33.3
N70 POLFMASK(X, Y)                           ; Activate retraction
                                             ; of x and y axis
```

G10.6 is used to group these part program commands internally in a command set.

In order to activate an interrupt input (SETINT(2)), an interrupt program (ASUP) must also be defined. If one has not been programmed, the part program will not be able to continue as it will be interrupted with a reset alarm once the retraction motion is complete. The interrupt program (ASUP) CYCLE3106.spf is always used for fast retraction with G10.6. If the part program memory does not contain program CYCLE3106.spf, alarm 14011 “Program CYCLE3106 not available or not enabled for processing” is output in a part program set with G10.6.

The behavior of the control following fast retraction is specified in ASUP CYCLE3106.spf. If the axes and spindle are to be stopped following fast retraction, M0 and M5 must be programmed accordingly in CYCLE3106.spf.
If CYCLE3106.spf is a dummy program, which only contains M17, the part program will continue uninterrupted following fast retraction.

If G10.6 <AxisPosition> is programmed to activate fast retraction, when the input signal of the second NC fast input changes from 0 to 1, the motion currently in progress is interrupted and the position programmed in set G10.6 is approached at rapid traverse. The positions are approached absolutely or incrementally according to the program settings in set G10.6.

The function is deactivated with G10.6 (without positional data). Fast retraction by means of the input signal of the second NC fast input is disabled.

Siemens

To some extent, the fast retraction function with G10.6 can be achieved using function POLF[<AxisName>] = <RetractionPosition>. This function will also retract the tool to the programmed position. However, it does not support the remainder of the ISO dialect original functionality. If the interrupt point cannot be approached directly, obstructions must be bypassed manually.

References: /PGA/, Programming Guide Advanced,
Chapter “Extended Stop and Retract”

Restrictions

Only one axis can be programmed for fast retraction.

2.2.8 Multiple threads with G33

Syntax G33 X.. Z.. F.. Q.. is used to program multiple threads in ISO dialect T and M mode, whereby:

X.. Z.. = Thread end position
F.. = Lead
Q.. = Initial angle

Threads with offset slides are programmed by entering starting points, which are offset from one another, in set G33. The starting point offset is entered at address "Q" as an absolute angular position. The corresponding setting data ($SD_THREAD_START_ANGLE) is changed accordingly.

Example: Q45000 means: Start offset 45.000 degrees

Range of values: 0.0000 to 359.999 degrees

The initial angle must always be programmed as an integer value. The input resolution for angular data is 0.001 degrees.

Example:

```
N200 X50 Z80 G01 F.8 G95 S500 M3
N300 G33 Z40 F2 Q180000
```

This produces a thread with a lead of 2 mm and a starting point offset of 180 degrees.

2.2.9 Threads with variable lead (G34)

Syntax G34 X.. Z.. F.. K.. is used to program threads with variable lead in ISO dialect T and M mode, whereby

X.. Z.. = Thread end position
F.. = Lead
K.. = Lead increase (positive value)/
lead decrease (negative value)

G34 is used to increment or decrement the lead by the value programmed at address K on each spindle revolution.

Example:

```
N200 X50 Z80 G01 F.8 G95 S500 M3
N300 G91 G34 Z25.5 F2 K0.1
```

The programmed distance of 25.5 mm corresponds to 10 spindle revolutions.

2.2.10 Dwell time in spindle revolutions G04

MD 20734: $MC_EXTERN_FUNCTION_MASK, bit 2 defines how the programmed dwell time will be interpreted in a G04 block. The hold time can be programmed using G04 X U or P.

Bit 2 = 0:	Dwell time is always interpreted in [s].
Bit 2 = 1:	If G95 is active, dwell time is interpreted in spindle revolutions.

In the case of standard notation, X and U values without a decimal point are converted into internal units depending on IS-B or IS-C. P is always interpreted in internal units.

Example:

`N5 G95 G04 X1000`	Standard notation 1000 * 0.001 = 1 spindle revolution pocket calculator notation: 1000 spindle revolutions

2.2.11 Scaling and mirroring: G51, G51.1 (ISO Dialect M)

G51 selects scaling and mirroring, G51.1.
There are two scaling modes:

- Axial scaling with parameters I, J, K

If I, J, K is not programmed in the G51 block, the default value from the setting data is effective.
Negative axial scaling factors have the additional effect of mirroring.

- Scaling in all axes with scale factor P

If P is not programmed in the G51 block, the default value from the setting data is effective. Negative P values are not possible.
The scale factors are multiplied by either 0.001 or 0.00001.

Note

If a factor other than “1” is programmed for parameters I, J, K or if the address is missing (default value is active for I, J, K), the contour is also scaled.

Example

`00512 (parts program)`	
`N10 G17 G90 G00 X0 Y0`	Approach start position
`N30 G90 G01 G94 F6000`	
`N32 M98 P0513`	1) Contour programmed as in the subprogram
`N34 G51 X0. Y0. I-1000 J1000`	2) Mirror contour around X

```
N36 M98 P0513
N38 G51 X0. Y0. I-1000 J-1000       3) Mirror contour around X and Y
N40 M98 P0513
N42 G51 X0. Y0. I1000 J-1000        4) Mirror contour around Y
N44 M98 P0513
N46 G50                             Deselect scaling and mirroring
N50 G00 X0 Y0
N60 M30

O0513 (subprogram)
N10 G90 X10. Y10.
N20 X50
N30 Y50
N40 X10. Y10.
N50 M99
```

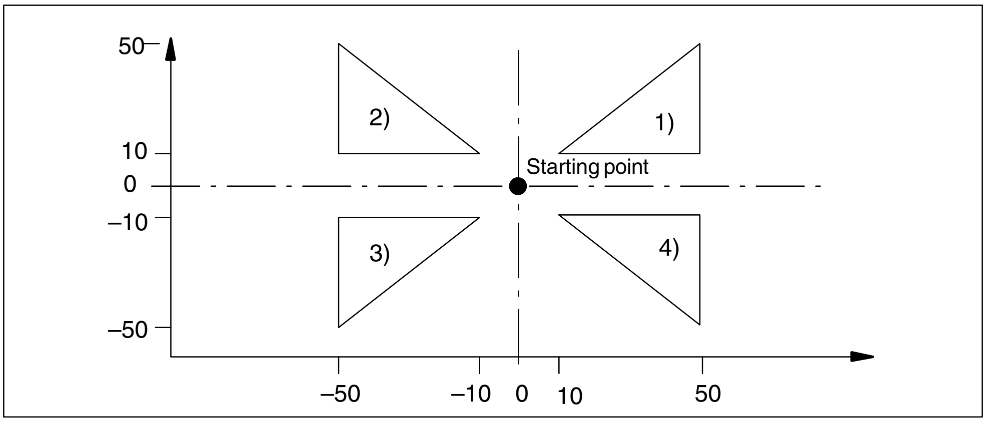


Fig. 2-2 Scaling and mirroring

System parameter settings for the scaling and mirroring example:

MD 22910 $MC_WEIGHTING_FACTOR_FOR_SCALE = 0
MD 22914 $MC_AXES_SCALE_ENABLE = 1
MD 10884 $MN_EXTERN_FLOATINGPOINT_PROG = 0
MD 10886 $MN_EXTERN_INCREMENT_SYSTEM = 0

Axial scaling is not possible when MD $MC_AXES_SCALE_ENABLE = 0.

The reference point during scaling is always the workpiece zero; it is not possible to program a reference point.

Mirroring

G51.1 selects mirroring.
Mirroring is performed around a mirror axis that runs parallel to X, Y or Z and whose position is programmed with X, Y or Z. G51.1 X0 is used to mirror about the X axis and G51.1 X10 is used to mirror about an axis that runs parallel to the X axis at a distance of 10 mm.

All axes in the channel and not just the geometry axes can be mirrored.

G51.1 functions additively, i.e. following N5 G51.1 X10 and N10 G51.1 Y10, mirroring in X and V is active.

Example: G51.1 X80.
Mirroring is performed around a mirror axis that runs parallel to Y and that crosses the X axis at position 80.

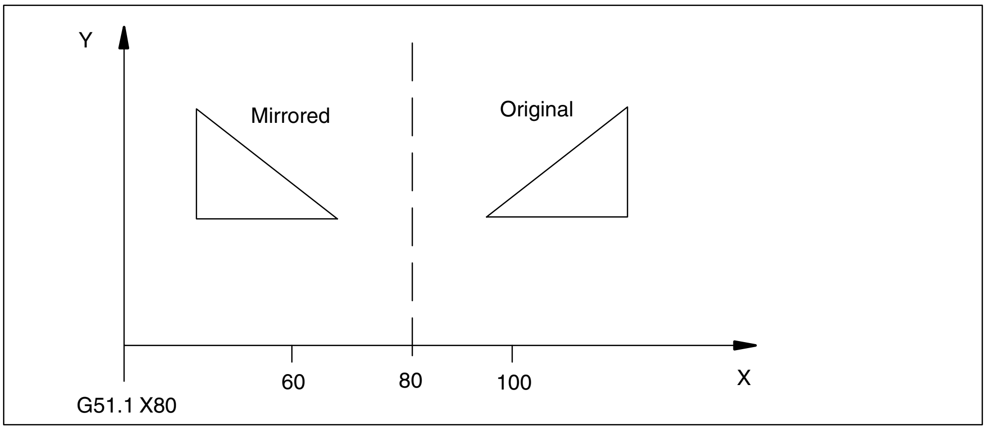


Fig. 2-3 Mirroring around a mirror axis parallel to Y

If the standard notation is active (see Subsection 2.2.6), the axis positions without a decimal point are interpreted in internal units.

Mirroring is deselected with G50.1 X0 Y0. It can also be deselected for individual axes. Following G50.1 X0, mirroring is only deselected for the X axis; mirroring around all other axes remains active.

G51.1 and G50.1 must be in a block of their own.
G51.1 is mapped onto channel-specific base frame [1]. For this purpose, MD 28081 $MC_MM_NUM_BASE_FRAMES >=2 must be set.
When base frame[1] is changed in Siemens mode, it directly affects the function in ISO mode.
If the frame is deleted in every frame component, this corresponds to a G50.1 X0 Y0.. in all axes.

G51.1 is deselected on a Reset.

2.2.12 2D/3D rotation G68 / G69 (ISO Dialect M)

2D rotation

The coordinate system is rotated about the vertical axis of the selected plane.

Programming

G68 X.. Y.. R..

X.. Y..: Coordinates of the pivot point related to the current workpiece zero. If a coordinate is not programmed, the pivot point is taken from the actual value memory. The value is always interpreted as an absolute value.

R: The angle of rotation is interpreted as an absolute or incremental value depending on G90/G91. If an angle is not programmed, the angle from setting data $SA_DEFAULT_ROT_FACTOR_R is active. G68 must be in a block of its own.

G69 Rotation Off; Additional codes can be programmed in this block.

G68 is mapped onto channel-specific base frame 2. For this purpose, machine data MD 28081: $MC_MM_NUM_BASE_FRAMES >= 3 must be set.

A programmed angle R is not entered in setting data 42150: $SA_DEFAULT_ROT_FACTOR_R. This setting data can only be written manually and is effective provided that no R has been programmed in the G68 block.

3D rotation

G code G68 has been expanded for 3D rotation.

Programming

G68 X.. Y.. Z.. I.. J.. K.. R..

X.. Y.. Z..: Coordinates of the pivot point related to the current workpiece zero. If a coordinate is not programmed the pivot point is at the workpiece zero. The value is always interpreted as an absolute value. The coordinates of the pivot point act likea zero offset. A G90/91 in the block has no effect on the G68 command.

I.. J.. K..: Vector in the pivot point. The coordinate system is rotated about this vector by the angle R.

R: Angle of rotation, always interpreted as an absolute value. If an angle is not programmed, the angle from setting data 42150 $SA_DEFAULT_ROT_FACTOR_R is active. G68 must be in a block of its own.

The distinction between 2D and 3D rotation is determined solely by programming the vector I, J, K. If no vector exists in the block, G68 2DRot is selected. If a vector exists in the block, G68 3DRot is selected.

If a vector of length 0 (I0, Y0, K0) is programmed, the alarm 12560 “Programmed value lies outside the permissible limits” is output.

With G68, two rotations can be connected in series. If a G68 is not already active in a block containing G68, the rotation is written into channel-specific base frame 2. If G68 is already active, the rotation is written in channel-specific base frame 3. This means that both rotations are activated in sequence.

With G69, 3D rotation is terminated. If two rotations are active, they are both deactivated with G69. G69 does not have to be in a block of its own.

2.2.13 Double-slide or double-turret machining G68 / G69

Function G68/G69 is used to control the two-sided machining of turned parts (both machining with a double slide in two channels and machining in one channel with two tools with a fixed connection at a distance x).

MD $MN_EXTERN_DOUBLE_TURRET_ON is used to define whether machining in the two channels is synchronized (= FALSE) or if one of two fixed-connection tools is used alternately for machining (= TRUE).

On fixed-connection tools, G68 is used to activate the distance x entered in MD 42162: $SC_EXTERN_DOUBLE_TURRET_DIST as an additive zero offset in the X axis. As the second tool machines the opposite side of the turned part, G68 also activates mirroring about the Z axis (directional reversal of the X axis). The next set with axis motions activates the zero offset and mirroring for the second tool.

G69 disables zero offset and machining continues with the first tool.

G68 and G69 must only be programmed in the set.

The sign of the offset must be taken into account for tool length offset in the X axis for the second tool. The sign must be entered as if the X axis was not mirrored or setting data $SC_MIRROR_TOOL_LENGTH (mirror tool length offset) and $SC_MIRROR_TOOL_WEAR (mirror tool length offset wear values) must be set.

Machine data $MN_MIRROR_REF_AX must = 0 or = 1 in order to always mirror the X or first axis.

Programming G68 when G68 is already active will read over the G function. The same is true of G69.

Double-turret head: $MN_EXTERN_DOUBLE_TURRET_ON = TRUE

The example below illustrates machining with two fixed-connection tools. In order for the function to be effective, machine data $MN_EXTERN_ DOUBLE_TURRET_ON must be set to TRUE.

If setting data 42162: $SC_EXTERN_DOUBLE_TURRET_DIST = 0, alarm “12728 Distance for double turret not set” will be output.

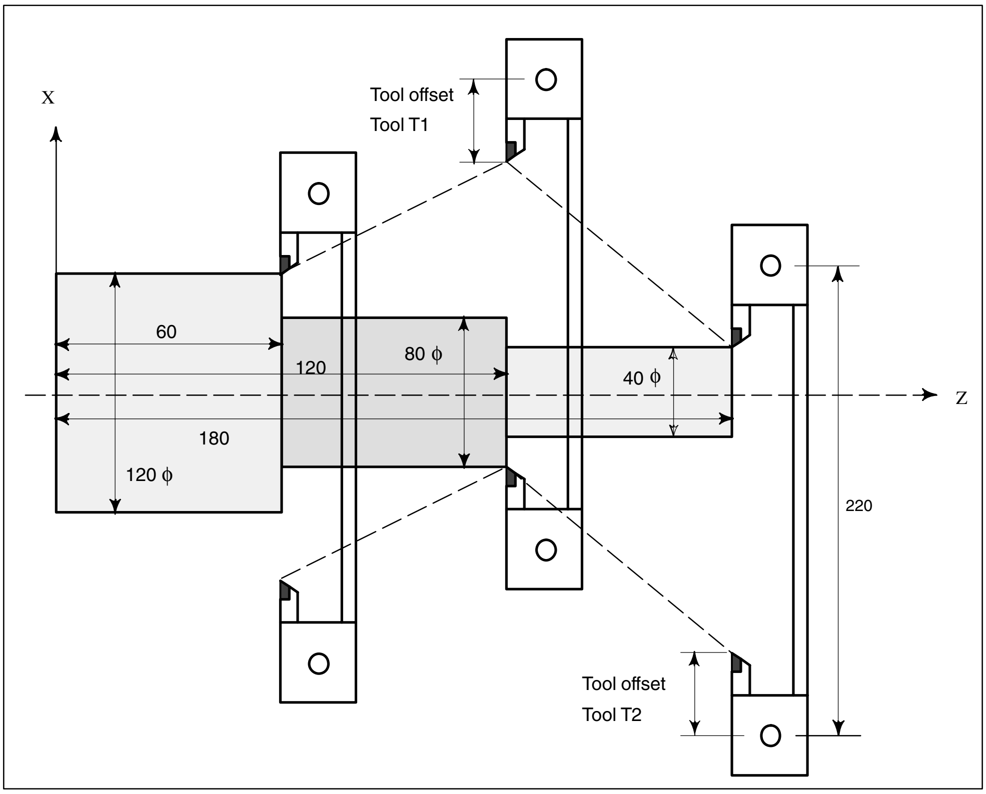


Fig. 2-4 Machining with 2 fixed-connection tools

Example:

```
N100 X40. Z180. G1 F1 G95 S1000 M3 T1
N110 G68  ; Activate mirroring about Z and additive zero offset (220 mm)
N120 X80. Z120. T2
N130 G69  ; Deactivate mirroring and additive zero offset
N140 X120. Z60 T1
```

Double-slide machining: $MN_EXTERN_DOUBLE_TURRET_ON = FALSE

Setting $MN_EXTERN_DOUBLE_TURRET_ON to FALSE activates channel synchronization with G68. If G68 is programmed in one channel, machining will cease until G68 is detected in the second channel. This function is used to synchronize the first and second channels. No other synchronizations are performed. In order for both tools to be synchronized during subsequent machining, the motions and feeds programmed in the two channels must be identical.

Wait mark 1 is used for G68 and wait mark 2 for G69 in order to synchronize the first 2 channels. Therefore, the first two M functions may not be used simultaneously for channel synchronization in the same part program (see Subsection 4.1.10).

G68 is only effective in the first two channels. If G68 is programmed in another channel and machine data $MN_EXTERN_DOUBLE_TURRET_ON = FALSE, G68 is read over.

The function is used to produce thin turned parts. The two tools should therefore execute the same motion on the respective opposite side of the turned part, mirroring about the Z axis. For this purpose, the same traversing motions and feeds must be programmed in both channels.

Example of synchronous machining with two channels.

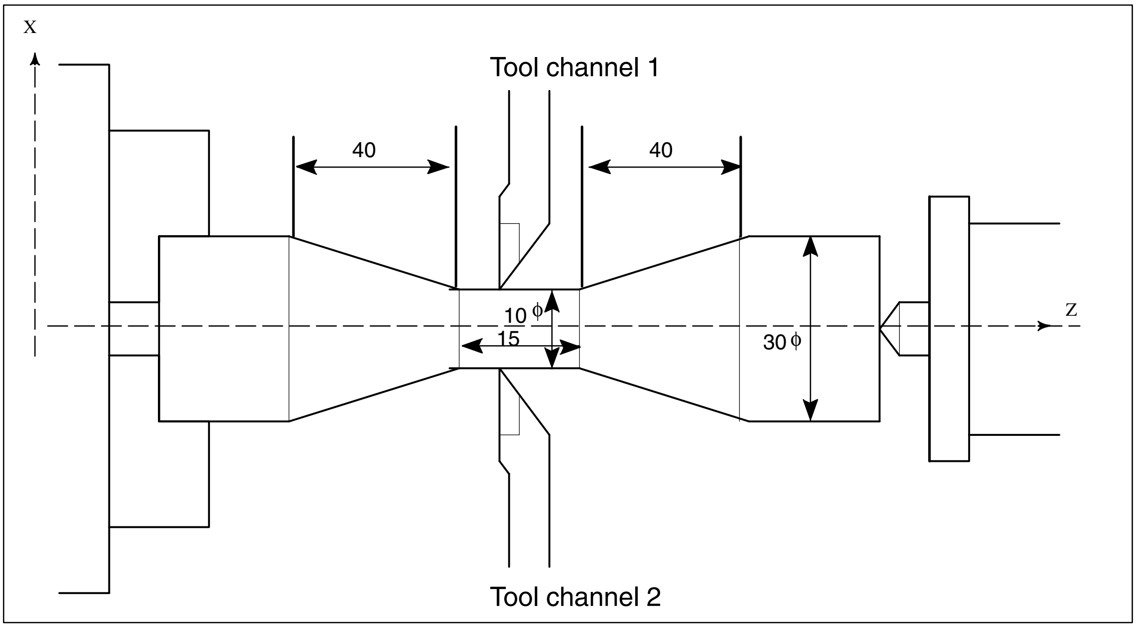


Fig. 2-5 Synchronous machining with 2 channels

Example:
Channel 1:

```
N10 ....
– " –
N1000 G68                 ; Start synchronization
N1010 G01 X30 Z120 G95 F1.2 S1000 M3
```

```
N1020 X15 Z80
N1030 Z65
N1040 Z25 X40
N1050 G69                ; Synchronization OFF
```

Channel 2:

```
N10 .....
– " –
N2000 G68                ; Start synchronization
N2010 X30 Z120 G01 G95 F1.2 S1000 M3
N2020 X15 Z80
N2030 Z65
N2040 X40 Z25
N2050 G69                ; Synchronization OFF
```

In ISO dialect original, channel synchronization will also be performed if G68 is active.

2.2.14 Polar coordinates: G15 (ISO Dialect M)

In ISO Dialect mode, NC program sections programmed with polar coordinates must commence with start command G16. Until the end command G15 is reached, the coordinates of the end points are interpreted as the polar coordinate values for radius and angle in the current plane.

The first axis of the plane is the polar radius, the second axis is the polar angle, i.e. X is the radius and Y is the angle for G17.

After G16 a new pole is set in each block up to G15, with the result as follows for G17:

- G90 X — The pole is at the workpiece zero
- G91 X — The pole is at the current position
- No X in the block — The pole is at the workpiece zero

If the pole is moved from the current position to the workpiece zero, the radius is calculated as the distance from the current position to the workpiece zero.

Example:

`N5 G17 G90 X0 Y0`	
`N10 G16 X100. Y45.`	Polar coordinates ON, pole is the workpiece zero, Position X 70,711 Y 70,711 in the Cartesian coordinate system
`N15 G91 X100 G90 Y0`	Pole is the current position, position X 170,711 Y 70,711
`N20 Y90.`	No X in block, pole is at workpiece zero, Radius = SORT(X*X +Y*Y) = 184,776

The polar radius is always traversed as an absolute distance; the polar angle can be interpreted as an absolute or incremental value.

Programmed angle

In the case of active polar coordinate programming, the programmed angle can be read via the system variable $P_AP.
This variable is inserted in the shell cycle. Before the new pole is set, with incremental programming, the angle must be stored because the angle will be deleted.

Polar programming is terminated by G15. The polar radius is set to 0.

2.2.15 Polar coordinate interpolation G12.1 / G13.1 (G112/G113)

G12.1 and G13.1 are used to switch on and switch off an interpolation in the processing level between an axis of rotation and a linear axis. A second linear axis passes vertically through this plane. This function corresponds to the Transmit function in Siemens mode. In Siemens mode, two Transmit transformations can be parameterized. For G12.1 the first TRANSMIT data block is always the one which must correspond to the second transformation record. For a detailed description of the TRANSMIT function please refer to the following documentation:

/FB2/ SINUMERIK 840D/810D(CCU2)
Description of Functions, Extended Functions, Chapter M1 and
/PGA/ SINUMERIK 840D/810D
Programming Guide, Advanced, Chapter “Transformations”

Example:

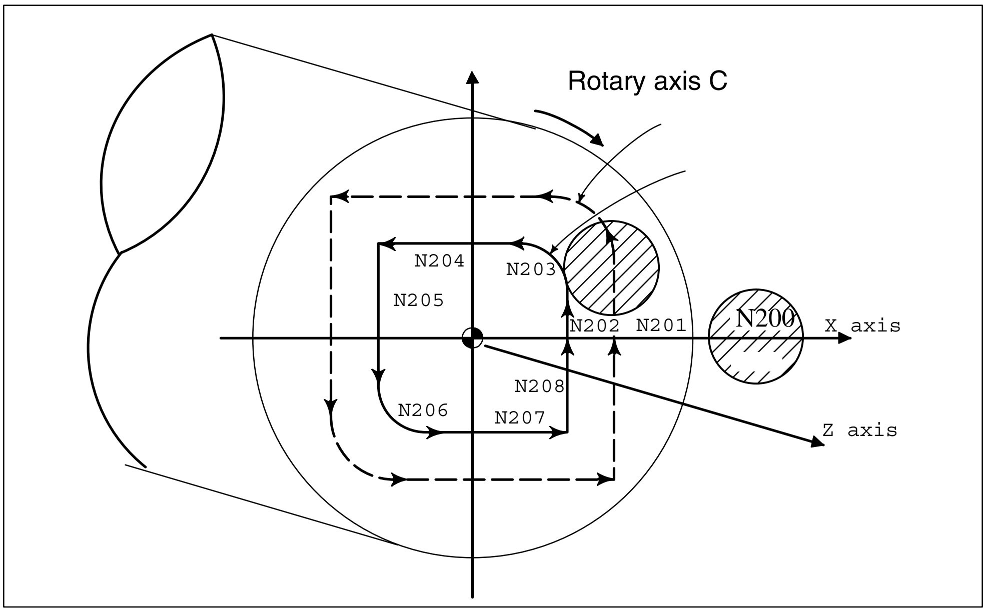


Fig. 2-6 Example of polar coordinate interpolation

```
O0001
N010 T0101
N0100 G90 G00 X60.0 C0 Z..
N0200 G12.1                    ;TRANSMIT selection
N0201 G42 G01 X20.0 F1000
N0202 C10.0 ;
N0203 G03 X10.0 C20.0 R10.0
N0204 G01 X-20.0
N0205 C-10.0
N0206 G03 X-10.0 C-20.0 I10.0 J0
N0207 G01 X20.0
N0208 C0
N0209 G40 X60.0
N0210 G13.1                    ;TRANSMIT deselection
N0300 Z..
N0400 X.. C..
N0900 M30
```

Note

Geo axis exchange (parallel axes with G17 (g18, G19)) must not be active.

2.2.16 Cylindrical interpolation G07.1 (G107)

Function G07.1 (cylindrical interpolation) can be used to mill any kind of grooving on cylindrical bodies. The path of the grooving is programmed by reference to the developed, level surface of the cylinder barrel. Cylindrical interpolation is started in function G07.1 by specifying the cylindrical radius G07.1 C<cylindrical radius> and ended with G07.1 C0 (radius = 0).

The function is mapped internally onto the Siemens functionality TRACYL. In ISO Dialect mode, G07.1 always activates the first TRACYL transformation and the first transformation record. The second TRACYL function cannot be activated in ISO Dialect mode. For a detailed description and the parameter setting for the first TRACYL function, please refer to the following documentation:

/FB2/ SINUMERIK 840D/810D(CCU2)
Description of Functions, Extended Functions, Chapter M1 and

/PGA/ SINUMERIK 840D/810D
Programming Guide, Advanced, Chapter “Transformations”

Restrictions

In Siemens mode the axis of rotation for cylindrical interpolation must be defined in machine data.
In ISO Dialect mode the axis of rotation for cylindrical interpolation is defined by programming G07.1 <axis of rotation>... .

Example

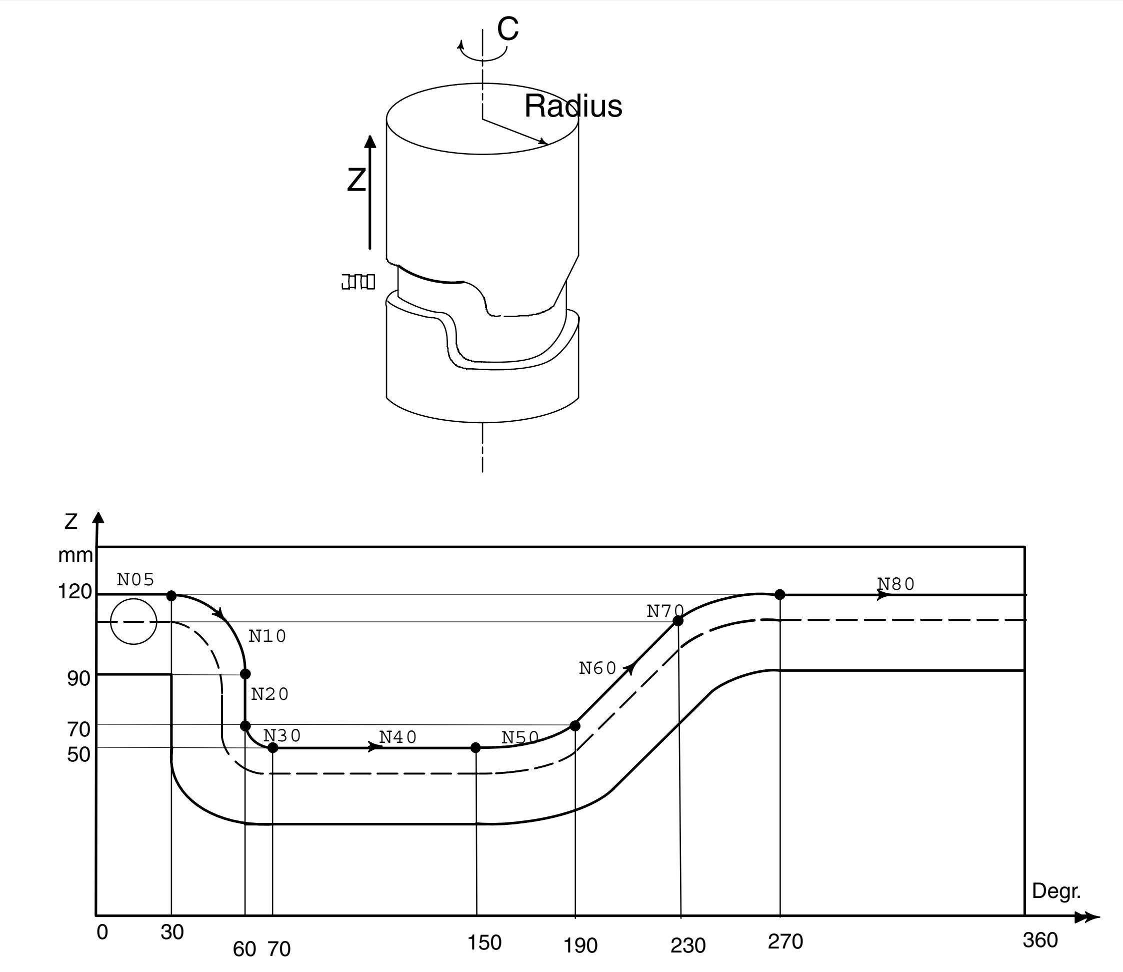


Fig. 2-7 Example of cylindrical interpolation G07.1

Programming example in ISO Dialect mode:

```
%0001
N05 G00 G90 Z100.0 C0
N10 G01 G91 G18 Z0 C0
N20 G07.1 C57299             ;Select cylindrical interpolation with radius
                             ; 57.299 mm
N30 G90 G01 G42 Z120.0 D01 F250
N40 C30.0
```

```
N50 G02 Z90.0 C60.0 R30.0
N60 G01 Z70.0
N70 G03 Z60.0 C70.0 R10.0
N80 G01 C150.0
N90 G03 Z70.0 C190.0 R75.0
N100 G01 Z110.0 C230.0
N110 G02 Z120.0 C270.0 R75.0
N120 G01 C360.0
N130 G40 Z100.0
N140 G07.1 C0                 ;Deselect cylindrical interpolation
N150 M30 ;
```

Programming example in Siemens mode: The Y axis is assigned to the axis of rotation as a linear axis.

```
%0001
N05 G00 G90 Z100 C0
N10 G01 G91 G18 Z0 C0;
N20 TRACYL(114.598)           ;Select cylindrical interpolation with
                              ; radius 57.299 mm
N30 G90 G01 G42 Z120 D01 F250
N40 Y30
N50 G02 Z90 Y60 RND=30
N60 G01 Z70
N70 G03 Z60.0 Y70 RND=10
N80 G01 Y150
N90 G03 Z70 Y190 RND=75
N100 G01 Z110 Y230
N110 G02 Z120 Y270 RND=75
N120 G01 Y360
N130 G40 Z100
N140 TRAFOOF                  ;Deselect cylindrical interpolation
N150 M30 ;
```

2.2.17 Interrupt program with M96 / M97 (ASUB)

M96

A subprogram can be defined as an interrupt routine with M96 P <program number>.

This program is started by an external signal. The first high-speed NC input of the 8 inputs available in Siemens mode is always used to start the interrupt routine. Machine data $MN_EXTERN_INTERRUPT_NUM_ASUP lets you select an other fast input (1 – 8).

The function is mapped onto standard syntax**:** SETINT(x) <CYCLE396> [PRIO=1].

In shell cycle CYCLE396, the interrupt program programmed with Pxxxx is called in ISO mode. The program number is in $C_PI. At the end of the shell cycle, machine data
10808: $MN_EXTERN_INTERRUPT_BITS_M96 BIT1 is evaluated, resulting either in positioning at the interruption point with REPOSA or in continuation with the next block. The new cycle variable $C_PI contains the value programmed with “P” without leading zeroes. These must be added to fill out to four digits in the shell cycle before the subprogram is called.

Example: N0020 M96 P5

Call in shell cycle
progName = “000” << $C_PI
ISOCALLprogName

See treatment of 8-digit program numbers, if MD $MC_EXTERN_FUNCTION_MASK, bit 6 is set.

M97

M97 is used to suppress starting of the interrupt routine. The interrupt routine can then only be started by the external signal following activation with M96.

This corresponds to Standard syntax**:** ENABLE(x).

x = content of $MN_EXTERN_INTERRUPT_NUM_ASUP

If the interrupt program programmed with M96 Pxx is called up directly with the interrupt signal (without the intermediate step with CYCLE396), machine data 20734: $MC_EXTERN_FUNCTION_MASK BIT10 must be set. The subprogram programmed with Pxx is then called on a 0 –> 1 signal transition in Siemens mode.

The M function numbers for the interrupt function are set via machine data. With machine data 10804: $MN_EXTERN_M_NO_SET_INT, the M number is used to activate an interrupt routine and with MD 10806: $MN_EXTERN_M_NO_DISABLE_INT the M number is used to suppress an interrupt routine.

Only non-standard M functions are permitted to be set. M functions M96 and M97 are set as defaults. To activate the function, bit 0 must be set in machine data 10808: $MN_EXTERN_INTERRUPT_BITS_M96. These M functions will not be output to the PLC in this case. If bit 0 is not set, the M functions will be interpreted as conventional auxiliary functions.

On completion of the “Interrupt” program, the end position of the parts program block that follows the interruption block is approached. If processing of the parts program has to continue starting from the interruption point, there must be a REPOS instruction at the end of the “Interrupt” program, e.g. REPOSA.
For this purpose the interrupt program must be written in Siemens mode.

The M functions for activating and deactivating an interrupt program must be in a block of their own. If further addresses other than “M” and “P” are programmed in the block, alarm 12080 (syntax error) is output.

Note about machining cycles

For ISO dialect original, you can set whether a machining cycle will be interrupted by an interrupt routine immediately or not until the end. The shell cycles must evaluate machine data
10808: $MN_INTERRUPT_BITS_M96 bit 3 for that purpose. If bit=1, the interrupt must be disabled at the beginning of the cycle with DISABLE(1) and reactivated at the end of the cycle with ENABLE(1) to avoid interrupting the machining cycle. Because the interrupt program is only started on a 0/1 signal transition, the interrupt input must be monitored with a disabled interrupt during the cycle runtime with a synchronized action in the shell cycle. If the interrupt signal switches from 0 to 1, the interrupt signal after the ENABLE(1) must be set once again at the end of the shell cycle, so that the interrupt program will then start. To permit writing to the interrupt input in the shell cycle, the machine data
10361: $MN_FASTO_DIG_SHORT_CIRCUIT[1] must be parameterized.

Machine data

MD $MN_EXTERN_INTERRUPT_BITS_M96:

Bit 0: = 0: Interrupt program is not possible, M96/M97 are conventional M functions
= 1: Activation of an interrupt program with M96/M97 permitted

Bit 1: = 0: Execution of parts program continues from the final position of the next block after the interruption block
= 1: Continue parts program as from interruption position **(evaluated in interrupt program (ASUB), return with/without REPOSL)**

Bit 2: = 0: The interrupt signal interrupts the current block immediately and starts the interrupt routine
= 1: The interrupt routine is not started until the block has been completed.

Bit 3: = 0: The machining cycle is interrupted on an interrupt signal
= 1: The interrupt program is not started until the machining cycle has been completed.
(evaluated in the shell cycles)

Bit 3 must be evaluated in the shell cycles and the cycle sequence must be adapted accordingly.

Bit 1 must be evaluated in the interrupt program. If bit 1 = TRUE, on completion of the program, REPOSL must be used to reposition at the interruption point.

Example:

```
N1000 M96 P1234       ; Activate ASUB 1234.spf in the case of a rising
                      ; edge on the first high-speed input, program 1234.spf
                      ; is activated
          "
          "
N3000 M97             ; Deactivate the ASUB
```

Rapid lifting (LIFTFAST) is not performed before the interrupt program is called. On the rising flank of the interrupt signal, depending on machine data MD 10808: $MN_EXTERN_INTERRUPT_BITS_M96, the interrupt program is started immediately.

Limitations in Siemens mode

The interrupt routine is handled like a conventional subprogram. This means that in order to execute the interrupt routine, at least one subprogram level must be free. (12 program levels are available in Siemens mode, there are 5 in ISO Dialect mode.)

The interrupt routine is only started on a signal transition of the interrupt signal from 0 to 1. If the interrupt signal remains permanently set to 1, the interrupt routine will not be restarted.

Limitations in ISO Dialect mode

One program level is reserved for the interrupt routine so that all permissible program levels can be reserved before the interrupt program is called.

Depending on the machine data, the interrupt program will also be started when the signal is permanently on.

2.2.18 Comments

In ISO dialect mode, round brackets are interpreted as comment characters.
In Siemens mode, ";" is interpreted as a comment. To simplify matters, ";" is also interpreted as a comment in ISO dialect model.
If the comment start character "(" is used again within a comment, the comment will not be terminated until all open brackets have been closed again.

Example:

N5 (comment) X100 Y100
N10 (comment(comment)) X100 Y100
N15 (comment(comment) X100) Y100

In blocks N5 and N10 X100 Y100 is executed, in block N15 only Y100, as the first bracket is closed only after X100. Everything up to this position is interpreted as a comment.

2.2.19 Block skip

The skip character "/" can be anywhere within the block, even in the middle. If the programmed skip level is active at the moment of compiling, the block will not be compiled from this position to the end of the block. An active skip level therefore has the same effect as an end of block.

Example:

N5 G00 X100. /3 YY100	—> Alarm 12080,
N5 G00 X100. /3 YY100	—> No alarm when skip level 3 is active

Skip characters within a comment are not interpreted as skip characters.

Example:

N5 G00 X100. (/3 part1) Y100 ;even when skip level 3 is active, the
;Y axis will be traversed

The skip level can be /1 to /9. Skip values <1 >9 give rise to alarm 14060
The function is mapped onto the existing Siemens skip levels. In contrast to ISO Dialect Original, / and /1 are separate skip levels and therefore have to be activated separately.

2.2.20 Auxiliary function output

M

ISO Dialect mode

M functions are output to the PLC as auxiliary functions. Only M98 and M99 are exceptions. All other predefined M functions are transferred to the PLC as auxiliary functions.
The following are predefined M functions:
M17, M40, M41, M42, M43, M44, M45, M70, M96, M97, M98, M99.

Spindle axis changeover using M29

In ISO Dialect mode the spindle is switched to axis operation with the aid of M29. The M function number can also be set variably with machine data.
MD 20095 $MC_EXTERN_RIGID_TAPPING_M_NR is used to preset the M function number. The machine data is only effective in external language mode and is initialized with M29. It may only be assigned M function numbers which are not used as default M functions. M function numbers M0-M5, M30, M98, M99 are not allowed.
The same function is executed in Siemens mode with M70.
MD 20094 $MC_SPIND_RIGID_TAPPING_M_NR is used to preset the M function number. The machine data is only effective in Siemens mode and is initialized with M70. This allows an M function other than M70 to be used for the spindle switchover in Siemens mode. The machine data may only be assigned M function numbers which are not used as default M functions. The following are not allowed:
M0–M5, M17, M19, M30, M40–M45.

H

All H functions are output to the PLC as auxiliary functions with ISO Dialect M. In ISO Dialect T, G code system A, H is the incremental distance of the 4th axis provided that a 4th axis exists.

T

T functions are output to the PLC as auxiliary functions. T has the additional meaning of a tool selection.

D

Die The D function is output to the PLC as an auxiliary function. With ISO Dialect M, tool length compensation is activated with address D.

B

If B is not an axis, the B function is output to the PLC as an auxiliary function with address extension H1=.
Example: B1234 is output as H1=1234.

2.2.21 Align first reference point: G28

CYCLE328 is called up automatically when ISO Dialect command “G28 <Axis>” is read in. <Axis> specifies the intermediate position (incremental or absolute) via which the reference point is to be approached. The intermediate position and the reference position are then approached in positioning mode.

The cycle is only valid for the axes supported by ISO Dialect:

- ISO Dialect M: X, Y, Z (A, B, C, U, V, W)
- ISO Dialect T: X, Z, Y (C)

The cycle always runs with radius programming (DIAMOF). When the cycle is terminated, the G commands that were active before the cycle was called are effective again.

Before the 1st reference point is approached, various machine data must be set, see Chapter 4 “Start-Up”.

2.2.22 Enable/disable feed-forward control using G08 P..

Feed-forward control reduces speed-related overtravel during contouring to virtually nil. Traversing with feed-forward control enables higher contouring precision and thus better finished results.

Note

Machine data is used to define the type of feed-forward control and which path axes are to be traversed under pilot actuation.
Default: Speed-dependent feed-forward control.
Option: Acceleration-dependent feed-forward control.

Example:

```
N0010 G08 P1                ; Enable feed-forward control
N0020 G1 X10  Y50  F900
N0030 G1 X10  Y50  F900
N1000 G08 P0                ; Disable feed-forward control
```

If G08 is programmed without “P”, alarm 12470 is produced.

To make it more convenient to use G08 P1 to activate other functions such as SOFT, BRISK etc., G08 P.. is used to call the CYCLE308.spf cycle.

G08 P1 has to be in a block of its own.

2.2.23 Compressor in ISO dialect mode

The commands COMPON, COMPCURV, COMPCAD are commands in the Siemens language and activate a compressor function grouping several linear blocks to form one machining section.
It is now possible to compress linear blocks, too, in ISO dialect mode with this function, if this function is activated in Siemens mode.
The blocks must consist of only the following commands:

- Block number
- G01, modal or in the block
- Axis assignments
- Feedrate
- Comments

If a block contains other commands (e.g. aux. functions, other G codes, etc.), compression is not performed.
Value assignments with $x for G, axes, and feedrate are possible, as is the Skip function.

Example: These blocks are compressed

```
N5   G290
N10  COMPON
N15  G291
N20  G01 X100. Y100. F1000
N25  X100 Y100 F$3
N30  X$3 /1 Y100
N35  X100 (axis 1)
```

These blocks are **not** compressed

```
N5   G290
N10  COMPON
N20  G291
N25  G01 X100 G17    ;G17
N30  X100 M22        ;aux. function in the block
N35 X100 S200        ;spindle speed in the block
```

2.2.24 Automatic corner override G62

At inside corners with active tool radius compensation it is often better to reduce the feedrate.

G62 only acts at inside corners with active tool radius compensation and active continuous-path operation. It only takes account of corners whose inside angle is smaller than $SC_CORNER_SLOWDOWN_CRIT. The inside angle is determined from the bend in the contour.

The feedrate is reduced by factor $SC_CORNER_SLOWDOWN_OVR:

traveled feedrate =
F * $SC_CORNER_SLOWDOWN_OVR * feedrate override.

The feedrate override is now composed of the multiplied feedrate override from the machine control panel and the override from synchronized actions.

The feedrate reduction is started at distance 42520: $SC_CORNER_SLOW-DOWN_START before the corner. It ends at distance 42522: $SC_COR-NER_SLOWDOWN_END after the corner (see Fig. 2-8). An appropriate path is used at curved contours.

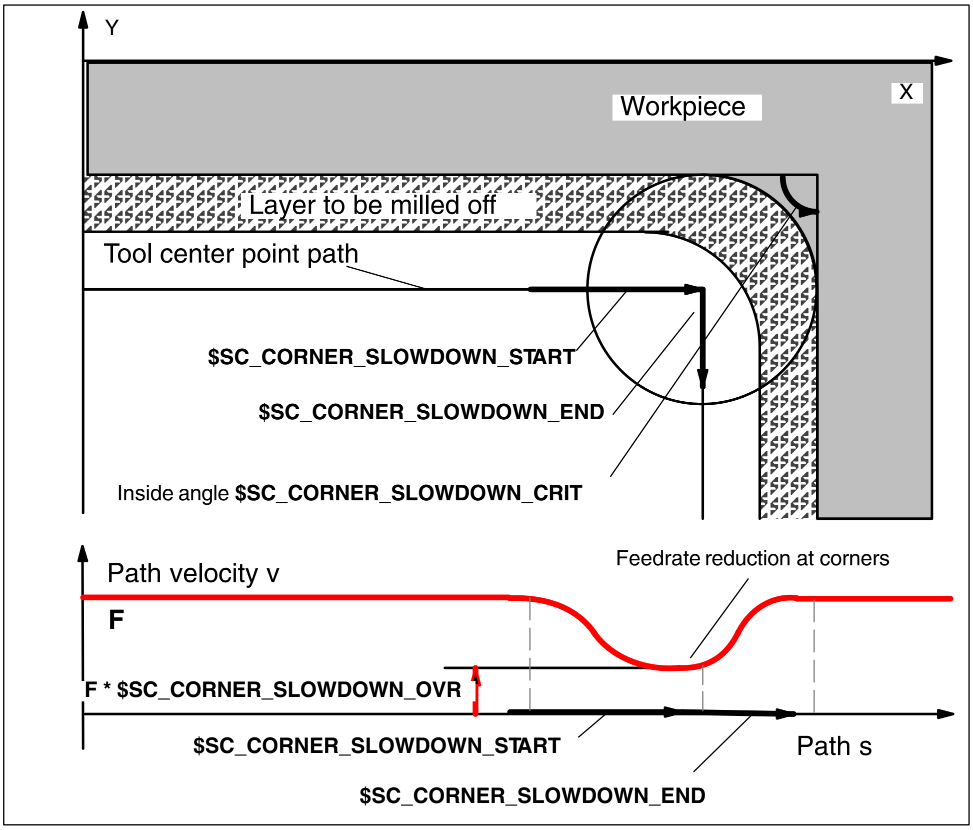


Fig. 2-8 Parameterization of feedrate reduction G62, example of a 90° corner

Parameterization

The override value is set in the following setting data:

42520: $SC_CORNER_SLOWDOWN_START
42522: $SC_CORNER_SLOWDOWN_END
42524: $SC_CORNER_SLOWDOWN_OVR
42526: $SC_CORNER_SLOWDOWN_CRIT

The setting data has default value 0.

- If $SC_CORNER_SLOWDOWN_CRIT == 0, the corner deceleration will only take effect at reversing points.
- If $SC_CORNER_SLOWDOWN_START and $SC_CORNER_SLOW-DOWN_END are equal to 0, the feedrate reduction will be approached with the permissible dynamic response.
- If $SC_CORNER_SLOWDOWN_OVR == 0, a brief stop will be inserted.
- $SC_CORNER_SLOWDOWN_CRIT refers to geometry axes with G62. It defines the maximum inside angle in the current machining plane up to which the corner deceleration will be applied. – G62 is not active on rapid traverse.

Activation

The function is activated with G62 or G621. The G code is activated either with the corresponding parts program command or with $MC_GCODE_RESET_VAL-UES[56].

Examples

```
$TC_DP1[1,1]=120
$TC_DP3[1,1]=0.           ;length offset vector
$TC_DP4[1,1]=0.
$TC_DP5[1,1]=0.

N1000 G0 X0 Y0 Z0 F5000 G64 SOFT
N1010 STOPRE
N1020 $SC_CORNER_SLOWDOWN_START = 5.
N1030 $SC_CORNER_SLOWDOWN_END = 8.
N1040 $SC_CORNER_SLOWDOWN_OVR = 20.
N1050 $SC_CORNER_SLOWDOWN_CRIT = 100.

N2010 G1 X00 Y30 G90 T1 D1 G64
N2020 G1 X40 Y0  G62 G41   ;Inside corner to N2030,
                           ;but TRC still being selected
N2030 G1 X80 Y30           ;Inside corner to N2040 127 degrees
N2040 G1 Y70               ;Inside corner to N2050 53 degrees
N2050 G1 X40 Y40           ;Outside corner to N2060
N2060 G1 X20 Y70           ;Inside corner to N2070 97 degrees
N2070 G1 X00 Y60           ;Inside corner to N2080 90 degrees
N2080 G1 X20 Y20           ;Outside corner to N2090,
                           ;irrelevant because TRC
                           ;deselection
N2090 G1 X00 Y00 G40 FENDNORM

M30
```

2.3 Subprogram and macro technology

2.3.1 Subprogram technology: M98

Subprogram calls

Subprogram calls are programmed with M98 in ISO Dialect.
For the program syntax, see Fig. 2-9.

```
M98 P xxxxyyyy
      │   └── Program number (max. 4 digits)
      └────── Number of repetitions (max. 4 digits)
```

Fig. 2-9 Description of parameters allowed

The program syntax M98 Pxxxxyyyy is used to call a subprogram with the number yyyy and repeat it xxxx times. If the xxxx is not programmed, the subprogram is executed only once. The subprogram name always consists of 4 digits or is extended to 4 digits by adding 0’s.
For example, if M98 P21 is programmed, the parts program memory is searched for program name 0021.spf and the subprogram is executed once. To execute the subprogram 3 times, program M98 P30021.

SW 6 upwards

Until now the number of program executions (number of repeats) has been programmed in ISO Dialect M/T in conjunction with the subprogram number at address “P”.

As an alternative, the number of subprogram executions can now also be programmed at address “L”. The number of the subprogram is still programmed as Pxxxx. If the number of executions is programmed at both addresses, the number of executions programmed at address “L” is valid. The number of subprogram executions lies between 1 and 9999.

Example:

```
N20 M98 P20123        ;Subprogram 123.spf will be executed twice
N40 M98 P55 L4        ;Subprogram 55.spf will be executed four times
N60 M98 P30077 L2     ;Subprogram 77.spf will be executed twice
                      ;The number of executions programmed
                      ;at address “P” =3 is ignored
```

Subprogram termination

M99 terminates the subprogram.
If M99 Pxxxx is programmed, execution resumes at block number Nxxxx on the return jump to the main program. The block number must always begin with “N”. The system initially searches forwards for the block number (from the subprogram call towards the end of the program). If a matching block number is not found, the parts program is then scanned backwards (towards the start of the program).

If M99 appears without a block number (Pxxxx) in a subprogram, the subprogram is terminated and the processor jumps back to the main program to the block following the subprogram call.

If M99 appears without a block number (Pxxxx) in a main program, the processor jumps back to the start of the main program and runs the program again.

These M commands are not output to the PLC.

Subprogram return jump with “RET”

Valid only for ISO Dialect T.

In the Siemens shell cycles for stock removal (as in ISO Dialect), it is necessary after roughing to resume program execution in the main program after the contour definition. To achieve this, the shell cycle must contain a subprogram return jump to the block after the end of the contour definition. The RET command has been extended with two optional parameters for skipping the blocks with the contour definition in the stock removal cycles after the subprogram call (with G71–G73).

The command RET (STRING: <block number/label>) is used to resume program execution in the calling program (main program) at the block with <blocknumber/label>.

If program execution is to be resumed at the next block after <block number/label>, the 2nd parameter in the RET command must be > 0; RET (<block number/label>, 1). If a value > 1 is programmed for the 2nd parameter, the subprogram also jumps back to the block after the block with <block number/label>.

In G70–G73 cycles, the contour to be machined is stored in the main program. The extended RET command is required in order to resume execution after the contour definition in the main program at the end of G70 (finish cut via contour with stock removal cycle). To jump to the next NC block after the contour definition at the end of the shell cycle for G70, the shell cycle must be terminated with the following return syntax:

RET (“N” << $C_Q, 1)

Search direction:

The search direction for <block number/label> is always forwards first (towards the end of the program) and then backwards (towards the start of the program).

Example

```
N10 X10. Y20.
N20 G71 P30 Q60 U1 W1 F1000 S1500

   N10 ...                    ;Shell cycle for stock removal cycle
   N20 DEF STRING[6]BACK
   N30 ...
   N90
   N100 RET ("N"<<$C_Q, 1)    ;Return jump to block after
                              ;contour def. -> N70

N30 X50. Z20.
N40 X60.
N50 Z55.
N60 X100. Z70.
N70 G70 P30 Q60
N80 G0 X150. Z200.
N90 M30
```

Note

M30 in Siemens mode: is interpreted as a return jump in a subprogram.

M30 in ISO Dialect mode: is also interpreted as the end of the parts program in a subprogram.

2.3.2 Siemens language commands in ISO Dialect mode

Certain Siemens language commands are also required in ISO Dialect mode for Shopmill. These commands are executed in ISO Dialect mode. They include subprogram calls with and without passed parameters (not calls with Lxx, because address L has a different meaning for ISO Dialect), program section repetition and control structures. All other Siemens language commands are denied with an alarm in ISO Dialect mode.

The following Siemens language commands can be programmed when ISO Dialect mode is active:

REPEAT:

REPEAT	<Block number> [<Block number>] [P..]
REPEAT	UNTIL
REPEATB	<Block number> [P..]

Only block numbers, not labels are allowed as start and end identifiers.

IF – ELSE – ENDIF
FOR – ENDFOR
WHILE – ENDWHILE
IF<Condition> **– GOTO F**<Condition>
CASE

Modal and non-modal subprogram calls

```
N100   CALL "SHAFT" or
N100   MCALL SHAFT or
N100   SHAFT
```

Modal and non-modal subprogram call with parameter passing

```
N100   MCALL SHAFT ("ABC"; 33.5) or
N100   SHAFT ("ABC"; 33.5) subprogram call specifying path
N100   CALL"/_N_SPF_DIR/SHAFT or
N100   MCALL/_N_SPF_DIR/SHAFT or
N100   PCALL/_N_SPF_DIR/SHAFT
```

2.3.3 Extending the subprogram call for contour preparation with CONTPRON

In ISO Dialect, the contour definition for stock removal cycles G70 – G73 is not stored separately in a subprogram (as in SINUMERIK), but appears in the parts program (main program). When the cycles are called, the contour definition section is defined by a start and end block number. The cycles receive this block number as a passed parameter. The indirect subprogram call has been extended for Siemens adaptation cycles.

Previously, subprograms were called indirectly with CALL <program name>.

The indirect subprogram call has been extended as follows for access to the contour definition in the main program:

CALL [<program name>] BLOCK <start label> TO <end label>

If no program name or an empty string is specified as the program name, i.e. CALL BLOCK <start label> TO <end label>, the search for the program section (start/end label) is made in the program which is currently selected. The search for the labels is also performed in the selected program with MDA, ASUB etc. (i.e. in the case of MDA, the search for the labels is performed not in the MDA buffer but in the program with the selected program name). Programming this syntax directly in a main program has the same effect as repeating a program section in a loop with REPEAT <start label> <end label>, i.e. the search for the start and end label is performed in the program containing the CALL BLOCK ... command.

If a program name is specified, e.g. CALL <progName> BLOCK <start label> TO <end label>, the system searches for the program section (surrounded by the start/end label) in subprogram “progName”.

Example

```
Nxx G71 Pxx Q1110 U.. W..   ;ISO Dialect G function calls
                            ; shell cycle CYCLE395.spf
             ; _N_CYCLE395_SPF
             N10 .......
             .......
             Nxxx CYCLE95(....., "N"<<$C_P, "N"<<$C_Q)
                            ;Stock removal cycle with additional
                            ; parameters for start and end label

             PROC CYCLE95(....., STRING[20] startlab, STRING[20]
                            endelab)
             N10 ........
             .........
             Nxxx CONTPRON(...)

             N.... CALL "" BLOCK startlab TO endelab
                            ;  Read contour definition or
             N.... CALL BLOCK startlab TO endelab
                            ;call the contour program
             EXECUTE(...)
             ........
             Nxx M17
Nxxx .....
Nxxx RET ("N"<<$C_Q, 1)     ;Return jump to the next block after
                            ; the contour definition
N1120 ....

Nxxx M30
```

Note

The actual CONTPRON and EXECUTE calls do not have to be modified.

Search for start block number

The start block number (start label) of the contour definition is always searched first toward the end of the program (forward) and then toward the start of the program (backward).

If the same block number is programmed more than once, the next block number (label) after the current block in the program in which the contour definition is contained, is recognized as the start of the contour definition (see example). The current block is usually the block in which the stock removal cycle (shell cycle) was called in the main program.

Example

In stock removal cycle CYCLE395, the contour definition which appears between blocks `N10 – N30` in the main program is to be used (with `CALL BLOCK N10 TO N30` in CYCLE395). `N40` is the current program line in the main program.

The contour definition block is printed in **bold** lettering in the example.

```
N10 X10. Y20. N20 X30.
N30 Y10.
N40 G71 P10 Q30...       ;Call shell cycle for stock removal cycle
...                      ;(In the stock removal cycle
...                      ;"CALL BLOCK N10 TO N30" is programmed)
...                      ;The contour definition is found in the
                         ;lines printed in bold

N50 G90 G54
N60 F1000 G94
N10 X50. Y10.
N20 X33. Y11.
N30 X10.
N50 ....
N..  .....
N800 G71 P10 Q30         ;Call shell cycle for stock removal cycle
...                      ;(In the stock removal cycle, "CALL BLOCK N10 TO
                         ;N30" is programmed)
...                      ;The contour definition is found in
...                      ;the lines printed in italics

N999 ....
N10 X15.
N20 Y25.
N30 X33.
N1010 ....
N1020 .....
```

2.3.4 Macro commands with G65, G66 and G67

ISO Dialect

In ISO Dialect mode, macros are called in the parts program with G65 Pxx, G66 Pxx. A macro is a set of parts program blocks that are terminated with M17.

When the subprogram is called, the mode is switched from ISO mode to Siemens mode.

The following commands are used for selection and deselection:

- G65 Macro call, non-modal
- G66 Macro call, modal
- G67 Deselect modal macro call

Siemens

G commands G65 Pxx and G66 Pxx are used to start macro xx. G65 calls subprogram Pxx once. G66 activates the Pxx subprogram call modally, and the subprogram is then executed in every block containing axis movements (same as MCALL xx). G67 deactivates the modal subprogram call again (equivalent to G80 for cycle calls).

In a parts program block with G65 or G66, the address Pxx is interpreted as the program number of the subprogram containing the macro functionality. Address Lxx can be used to define the number of passes of the macro. If a number of passes is not programmed in the calling block, the macro is executed once. All further addresses in this parts program block are interpreted (as in ISO Dialect "Macro B") as passed parameters, and their programmed values are saved in system variables $C_A–$C_Z. These system variables can be read in the subprograms and evaluated for the macro functionality. If further macros are called with parameters within a macro (subprogram), the passed parameters must be saved in internal variables in the subprogram before the new macro call.

As in the case of the machining cycles, the language mode is switched implicitly to Siemens mode to allow the definition of internal variables. Therefore, if a further macro call appears in the subprogram, ISO Dialect mode must be selected again first.

System variables for the addresses I, J, K

Because addresses I, J, and K can be programmed up to ten times in a block by macro call, an array index must be used to access the system variables for these addresses. The syntax for these three system variables is then $C_I[..], $C_J[..], $C_K[..]. The values are stored in the array in the order programmed. The number of addresses I, J, K programmed in the block is stored in variables $C_I_NUM, $C_J_NUM, $C_K_NUM.

The passed parameters I, J, K for macro calls are treated as one block, even if individual addresses are not programmed. If a parameter is programmed again or a following parameter has been programmed with reference to the sequence I, J, K, it belongs to the next block.

To recognize the programming sequence in ISO mode, system variables $C_I_ORDER, $C_J_ORDER, $C_K_ORDER are set. These are identical arrays to $C_I, $C_K and contain the associated number of parameters.

Example:

```
N5 I10 J10 K30 J22 K55 I44 K33
    set1        set2     set3
$C_I[0]=10
$C_I[1]=44
$C_I_ORDER[0]=1
$C_I_ORDER[1]=3

$C_J[0]=10
$C_J[1]=22
$C_J_ORDER[0]=1
$C_J_ORDER[1]=2

$C_K[0]=30
$C_K[1]=55
$C_K[2]=33
$C_K_ORDER[0]=1
$C_K_ORDER[1]=2
$C_K_ORDER[2]=3
```

Cycle parameter $C_x_PROG

In ISO dialect 0 mode, the programmed values can be evaluated differently depending on the programming method (integer or real value). The different evaluation is activated via machine data.

If the MD is set, the control will behave as in the following example:

```
X100.     ;X axis is traveled 100 mm (100. with point => real value
Y200      ;Y axis is traveled 0.2 mm (200 without point => integer value
```

If the addresses programmed in the block are passed as parameters for cycles, the programmed values are always real values in the $C_x variables. In the case of integer values, the cycles do not indicate the programming method (real/integer) and therefore no evaluation of the programmed value with the correct conversion factor.

To indicate whether REAL or INTEGER has been programmed, there is the system variable $C_TYP_PROG. $C_TYP_PROG has the same structure as $C_ALL_PROG and $C_INC_PROG. For each address (A–Z) there is one bit. If the value is programmed as an INTEGER, the bit is set to 0, for REAL it is set to 1. If the value is programmed in variable $<number>, bit 2 = 1 is set.

Example:

M98 A100. X100 –> $C_TYP_PROG == 1.
Only bit 0 is set because only A is programmed as a REAL.

M98 A100. C20. X100 –> $C_TYP_PROG == 5.
Only bits 1 and 3 are set (A and C).

Restrictions:

Up to ten I, J, K parameters can be programmed in each block. Variable $C_TYP_PROG only contains one bit each for I, J, K. For that reason bit 2 is always set to 0 for I, J, and K in $C_TYP_PROG. It is therefore not possible to ascertain whether I, J or K have been programmed as REAL or INTEGER.

Parameters P, L, O, N can only be programmed as integers. A real value generates an NC alarm. For that reason the bit in $C_TYP_PROG is always 0.

Modal macro calls

With modal macro calls, the programmed addresses are only copied into the system variables in the block containing the call (block with G66). The macro is then executed in every block with an axis movement until it is deselected by G67 or a new macro call is programmed with G66. Only the macro parameters are passed in the block containing the call (= block with G66) for modal macros. The macro is executed for the first time in the next block containing an axis movement. (Same procedure as MCALL xx in Siemens mode)

Example of a macro call:

```
_N_M10_MPF:

N10 M3 S1000 F1000
```

```
N20 X100. Y50. Z33.
N30 G65 P10 F55 X150. Y100. S2000
N40 X50.
N50 ....
N200 M30
```

Example of a subprogram as macro in Siemens mode:

```
_N_10_SPF:
N10 DEF REAL X_AXIS, Y_AXIS, SPEED, FEEDRATE
N15 X_AXIS = $C_X Y_AXIS = $C_Y SPEED = $C_S FEEDRATE = $C_F
N20 G01 F=FEEDRATE  G95 S=SPEED
...
M17
```

2.3.5 Mode changing in macro calls with G65/G66

Until now, automatic switchover to Siemens mode was performed for macro calls with G65/G66.
The user now has the choice whether switchover to Siemens mode takes place when the macro starts or not. Switchover to Siemens mode only takes place when the PROC<program name> instruction is used in the first line of the macro program. If this instruction is missing, ISO mode will remain active during execution of the macro program.

The user can therefore decide whether to create local variables (with DEF...) for the purpose of storing transfer variables. It is necessary to switch to Siemens mode to do this using the PROC instruction. The user can also specify that the macro program (e.g. an existing ISO Dialect M/T macro) is executed in ISO mode.

Example of a macro call:

```
_N_M10_MPF:
N10 M3 S1000 F1000
N20 X100. Y50. Z33.
N30 G65 P10 F55 X150. Y100. S2000
N40 X50.
N50....
N200 M30
```

Example of a subprogram as macro in Siemens mode:

```
_N_0010_SPF:
PROC 0010    ;Switchover to Siemens mode
N10 DEF REAL X_AXIS, Y_AXIS, SPEED, FEEDRATE
N15 X_AXIS=$C_X Y_AXIS=$C_Y SPEED=$C_S FEEDRATE=$C_F
N20 G01 F=FEEDRATE G95 S=SPEED
....
N80 M17
```

Example of a subprogram as macro in ISO mode:

```
_N_0010_SPF:
G290          ;Switchover to Siemens mode

              ;If transfer variables have to be read
N15 X_AXIS=$C_X Y_AXIS=$C_Y SPEED=$C_S
N20 G01 F=$C_F G95 S=$C_S
N10 G1 X=$C_X Y=$C_Y
G291          ;switch to ISO mode
N15 M3 G54 T1
N20
....
N80 M99
```

2.3.6 Macro call with M function

A macro can be called using M numbers in the same way as G65 (see 2.3.5).

10 M function substitutions are configured with machine data
10814: $MN_EXTERN_M_NO_MAC_CYCLE and
10815: $MN_EXTERN_M_NO_MAC_CYCLE_NAME.

Parameter transfer is executed in exactly the same way as with G65. Repetitions can be programmed with address L.

Restrictions

Only one M function substitution (and/or only one subprogram call) can be executed in each line of a parts program. Conflicts with other subprogram calls are output with alarm 12722. No further M function substitutions are made in the replaced subprogram.

Otherwise, the same restrictions apply as for G65

Conflicts with predefined and other defined M numbers are rejected with an alarm.

Configuration examples

Subprogram M101_MACRO call with M function M101

$MN_EXTERN_M_NO_MAC_CYCLE[0] = 101
$MN_EXTERN_M_NO_MAC_CYCLE_NAME[0] = "M101_MACRO"

Subprogram M6_MAKRO call with M function M6.

$MN_EXTERN_M_NO_MAC_CYCLE[1] = 6
$MN_EXTERN_M_NO_MAC_CYCLE_NAME[1] = "M6_MACRO"

Programming example for tool change with M function:

```
PROC MAIN
...
N10          M6 X10 V20
...
N90          M30

PROC M6_MACRO
...
N0010        R10 = R10 + 11.11
N0020        IF $C_X_PROG == 1 GOTOF N40
display($C_X_PROG)
N0030        SETAL(61000)  ;programmed variable incorrectly
                           ;transferred
N0040        IF $C_V == 20 GTOF N60
display($C_V)
N0050        SETAL(61001)
N0060        M17
```

2.3.7 Macro call with G function

A macro can be called using G numbers in the same way as G65 (see 2.3.5).

50 G function substitutions are configured with machine data
10816: $MN_EXTERN_G_NO_MAC_CYCLE and
10817: $MN_EXTERN_G_NO_MAC_CYCLE_NAME.

The parameters programmed in the block are saved in the $C_ variables. Address L is used to define the number of times a macro is repeated. The number of the programmed G_macro is stored in variable $C_G. All other G functions programmed in the block are treated like normal G functions. The sequence in which addresses and G functions are programmed in the block is irrelevant and has no effect on the functionality.

All ISO G codes, even G codes with a decimal point (= real value) can be replaced by a macro call.

G functions that are replaced by a macro do not exist in the control and can be redefined with
10822: $MN_NC_USER_EXTERN_GCODES_TAB[ ].

Restrictions

Only one G/M function substitution (and/or only one subprogram call) can be executed in each line of a parts program. Conflicts with other subprogram calls, e.g. when a modal subprogram call is active, are signaled with alarm 12722.

If a G macro is active no more G/M macros or M subprograms are called. M macros/subprograms are then executed as M functions, and G macros as G functions if the relevant G function exists. Otherwise alarm 12470 is output.

Otherwise, the same restrictions apply as for G65

Configuration examples

Subprogram G21_MAKRO call with G function G21

$MN_EXTERN_G_NO_MAC_CYCLE[0] = 21
$MN_EXTERN_G_NO_MAC_CYCLE_NAME[0] = “G21_MACRO”

$MN_EXTERN_G_NO_MAC_CYCLE[1] = 123
$MN_EXTERN_G_NO_MAC_CYCLE_NAME[1] = “G123_MACRO”
$MN_EXTERN_G_NO_MAC_CYCLE[2] = 421
$MN_EXTERN_G_NO_MAC_CYCLE_NAME[2] = “G123_MACRO”

Programming example:

```
PROC MAIN
...
N0090 G291                          ;ISO mode
N0100 G1 G21 X10 V20 F1000 G90      ;G21_MACRO.spf, G1, and
                                    ;G90 calls activated
                                    ;before G21_MACRO.spf
                                    ;is called
...
N0500 G90 X20 Y30 G123 G1 G54       ;G123_MACRO.spf, G1,
                                    ;G54, and G90 calls
                                    ;activated before
                                    :G123_MACRO.spf
                                    ;is called
...
N0800 G90 X20 Y30 G421 G1 G54       ;G421_MACRO.spf, G1,
                                    ;G54, and G90 calls
                                    ;activated before
                                    ;G123_MACRO.spf is called
...
N0900 M30
PROC G21_MACRO
...
N0010  R10 = R10 + 11.11
N0020  IF $C_X_PROG == 0
N0030  SETAL(61000)                ;programmed variable incorrectly
                                   ;transferred
N0040  ENDIF
N0050  IF $C_V_PROG == 0
N0060  SETAL(61001)
N0070  ENDIF
N0080  IF $C_F_PROG == 0
N0090  SETAL(61002
N0100  ENDIF
N0110 G90 X=$C_X V=$C_V
N0120 G291
N0130 G21 M6 X100                  ;G21->activates metric system of
                                   ;units (no macro call)
N0140 G290
...
N0150 M17
PROC G123_MACRO
...
N0010  R10 = R10 + 11.11
N0020  IF $C_G == 421 GOTOF label_G421
                           ;macro functionality for G123
N0040 G91 X=$C_X Y=$C_Y F500
...
```

```
...
N1990  GOTOF label_end
N2000 label_G421:      ;macro functionality for G421N2010 G90 X=$C_X
Y=$C_Y F1000
N2020
...
...
N3000 G291
N2010 G123             ;alarm 12470 because G123 is not a G function
                       ;and a macro cannot be called when a macro is
                       ;active. Exception: the macro was called as
                       ;a subprogram with CALL G123_MACRO.
N4000 label_end: G290
N4010 M17
```

2.3.8 High-speed cycle cutting G05 P..

G05 P.. high-speed cycle cutting takes the form of a subprogram call.

Programming G05 P.. L..

Pxxxxx Subprogram number, max. 10 characters
When called it is not necessary to fill with zeros as is the case with M98.

Lxxxx Number of passes. If L is not programmed, L1 is assumed.

Example:
G05 P10123 L3 10123.mpf is passed through three times.

This call can be used to fetch any subprogram. This subprogram can be a precompiled program, but does not have to be. However, only a Siemens parts program can be precompiled.

There is not equivalent of ISO Dialect function G05 in Siemens mode. CYCLE305 enables users to program their own functionality in the context of the Siemens functionality.

CYCLE305.spf is called when programming G05 in the following cases:

- G05 without P in the block is skipped without an alarm.
- G05.1 with and without P is skipped without an alarm.
- G05 P0 or P01 are reserved for high-speed remote buffer B. This function is not supported.

In the cases mentioned, all addresses programmed in the block are defined in cycle parameter $C_xx. When CYCLE305 is called there is no automatic change of mode from ISO to Siemens. If it is intended to process CYCLE305.spf in Siemens mode, the first program line must contain a PROC instruction as in the case of macro calls with G65/G66.

All functions programmed in the block are executed, as previously mentioned in the case of programming G05, that is to say, programmed axes are traversed, auxiliary functions are produced, etc. The programmed addresses are defined in the cycle parameters only for the purpose of supplementary information.

If G05 and a subprogram call (M98 P..) are programmed in the same block, alarm 12722 is produced.

2.3.9 Switchover modes for DryRun and skip levels

Switching over the skip levels (DB21 DBB2) always meant intervening in the program run which until now resulted in a momentary drop in velocity along the path. The same applies when the dry run mode DryRunOff (DryRun = dry-run feedrate DB21.DBB0.BIT6) it switched to DryRunOn and vice versa.

This drop in voltage can now be avoided with a new switchover mode which has a restricted functionality.

With machine data assignment $MN_SLASH_MASK==2 a drop in voltage is no longer necessary when switching skip levels (i.e. a new value in PLC–>NCK-Chan interface DB21.DBB2).

Note

The NCK processes blocks in two stages, preliminary or preprocessing and main processing. The result of preprocessing is put into the preprocessing memory. The main processing stage takes the oldest block from the pre–processing memory and traverses its geometry.

Attention

With machine data assignment $MN_SLASH_MASK==2, preprocessing is switched over when the skip levels are switched! All blocks in the preprocessing memory are executed with the old skip level. As a rule, the user has no influence over the level of the preprocessing memory. The user observes the following: **The new skip level can take effect at any time after switchover!**

Note

The parts program command STOPRE empties the preprocessing memory. If the skip level is switched over before STOPRE, it is certain that all blocks after STOPRE will be switched over. The same applies to an implicit STOPRE.

Switching over DryRun mode results in the same restrictions.

With machine data assignment 10704: $MN_DRYRUN_MASK==2 no drop in velocity is necessary when DryRun mode is changed. However, here too, only preprocessing is switched over, which results in the above restrictions. Analogously the following applies: **Caution DryRun mode can become active any time after switchover!**

2.3.10 Eight-digit program numbers

Eight-digit program number selection is activated with machine data $MC_EXTERN_FUNCTION_MASK, bit6=1. This function has an effect on M98 (see Subsection 2.3.1), G65/66 (see Subsection 2.3.5), and M96 see Subsection 2.2.17).

y: Number of program runs
x: Program number

Subprogram call M98

$MC_EXTERN_FUNCTION_MASK, bit6 = 0
M98 Pyyyyxxxx or
M98 Pxxxx Lyyyy
Program number max. four digits
Extension of program number always to four digits with 0

E.g.:	M98 P20012	calls 0012.mpf 2 passes
	M98 P123 L2	calls 0123.mpf 2 passes

$MC_EXTERN_FUNCTION_MASK, bit6 = 1
M98 Pxxxxxxxx Lyyyy
No extension with 0 even if the program number is less than four digits long.
It is not possible to program the number of passes and program number in P (Pyyyyxxxxx),
the number of passes must always be programmed with L!

E.g.:	M98 P123	calls 123.mpf 1 pass
	M98 P20012	calls 20012.mpf 1 pass,
	Caution: This is no longer compatible with the ISO dialect original	
	M98 P12345 L2	calls 12345.mpf 2 passes

Modal and block-by-block macro G65/G66

$MC_EXTERN_FUNCTION_MASK, bit6 = 0
G65 Pxxxx Lyyyy
Extension of program number always to four digits with 0. Program number with more than four digits triggers an alarm.

$MC_EXTERN_FUNCTION_MASK, bit6 = 1
G65 Pxxxxxxxx Lyyyy
No extension with 0 even if the program number is less than four digits long. Program number with more than eight digits triggers an alarm.

Interrupt M96

$MC_EXTERN_FUNCTION_MASK, bit6 = 0
M96 Pxxxx
Extension of program number always to four digits with 0

$MC_EXTERN_FUNCTION_MASK, bit6 = 1
M96 Pxxxx
No extension with 0 even if the program number is less than four digits long. Program number with more than eight digits triggers an alarm.

2.3.11 System variable for level stack in ISO mode

In standard mode, the current program level is displayed in system variable $P_STACK. Every subroutine call and return affects this variable. However, there are subroutine calls in ISO mode for which the current user variable level does **not** change. The implementation of level-specific variables using GUDs requires knowledge of the current program level in ISO mode. System variable $P_IPO_STACK supplies the current program level in ISO dialect mode.

Table 2-6 shows all possible subroutine and macro calls in ISO mode and how they affect the current program level.

Calls in ISO mode are mapped to calls in standard mode so that variable $P_STACK contains the same information as before even in ISO mode.

The maximum possible number of subroutine calls remains unchanged.

System variable $P_IPO_STACK is always incremented when a subroutine programmed in ISO mode as a macro call with G65, G66, G code or M macro starts. On return from this type of ISO macro, $P_IPO_STACK is decremented again. If no ISO macros are active, $P_IPO_STACK = 0. $P_IPO_STACK therefore supplies the number of currently active ISO macros.

When a subroutine defined with M96 Pxx is called, variable $P_IPO_STACK is also incremented on the basis of MD $MC_EXTERN_FUNCTION_MASK bit 11.
If $MC_EXTERN_FUNCTION_MASK
bit 12 = 0, $P_IPO_STACK is not modified by the interrupt program.
If bit 12 = 1, $P_IPO_STACK is incremented by the interrupt program.

Cycle calls with e.g., G81, G77 etc. or functions implemented internally with cycles, e.g., G05, G72.1, etc. and subroutine calls with M98 Pxx have no effect on $P_IPO_STACK.

Example: Subroutine calls in ISO and standard mode.
M98 indicates subroutine calls without level incrementation.
G65 P indicates macro calls with level incrementation.

Table 2-6 Subroutine and macro calls

$P_STACK	$P_IPO_STACK	Level 1	Level 2	Level 3
1	1	O111.mpf		
1	1	N5 M98 P2222		
2	1		O2222.mpf	
2	1		G65 P3333	
3	2			O3333.mpf
3	2			M99
2	1		M99	

Table 2-6 Subroutine and macro calls

$P_STACK	$P_IPO_STACK	Level 1	Level 2	Level 3
M98 does not increment the levels. O1111.mpf and O2222.mpf work with the same $P_ISO_STACK content, G65 does increment the levels, so that the content seen by O3333.mpf is different. $P_STACK continues to display the levels in standard mode.				
$P_STACK	**$P_IPO_STACK**	**Level 1**	**Level 2**	**Level 3**
1	1	O1111.mpf		
1	1	N5 G65 P2222		
2	2		O2222.mpf	
2	2		M98 P3333	
3	2			O3333.mpf
3	2			M99
2	1		M99	
Switching from ISO mode to standard mode				
$P_STACK	**$P_IPO_STACK**	**Level 1**	**Level 2**	**Level 3**
1	1	O1111.mpf		
1	1	G291		
1	1	N5 M98 P2222		
2	1		O2222.mpf	
2	1		G290	
2	1		3333( )	
3	2			3333.mpf
3	2			M30
2	1		G291	
2	1		M99	
1	1	N10 M30		
Switching from standard mode to ISO mode				
1	1	1111.mpf		
1	1	N5 G290		
1	1	N10 2222( )		
2	2		2222.mpf	
2	2		G291	
2	2		M98 P3333	
3	2			O3333.mpf
3	2			M99
2	2		G290	
2	2		M17	
1	1	N15 M30		

List of possible subroutine and macro calls in ISO mode

M98 Pxxxx	Subroutine call	Level does not change
M98 Pxxxx Lyyyy	Up call with iteration	Level does not change
G65 P	Non-modal macro	Level incremented
G66 P	Modal macro	Level incremented
G05	UP call CYCLE305	Level does not change
M macro subst		
10814: EXTERN_M_NO_MAC_CYCLE		Level incremented
M Up subst.		
0715: M_NO_FCT_CYCLE		Level does not change
T subst		
10717: T_NO_FCT_CYCLE_NAME0		Level does not change
G subst		
10816__EXTERN_G_NO_MAC_CYCLE		Level incremented
M96	Interrupt ASUP	Level changes depending on $MC_EXTERN_FUNC-TION_MASK, bit12
Shell cycles:		Level is not incremented
G code cycles:		
G22 G23 G27 G28 G30 G30.1 G72.1 G50		Level is not incremented

G code cycles, Shell cycles:
$P_ISO_STACK has no relevance for the user as write access is not supported for these cycles.

Depending on machine data $MC_EXTERN_FUNCTION_MASK, bit 12, variable $P_ISO_STACK is incremented when an interrupt program (ASUP) is called.

Bit12 = 0 Variable $P_ISO_STACK does not change when an interrupt program defined with M96 Pxx is called

Bit12 = 1 Variable $P_ISO_STACK is incremented when an interrupt program defined with M96 Pxx is called

2.4 Tool change and tool offsets

2.4.1 Tool offsets: T, D, M (ISO Dialect M)

Tool data T/D number H number

As Siemens and ISO dialect programs are intended to run alternately in the control they must be implemented with the Siemens tool data memory.
In each offset memory that exists for a tool, the length, geometry and wear in each case are specified.
In Siemens mode, the offset memory is addressed with T (tool number) and D (cutting edge number), or **T/D number** for short.
In ISO Dialect M programs, the offset memory is addressed with D (radius) or H (length). This is referred to below as the **H number.**
In order to establish a unique assignment between this H number and a T/D number, an element $TC_DPH[t,d] has been added to the offset data set. The H number of the ISO Dialect is entered in this element.

Table 2-7 Example: Tool offset data set

T	D/cutting edge	H number $TC_DPH	Radius	Length
1	1	10		
1	2	11		
1	3	12	100.00	250.00
2	1	13		
2	2	14		
2	3	15		

Example:

Siemens program	ISO Dialect program
N5 T1	N5 T1
N10 G41 D3	N10 G41 H12 or G41 D12

When **the H value is programmed** in the ISO dialect M program, the system searches for and activates the matching $TC_DPH in the active T after the correction block.

If the correction block does not contain an H number, the compensation cannot be activated in ISO Dialect mode.

If an H is programmed but a correction block with the corresponding H number is not found or the associated tool T is not selected, an alarm is output.

2.4.2 Possible H numbers

H = 0

All data of the tool edges with H number 0 are not linked. Every tool edge has its own parameters.

H = unique

An H number in a TO unit must exist only once otherwise clear addressing of the compensation block is not possible. In case an H number has been allocated for a second time, alarm “17183 channel %1 block %2 H number already exists in T= %3 with D=%4” is given when writing from the program,. The alarm is compensation block compatible with NC Start clear.

Example:

```
N5   $TC_DPH[1,1] = 5
N10  $TC_DPH[2,1] = 5
```

An attempt to allocate an H number twice via OPI (HMI, PLC) will lead to a negative acknowledgement when writing.

Changing the offset memory

Existing tool offsets can be overwritten with G10. New tool offsets are not created by G10.

Tool length compensation, geometry:	G10 L10 Pxx Ryy
Tool length compensation, wear:	G10 L11 Pxx Ryy
Tool radius compensation, geometry:	G10 L12 Pxx Ryy
Tool radius compensation, wear:	G10 L13 Pxx Ryy

P specifies the H number of the compensation memory and R specifies the value. L1 can be programmed instead of L11.

Active plane

Setting data $SC_TOOL_LENGTH_CONST must be assigned value 17 if the assignment of tool length offsets to geometry axes is to be independent of plane selection. Length 1 is then always assigned to the Z axis.

Selecting the tool length

The tool length and the tool radius are always programmed with D or H.

Example:

T	D/cutting edge	H number $TC_DPH	Radius	Length
2	3	4	10	15

ISO Dialect M:
T2
G43 H4 or D4 ;Length selection
G42 D4 or H4 ;Radius selection

The offset value must be entered twice for ISO Dialect M programs which are programmed with different D and H numbers.

Example:

T	D/cutting edge	H number $TC_DPH	Radius	Length
2	3	4	10	15
2	4	5	10	15

ISO Dialect M:
T2
G43 H4 ;Length offset from T2 D3
G42 D5 ;Radius and length offset from T2 D4

Flat D number

If flat D numbers are active, the T is programmed independently of the H number. The H number is no longer checked for compatibility with the selected tool.

An H number must be assigned to every offset memory, even with flat D numbers.

Tool management

If tool management is active, replacement tools have the same H number. Duplo numbers are used in order to differentiate.
Offset D1 is activated for the currently selected tool on H99 with active **tool management**.

In ISO Dialect M, only numerical expressions are permitted as tool identifiers. Strings are no longer permitted as identifiers.
Example: T = “2”, selection with T2.

Tool length compensation in multiple axes

Tool length offsets can be activated on multiple axes. However, the resulting tool length compensation cannot be displayed.

The Siemens T and D numbers appear in the display for active T and D numbers. New OPI variables which can be displayed are available for the active ISO Dialect H and D number.

Machine data 22220: $MC_AUXFU_T_SYNC_TYPE is used to define whether the output to PLC takes place during or after the movement.
Machine data 20110: $MC_RESET_MODE_MASK, bit 6 can be used to activate tool length compensation beyond a reset.

Example: Tool selection in ISO dialect M:

```
; (Fanuc 0 M tool offset with T, cutting edge number
; (the offsets are written)
; (with G10)
G290

; Tool offset memory T2 cutting edge 1:
N5000 $TC_DP1[2,1]=10      ;type
N5000 $ TC_DP1[2,1]=7      ;ISO H number

; Tool offset memory T3 cutting edge 2:
N5000 $TC_DP1[3,2]=10      ;type
N5000 $TC_DP1[3,2]=3       ;ISO H number

; Tool offset memory T4 cutting edge 3:
N5000 $TC_DP1[4,3]=10      ;type
N5000 $TC_DP1[4,3]=8       ;ISO H number

G291

;Write tools offsets
;------------------------
;T2 cutting edge 1
G10 L12 P7 R5
; T3 cutting edge 2
G10 L10 P3 R15
G10 L12 P3 R10

N8        G01 G40 F5000 X0 Y0 Z0
N10       X50.
N15       50
N17       Z10.
N20       X0
N25       Y0
N30       X-10 Y-10
```

```
N30     T2          ;Tool 2
N33     G43 H7 Z0   ;H number 7
N35     G41 X0 Y0 Z0 D7
N40     X50.
N45     Y50.
N48     Z10.
N50     X0
N55     Y0
N60     G40 X-10 Y-10

N65     T3
N68     G43 H3 Z0
N70     G42 X0 Y0 Z0 D3
N75     X50.
N77     Y50.
N78     Z10.
N80     X0
N85     Y0
N90     G40 X-10 Y-10

N95     T4
N98     G43 H8 Z0
N100    G41 X0 Y0 Z0 D8
N105    X50.
N110    Y50.
N112    Z10.
N115    X0
N120    Y0
N125    G40 X-10 Y-10

M30
```

Machine data 20382: $MC_TOOL_CORR_MOVE_MODE defines whether the compensation is applied in the block containing the selection or the next time the axis is programmed.

2.4.3 Tool offset: T (ISO dialect T)

Tool data are stored in the Siemens tool data memory.
Every tool comprises four entries, one each for the X axis, Z axis, radius and cutting edge position. The range of values for tool length and radius offset is $\pm$999.999 mm. The range of values for the cutting edge position is 0 – 9, where 0 and 9 are identical.
The meaning is equivalent to the tool point direction on Siemens turning tools.

T is used for selection. T contains the tool number and offset number.

The offset is addressed either with the Siemens cut number D or with the H number from $TC_DPH. Addressing with D is only possible for “flat D numbers”. If tool management is used, H is always used for addressing.

Txxxxyyyy: xxxx=Tool number, yyyy=offset number

Machine data 10890: $MN_EXTERN_TOOLPROG_MODE, bit 0 defines how the T value is interpreted.

The number of digits in the tool number is defined in machine data 10888: $MN_EXTERN_DIGITS_TOOL_NO. The digits are counted from left to right. Subsequent digits indicate the offset number.

Bit 0=1 in MD 10890 sets the offset number to the same value as the tool number.

Example:

```
$MN_EXTERN_TOOLPROG_MODE=0
$MN_EXTERN_DIGITS_TOOL_NO=2
T1234          ;Auxiliary function T1234 on PLC
               ;Tool number 12
               ;Offset selection D34/H34

T123           ;Auxiliary function T123 on PLC
               ;Tool number 12
               ;Offset selection D3/H3

$MN_EXTERN_TOOLPROG_MODE, Bit0=1
T12            ;Auxiliary function T12 on PLC
               ;Tool number 12
               ;Offset selection 12
```

Machine data 20382: $MC_TOOL_CORR_MOVE_MODE is used to select when the offset is applied: immediately when the set is selected or not until the axis is programmed.

MD 20110: $MC_RESET_MODE_MASK, bit 6 is used to define whether the offset is maintained in the event of a rest or deselected.

MD 20360: $MC_TOOL_PARAMETER_DEF_MASK, Bit 0 is used to activate the calculation of the wear value for the transverse axis as a diameter value. The geometry offset is always applied as a radius.

Example: Tool selection, ISO dialect T:

```
G290
N5000  ;Definition of tool offset memory D1:
N5000  $TC_DP1[1,1]=10        ;Type
N5000  $TC_DP2[1,1]=9         ;Cutting edge position
N5000  $TC_DP6[1,1]=5         ;Radius

N5000  ;Definition of tool offset memory D2:
N5000  $TC_DP1[2,1]=10        ;Type
N5000  $TC_DP2[2,1]=1         ;Cutting edge position
```

```
N5000   $TC_DP6[2,1]=5          ;Radius

G291
;Write tool offset data
N3      G10 P1 X10 Z20 Y30
N5      G10 P2 X20 Y20 Z100

N10     G00 G18 X0 Y0 Z0
N10 T0101                       ;Tool 1, cut 1
N15     G00 X10 Y10 Z10
N20     T0201                   ;Tool 1, cut 1
N25     G00 X10 Y10 Z10
...
M30
```

Changing the offset memory

Although existing tool offsets can be overwritten with G10, new tool offsets are not created with G10.

G10 P<100 / 10000 X Y R Q Geometry
G10 P>100 / 10000 X Y R Q Wear

P100/10000	;MD 20734: EXTERN_FUNCTION_MASK, bit 1 is used to select ;whether a differentiation is made on the basis of geometry or ;wear if P<100 or 10000.
X Y Z	;Absolute or incremental offset values, depending on G90/91
U V W	;Incremental offset values
R	;Radius
Q	;Cutting position

Tool offset selection with $TC_DPH

Previously, the “flat D number” function was always active for ISO dialect T. D numbers are unique and command Txxyy or G10 Pyy is used to address the Siemens cut number with yy. In order to use tool management, structured D numbers must be addressed in ISO dialect T. Exactly as in ISO dialect M, every cut is assigned a parameter $TC_DPH[ ], which enables a cut to be addressed uniquely within a TO unit.

The function is switched on by setting MD 10890: $MN_EXTERN_TOOLPROG_ MODE bit 2=1.
When the function is active, the tool offset must always be addressed with the H number in ISO dialect T. Programs, which address the cut number, no longer run. Parameter $TC_DPH[ ] is only created if $MN_EXTERN_TOOLPROG_MODE bit 2=1. H numbers must be assigned uniquely within a TO in order to prevent alarms.

There are 3 options:

1. **Flat D number + $MN_EXTERN_TOOLPROG_MODE bit 2=0**
   The offset is always addressed with cut D.

```
G290
N605 $TC_DP1[1]= 500
N615 $TC_DP1[2]= 500
N625 $TC_DP1[3]= 500
N635 $TC_DP1[4]= 500

G291
N650 G10 P2   X10            ; Write geometry cut 2
N655 G10 P102 X1             ; Write wear cut 2

N670 T0102      ;Select cut 2
N675 T0105      ;Alarm because cut 5 is not available.
```

2. **Flat D number + $MN_EXTERN_TOOLPROG_MODE bit 2=1**
   The offset is always addressed with the H number;

```
G290
N705 $TC_DP1[1]= 500
N708 STC_DPH[1]=11

N710 $TC_DP1[2]= 500
N715 STC_DPH[2]=22
N720 $TC_DP1[3]= 500
N725 STC_DPH[3]=33
N730 $TC_DP1[4]= 500
N735 STC_DPH[4]=44

G291
N740 G10 P22   X10.          ; Write geometry cut 2
N745 G10 P122 X1.            ; Write wear cut 2
N747 G10 P55   X10.          ; Alarm 12550, because cut is
                             ; not available with H55

N750 T0122      ; Cut 2 is selected
N752 T0155      ; Alarm 12550, because cut is
                ; not available with H55
```

3. **Structured D number + $MN_EXTERN_TOOLPROG_MODE bit 2=1**
   The offset is always addressed with the H number.

```
G290
N805 $TC_DP1[1,1]= 500
N808 STC_DPH[1,1]=11

N810 $TC_DP1[1,2]= 500
N815 STC_DPH[1.2]=22

N820 $TC_DP1[2,1]= 500
N825 STC_DPH[2,1]=33
```

```
N830 $TC_DP1[2,2]= 500
N835 STC_DPH[2,2]=44

G291
N840 G10 P22   X10      ; Write geometry T1 cut 2.
N845 G10 P122 X1        ; Write wear T1 cut 2.
N847 G10 P55 X1         ; Alarm 12550, because cut is
                        ; not available with H55


N850 T0122    ; Select T1 cut 2.
N855 T0244    ; Select T2 cut 2. Alarm, because no cut with H22 is available
              ; in T2.
```

2.4.4 Tool-changing cycle

ISO Dialect mode

MD 10717 T_NO_FCT_CYCLE_NAME is used to assign a subprogram to the T command. Every block that contains a T command is executed and the subprogram is subsequently called up. The T value is not output; the T command must be programmed again in the cycle.

System variable $C_T_PROG or $C_D_PROG can be used in the subprogram to check whether the T or D command was programmed. The values can be read out with system variable $C_T or $C_D. If another T command is programmed in the subprogram, no substitution takes place, but the T word is output to the PLC.

The machine data 10715 M_NO_FCT_CYCLE and 10716: M_NO_FCT_CYCLE_NAME can be used to assign a subprogram to an M function (e.g. M06).

The mapping of M and T programming onto cycle calls has the same effect in ISO Dialect mode as in Siemens mode.

If T and M6 are programmed in the same block, the programmed T number can be scanned with $P_TOOL in the cycle called by M6. The M/T call is also mapped onto the cycle call in the block search. The start of the change cycle after the end of the search run must be initiated by the PLC.

Sequence:

```
N20 T1234
N30 M6        ;Change tool
N40 H3 G43    ;Activate tool length compensation in T1234
N50 T333      ;Tool preselection
N60 G1 X10    ;Offset T1234 is active
N70 M6        ;Load tool 333, D0 H0 active
N80 H4        ;Activate new tool length compensation
N90 .....
```

■

Cycles and Contour Definition 3

3.1 Calling cycles in the external CNC system using G commands

General description

The functionality of the ISO Dialect cycles is implemented in the standard Siemens cycles:
A shell cycle is called from the ISO Dialect program. All addresses programmed in the block are passed to this shell cycle in the form of system variables. The shell cycle matches the data to the standard Siemens cycle and calls it by name.
Machine manufacturers can replace these shell cycles with their own cycles.

Cycle parameters

Various cycle parameters in channel-specific GUD (Global User Data) must be initialized for the machining cycles. The names and meanings of the GUD are listed in Section 3.2.

Procedure for cycle call via G command

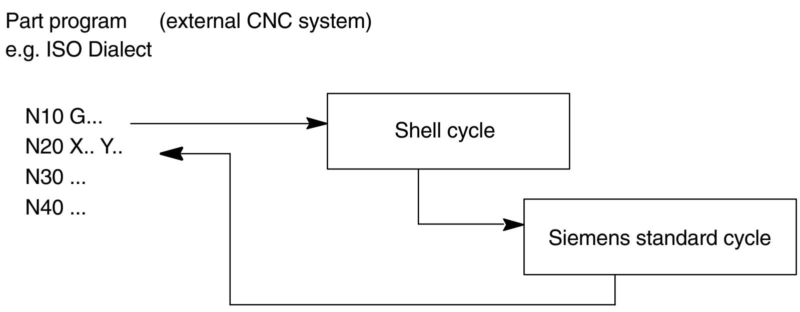


Fig. 3-1 General cycle call in ISO Dialect mode

Shell cycle

The modifications required due to the ISO Dialect programming syntax are made in the shell cycle. This means that the existing SINUMERIK cycles do not have to be changed. The name of the shell cycle is permanently defined.

Procedure:

4. The cycle (e.g. G81) is programmed in ISO Dialect mode
5. Siemens mode is activated automatically and the associated shell cycle is called (see Fig. 3-2)
6. The shell cycle calls the associated Siemens standard cycle

It is not necessary to program G290. The external CNC system is automatically activated on the return jump.

Important

The cycles must only be called with G commands.
This ensures that the appropriate cycle parameters are passed to the shell cycle.

The shell cycle **must not** be activated directly with CALL CYCLE3xx!

Modal cycles

If a modal cycle is active, the shell cycle is called in every NC block. If no axis positions (X, Y or Z) are programmed in the NC block, the Siemens standard cycle is not called.

Addresses programmed in the block (F etc.) are activated via the shell cycle. If no feedrate was programmed, for example, the current feedrate is used as the path feed.

Cycle parameters can be programmed in the following blocks while a modal cycle is active. These parameters are copied into the system variables so that the shell cycle uses the modified parameters.
Modal cycles are, in contrast to modal macros, already executed in the calling block (e.g. block with G81 etc.).

Deselecting the cycle:
Deselection is performed with G80 or with a function of the 1st G group.

Example:

```
N10 G81 X10. Y20. Z–15. R5 F1000
                            Drilling position X10mm, Y20mm
                            Drilling depth Z-15mm
                            Reference plane 5mm
                            Drilling feed F.. (mm/min or mm/rev)
N20 X50. Y30. R10           Drilling position X50mm, Y30mm,
                            New reference plane 10mm
N30 G80                     Delete cycle G81
```

Write cycle variable depending on addresses programmed in set

Previously, if modal cycles were active, all programmed addresses in the set were always written to the cycle variables. During the cycle, the variables are evaluated and decisions are made about how the variables must be used on the basis of the cycle logic.

In some cases, this means that the cycle parameters will be written even if they may not be interpreted as cycle parameters on the basis of the programming syntax.

Therefore, for the following functions, none or only some of the programmed addresses are written to the cycle parameters:

M98 P3 L2 X10 Y20	Addresses Pxx and Lxx are not written to the cycle parameters.
G05 P5 L2 X10 Y20	Addresses Pxx and Lxx are not written to the cycle parameters.
G05 P1 L2 X10 Y20	If a modal cycle is active, alarm 12722 will be output because the call is for the modal cycle for which the programmed values are actually intended.
G54 P10 X10 Y20 M44	Address Pxx is not written to the cycle parameters.
G31 P98 X30 F100	Addresses Pxx, Fxx and the axis values are not written to the cycle parameters.
G31 P1 X30 Y20 F100	None of the programmed addresses are written to the cycle parameters.
G51 P1000 I2 J3 K2 X30 Y40	None of the programmed addresses are written to the cycle parameters.
G50 P10000 X10 Y30	All parameters are written to the cycle parameters.

3.2 Global user data (GUD)

Table 3-1 GUD7 for programmed cycle values (ISO Dialect program data)

GUD	Description/use	CYCLE
Real values		
_ZFPR[0]	Initial plane (current position on 1st call with G..), retraction position active on G98	381M, 383M, 384M, 387M
_ZFPR[1]	Reference plane, retraction position active on G99 (retraction only possible to initial position with G87).	381M, 383M, 384M, 387M
_ZFPR[2]	Final drilling depth, absolute	381M, 383M, 384M, 387M
_ZFPR[3]	Retraction position, depending on G98/G99 (initial plane/R plane)	381M, 383M, 384M, 387M
_ZFPR[4]	Drilling feed	381M, 383M, 384M, 387M
_ZFPR[5]	Dwell time (s) at final depth (G82/G89/G76/G87)	381M, 384M, 387M
_ZFPR[6]	1st drilling depth (single drilling depth) incr. (G73/G83)	383M
_ZFPR[7]	1st drilling depth, absolute (G73/G83)	383M
_ZFPR[8]	Lift-off/infeed distance (G76)	387M
_ZFPR[9]	Speed for tapping (G74/G84)	384M
_ZFPR[20]	Initial plane (current position on 1st call)	383T, 384T, 385T
_ZFPR[21]	R plane	383T, 384T, 385T
_ZFPR[22]	Final drilling depth, absolute	383T, 384T, 385T
_ZFPR[23]	Retraction position (1=G98, 2=G99)	383T, 384T, 385T
_ZFPR[24]	Thread pitch/drilling feed	376T, 383T, 384T, 385T
_ZFPR[25]	Dwell time at final depth	383T, 384T, 385T
_ZFPR[26]	Speed for tapping	384T
_ZFPR[27]	End point X	371T, 372T, 373T, 376T
_ZFPR[28]	End point Z	371T, 372T, 373T, 376T
_ZFPR[29]	Start point offset X (taper thread)	371T, 372T, 376T
_ZFPR[30]	Thread start point X	376T
_ZFPR[31]	Thread start point Z	376T
_ZFPR[32]	First drilling depth	383T
Integer values		
_ZFPI[0]	Current G code of ISO Dialect drilling cycle	381M, 383M, 384M

Table 3-1 GUD7 for programmed cycle values (ISO Dialect program data)

GUD	Description/use	CYCLE
_ZFPI[1]	M function for spindle start (M3, M4) after spindle stop	381M, 384M
_ZFPI[20]	Current G code of threading cycle/drilling cycle	383T, 384T, 385T
_ZFPI[21]	Spindle direction (3=M3, 4=M4)	383T, 384T, 385T
_ZFPI[22]	Stock removal mode Roughing	370T, 371T, 372T, 373T
_ZFPI[23]	Machining mode Deep hole/Drilling	383T

Table 3-2 GUD7 for cycle setting data (ISO Dialect setting data)

GUD	Description/use	
Real values		
_ZSFR[0]	Safety clearance to reference plane	381M, 383M
_ZSFR[1]	Retraction amount for chipbreaking (G73)	383M
_ZSFR[2]	Angle offset for oriented spindle stop, tool must be oriented in +X direction (G76) Retraction direction: –X G17 plane XY –Z G18 plane ZX –Y G19 plane YZ	387M
_ZSFR[20]	Safety clearance to reference plane	383T, 384T
_ZSFR[21]	Safety clearance to chip break	383T, 385T
Integer values		
_ZSFI[0]	0=Drilling axis is perpendicular to plane (default) 1=Drilling axis always “Z”	381M, 383M, 384M, 387M
_ZSFI[1]	0= Rigid tapping 1= Tapping with compensating chuck 2= Deep hole tapping with chipbreaking 3= Deep hole tapping with swarf removal	384M,387M
_ZSFI[2]	Retraction speed factor (1–200%) for tapping (G74/G84)	384M
_ZSFI[3]	Polar coordinates 0 = OFF 1 = ON	381M, 383M, 384M, 387M
_ZSFI[20]	Deep hole drilling with chip breaking/removal	383T, 385T
_ZSFI[22]	Factor for retraction speed	384T
_ZSFI[23]	Dwell time with G95, 0=seconds, 1=revolution	383T
_ZSFI[24]	Number of noncuts	376T
_ZSFI[25]	Cutting edge angle	376T
_ZSFI[26]	Thread run-out distance (n*pitch)	376T
_ZSFI[27]	Min. infeed depth	376T
_ZSFI[28]	Final machining allowance	376T
_ZSFI[29]	Distance traversed for grooving cycle	374T
_ZSFI[30]	Cutting depth for stock removal cycle	371T, 372T
_ZSFI[31]	Distance traversed for stock removal cycle	371T, 372T

Table 3-2 GUD7 for cycle setting data (ISO Dialect setting data)

GUD	Description/use	
_ZSFI[32]	X axis infeed value for contour repetition	373T
_ZSFI[33]	Z axis infeed value for contour repetition	373T
_ZSFI[34]	Number of divisions for contour repetition	373T
_ZSFI[39]	G code system 2=B, 1=A, 3=C	300, 328, 330, 370T, 371T, 372T, 373T, 374T, 376T

3.3 Drilling cycles (ISO Dialect M)

3.3.1 Overview and parameter description

The drilling cycles are modal. While a drilling mode is active, you only have to program the new parameters in order to make parameter modifications.

There is **no** traversing movement if:

- No value is programmed in the NC block for X, Y and Z
- The number of repetitions K=0 was programmed

The retraction position is valid for all drilling cycles

- G98 Retraction to initial plane
- G99 Retraction to reference plane

Overview

Table 3-3 Overview of drilling cycles

External cycle call	Description
G73 X.. Y.. Z.. R.. F.. Q..	Deep hole drilling cycle with chipbreaking
G74 X.. Y.. Z.. R.. F.. P..	Counterclockwise tapping cycle
G76 X.. Y.. Z.. R.. F.. Q.. P..	Fine drilling cycle
G80	Cycle off; the cycle is also deselected by programming a G function of the 1st G group.
G81 X.. Y.. Z.. R.. F..	Drilling cycle; drilling, retraction with G00
G82 X.. Y.. Z.. R.. F.. P..	Drilling cycle; drilling, dwell, retraction with G00
G83 X.. Y.. Z.. R.. F.. Q..	Deep hole drilling cycle with swarf removal
G84 X.. Y.. Z.. R.. F.. P..	Clockwise tapping cycle
G85 X.. Y.. Z.. R.. F..	Drilling cycle; drilling, retraction with drilling feed
G86 X.. Y.. Z.. F.. R.. K..	Drilling cycle, retraction with G00
G87 X.. Y.. Z.. F.. R.. P.. Q.. K..	Reverse countersinking
G89 X.. Y.. Z.. F.. R.. P.. K..	Drilling cycle, retraction with machining feed

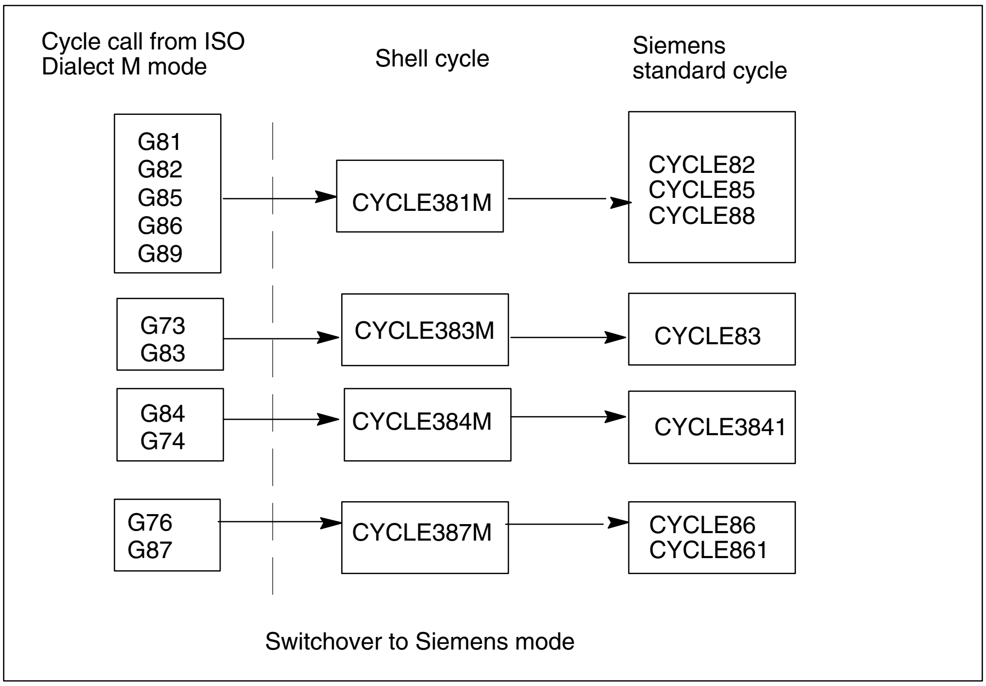


Fig. 3-2 Assignment of the cycle call in ISO Dialect M mode via shell cycle for Siemens standard cycle

Example: ISO Dialect M

```
N10 G81 X100. Z-50. R20 F100
```

G81 automatically calls the shell cycle CYCLE381M.
The calculations are performed in the shell cycle and the standard drilling cycle CYCLE81 is then called.

Parameter description

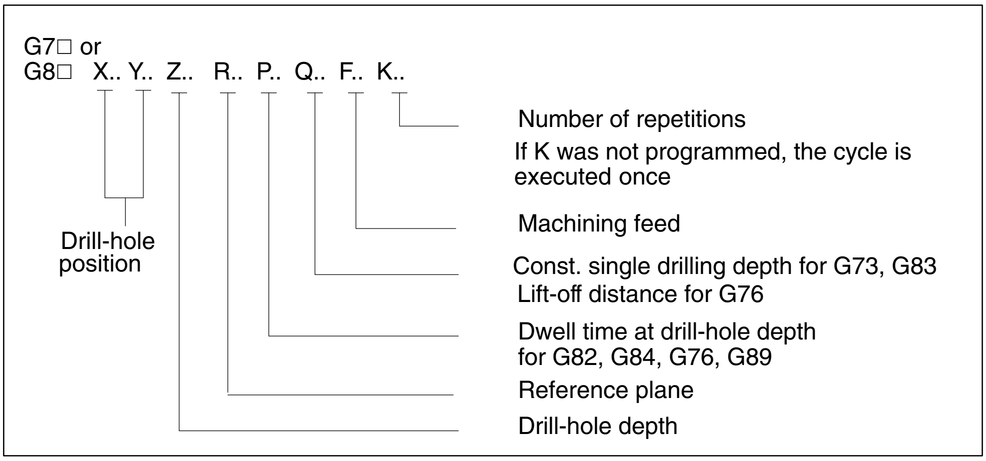


Fig. 3-3 Description of parameters allowed for G17 (X/Y plane)

Plane

Table 3-4 Definition of the plane

Defined plane	Position of hole	Depth
G17	X, Y	Z
G18	Z, X	Y
G19	Y, Z	X

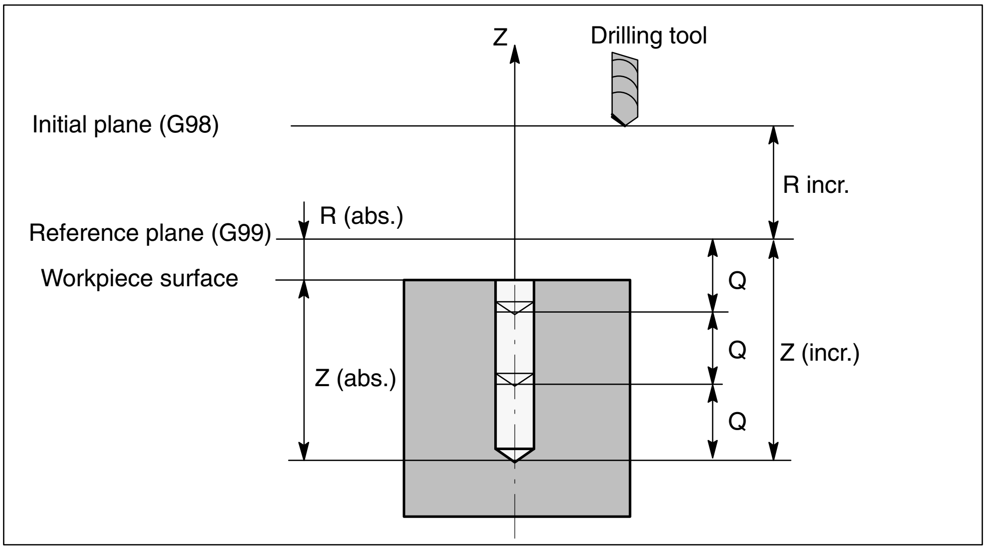


Fig. 3-4 Example of deep hole drilling cycle G83 with defined plane G17
Representation of initial plane, reference plane and parameters

3.3.2 Description of shell cycle CYCLE381M

In ISO Dialect M mode, the call is performed with G commands G81, G82, G85.

Notes

The drilling axis must be defined via GUD _ZSFI[0] (see Section 3.2).

The direction from initial plane to reference plane must be identical to the direction from reference plane to final depth.

GUD _ZSFR[0] can be used to enter a safety clearance. If the safety clearance was already allowed for when programming the reference plane, the value 0 must be entered in _ZSFR[0].

If no reference plane was programmed, the drilling is performed starting on the initial plane (current position).

The final drilling depth must be programmed. Otherwise, an alarm is displayed.

If no feedrate is programmed, the current feedrate is used as the drilling feed.

Note

Alarms are listed with their alarm number and description in Chapter "Alarms".

3.3.3 Description of shell cycle CYCLE383M

In ISO Dialect M mode, the call is performed with G commands G73, G83.

Notes

The drilling axis must be defined via GUD _ZSFI[0] (see Section 3.2).

The direction from initial plane to reference plane must be identical to the direction from reference plane to final depth.

GUD _ZSFR[0] can be used to enter a safety clearance. If the safety clearance was already allowed for when programming the reference plane, the value 0 must be entered in _ZSFR[0].

If no reference plane was programmed, the drilling is performed (in steps) starting on the initial plane (current position).

The final drilling depth and the single drilling depth Q must be programmed. Otherwise, an alarm is displayed.

If no feedrate is programmed, the current feedrate is used as the drilling feed.

Note

Alarms are listed with their alarm number and description in Chapter “Alarms”.

Deep hole drilling with chip removal

The safety clearance in the standard Siemens cycle is determined by a formula, i.e. it cannot be selected by the user.

For stock removal, retraction is to the reference plane.

Single drilling depth “Q”:

- If “Q” is missing or Q $\leq$ 0, an alarm is displayed.
- If Q > total depth, one drilling operation is executed up to the final depth.
- If Q > total depth/2, the 1st drilling operation is performed with the value of Q. The remainder is then drilled in one drilling operation.
- If Q < total depth/2, the single depth is machined until the remaining depth < Q/2. The remainder is then subdivided into 2 infeed movements of the same size.

Deep hole drilling with chipbreaking

The amount of retraction for chipbreaking is defined via GUD _ZSFR[1].

- _ZSFR[1] > 0 Retraction amount as entered
- _ZSFR[1] $\leq$ 0 Retraction amount is always 1mm

Single drilling depth “Q”:

- If “Q” is missing or Q $\leq$ 0, an alarm is displayed.
- If Q > total depth, one drilling operation is executed up to the final depth.
- If Q < total depth, the single drilling depth is machined until the remaining depth $\leq$ Q. The remainder is then machined in one drilling operation.

3.3.4 Description of shell cycle CYCLE384M

In ISO Dialect M mode, the call is performed with G commands G74, G84.

Notes

The drilling axis must be defined via GUD _ZSFI[0] (see Section 3.2).

The direction from initial plane to reference plane must be identical to the direction from reference plane to final depth.

GUD _ZSFR[0] can be used to enter a safety clearance. If the safety clearance was already allowed for when programming the reference plane, the value 0 must be entered in _ZSFR[0].

The speed of rotation during retraction can be controlled via GUD _ZSFI[2]. Example: _ZSFI[2]=80, the retraction takes place with 80 % of the drilling speed.

If no reference plane was programmed, the drilling is performed starting on the initial plane (current position).

The final drilling depth must be programmed. Otherwise, an alarm is displayed.

If no feedrate is programmed, the current feedrate is used as the drilling feed.

If the drilling feed is specified in mm/min (inch/min), the programmed feed value is converted to the appropriate revolutional feedrate, depending on the speed last programmed, and passed to the standard tapping cycle CYCLE84 as a lead value.

Note

Alarms are listed with their alarm number and description in Chapter “Alarms”.

3.3.5 Description of shell cycle CYCLE387M

In ISO Dialect M mode, the call is performed with G commands G76 and G87.

Notes

The drilling axis must be defined via GUD _ZSFI[0] (see Section 3.2).

The direction from initial plane to reference plane must be identical to the direction from reference plane to final depth.

GUD _ZSFR[0] can be used to enter a safety clearance. If the safety clearance was already allowed for when programming the reference plane, the value 0 must be entered in _ZSFR[0].

The lift-off path is always with reference to the negative direction of the first geometry axis

For plane G17	Lift-off path in –X
For plane G18	Lift-off path in –Z
For plane G19	Lift-off path in –Y

The angle must be therefore be entered such that the tool tip points in the positive direction (+) in the defined plane after the spindle stop.
Example: If plane G17 is active, the tool tip must point in the +X direction.

If no reference plane was programmed, the drilling is performed starting on the initial plane (current position).

The final drilling depth must be programmed. Otherwise, an alarm is displayed.

If no feedrate is programmed, the current feedrate is used as the drilling feed.

If no lift-off amount is programmed, Q = 0 is set. In this case, the cycle is executed without lift-off.

After retracting to the return plane, the tool is moved back to the center of the drillhole and the spindle is started in the direction of rotation for machining.

Note

Alarms are listed with their alarm number and description in Chapter “Alarms”.

3.4 Turning and drilling cycles (ISO Dialect T)

3.4.1 Turning cycles G70 to G76

Table 3-5 Overview of turning cycles

G command	Description
G70	Finishing cycle
G71	Stock removal cycle longitudinal axis
G72	Stock removal cycle transverse axis
G73	Repeat contour
G74	Deep hole drilling and recessing in longitudinal axis (Z)
G75	Deep hole drilling and recessing in facing axis (X)
G76	Multiple thread cutting cycle

The cycle parameters for G71 to G76 can comprise two G commands. Depending on the addresses programmed in the block, the values of the addresses programmed in the NC block are only saved for use in a subsequent cycle call, or the actual cycle is started. The evaluation of the two cases is performed within the shell cycle.

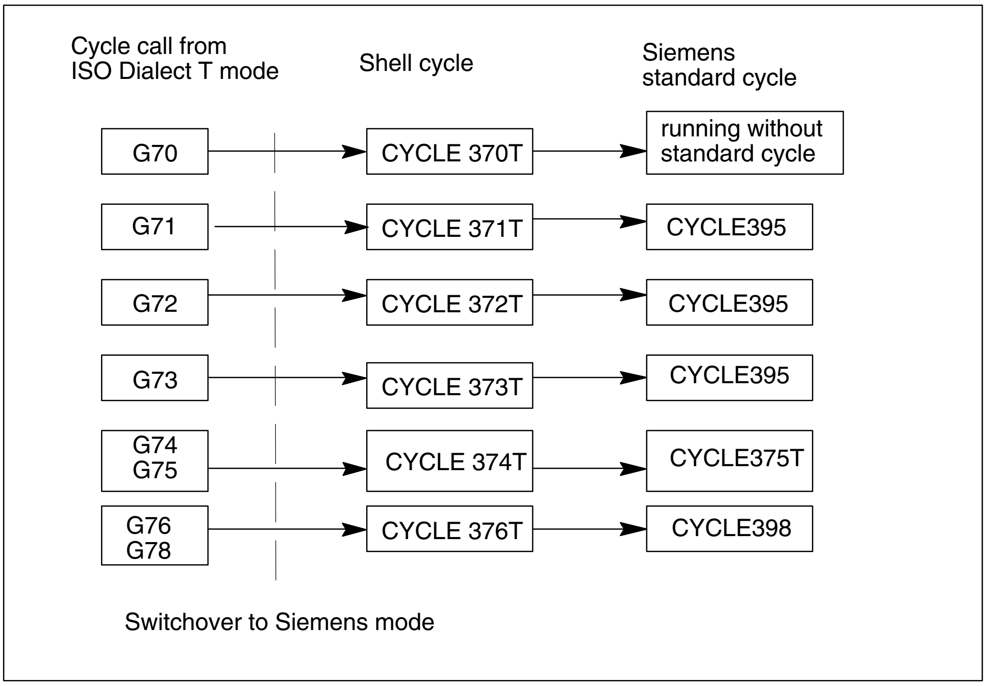


Fig. 3-5 Assignment of the cycle call in ISO Dialect T mode via shell cycle for Siemens standard cycle

Finishing cycle G70

G70 P.. Q..
- Q: Block number in main program at which the contour definition ends
- P: Block number in main program at which the contour definition begins

Fig. 3-6 Description of parameters allowed

In ISO Dialect mode, the contour in the main program is not skipped with G70. The program is always resumed at the next parts program block following the cycle call.

Example:

When the cycle is called, the contour in N20–N50 is traversed; the parts program continues running at N20 after the end of the cycle. G70 is naturally always called up after the contour definition.

```
N10 G70 P20 Q50
N20 X100. Z50.
N30 X200.
N40 Z100.
N50 X250. Z111.
N60 M30
```

Blocks N20–N50 are executed once by the finishing cycle and again by the normal program run.

Stock removal cycle, longitudinal axis G71

G71 U.. R..
- R: Retraction path
- U: Infeed depth for stock removal

Fig. 3-7 Description of parameters allowed; saving values in GUD

G71 P.. Q.. U.. W.. F.. S.. T..
- T: Tool selection
- S: Spindle speed
- F: Machining feed
- W: Final machining allowance in Z axis
- U: Final machining allowance in X axis
- Q: Block number at which the contour definition ends
- P: Block number at which the contour definition begins

Fig. 3-8 Description of parameters allowed; running cycles

With G71 in ISO Dialect T, the contour is followed with the final machining allowance in the main program after stock removal (to remove any remaining straight edges). The program is always resumed at the parts program block following the last block of the contour definition. parts program blocks between the cycle call and the first block of the contour description are not executed.

```
N10 G71 P50 Q80 U4 W3 F1000 ...
N20 G1 F0.5 G95 S1000
N30 X30. Z10.
N40 M30
N50 X100. Z50.
N60 X200.
N70 Z100.
N80 X250. Z111.
N90 M30
```

Blocks N20–N40 are skipped and are not executed. If G71 is called in the parts program after the contour definition, the program goes into an infinite loop. Allowance must be made for the different continuation patterns of the parts program in the shell cycles.

Example: G71 U6 R5 Saving values in GUD

In the example, the values programmed in the shell cycle are read from the system variables ($C_xx) and stored in channel-specific GUD.
Separate GUD are available for each cycle (G71–G76); the programmed values therefore remain valid until the next time they are programmed (in an NC block with G71–G76). Case 1 does not have to be programmed, therefore it is advisable to initialize the assigned GUD.

Example: G71 P30 Q50 U3 Starting cycle execution

In the example, the programmed values are only saved in system variables ($C_xx). The system variables are overwritten in every NC block containing a G function for a cycle call. With G71–G73, the cycle always starts running on the G command after which the “P” and “Q” are programmed. With G74–G76, the cycle starts running on the G command after which the X/U or Z/W addresses are programmed. The F, S and T commands in the call line of the cycle are also stored in system variables. A cycle-specific distinction does not therefore have to be made between the addresses. The shell cycle assigns the meaning to the parameters (e.g. for G76, the address F means pitch and not feed). For G70, the feed, speed and tool selection commands (F, S, T) from the program section of the contour definition are relevant.

The same shell cycle is always called in both cases.

Stock removal cycle, transverse axis G72

Parameters allowed: see G71 (stock removal cycle, longitudinal axis)

G72 W.. R..
- R..: Retraction path
- W..: Infeed depth for stock removal

Fig. 3-9 Description of parameters allowed; saving values in GUD

Contour repetition G73

G73 U.. W.. R..
- R..: Number of cuts parallel to the contour
- W..: Offset from start point to current tool position in Z axis
- U..: Offset from start point to current tool position in X axis

Fig. 3-10 Description of parameters allowed; saving values in GUD

G73 P.. Q.. U.. W.. F.. S.. T..
- T..: Tool selection
- S..: Spindle speed
- F..: Machining feed
- W..: Final machining allowance in Z axis
- U..: Final machining allowance in X axis
- Q..: Block number at which the contour definition ends
- P..: Block number at which the contour definition begins

Fig. 3-11 Description of parameters allowed; running cycles

Deep hole drilling and recessing in longitudinal axis G74

G74 R..
R.. – Retraction path for chipbreaking

Fig. 3-12 Description of parameters allowed; saving values in GUD

G74 X/U.. Z/W... P.. Q.. R.. F..

- F.. – Machining feed
- R.. – Retraction path at entry base
- Q.. – Infeed in longitudinal axis (Z axis) until chipbreaking
- P.. – Infeed in transverse axis (X axis) for the next recessing cut
- Z/W... – Start position in Z axis
  Z=absolute position
  W=incremental position
- X/U.. – Start position in X axis
  X=absolute position or
  U=incremental position

Fig. 3-13 Description of parameters allowed; running cycles

Note

The cycle can be used as a drilling or recessing cycle. If the cycle is used for drilling, addresses X/U and P must not be used.

Deep hole drilling and recessing in the transverse axis G75

G75 R..
- R..: Retraction path for chipbreaking

Fig. 3-14 Description of parameters allowed; saving values in GUD

G75 X/U.. Z/W.. P.. Q.. R.. F..
- F..: Machining feed
- R..: Retraction path at recess base
- Q..: Infeed in longitudinal axis (Z axis) until chipbreaking
- P..: Amount of infeed in transverse axis (X axis) until chipbreaking
- Z/W..: Start position in Z axis
  Z=absolute position
  W=incremental position
- X/U..: Start position in X axis
  X=absolute position or
  U=incremental position

Fig. 3-15 Description of parameters allowed; running cycles

Note

The cycle can be used as a drilling or recessing cycle. If the cycle is used for drilling, addresses Z/W and Q must not be used.

Multiple thread cutting cycle G76

G76 Pxxxxxxx Q.. R..

- R: Final machining allowance
- Q: Minimum infeed depth; the depth is never below this value during infeed not exceeded
- P (digits 5–6): Angle of tool cutting edge
- P (digits 3–4): Size of the chamfer at the end of the thread in 1/10x thread pitch
- P (digits 1–2): Number of finish cuts

Fig. 3-16 Description of parameters allowed; saving values in GUD

G76 X/U.. Z/W.. R.. P.. Q.. F..

- F: Lead
- Q: Amount of first infeed
- P: Thread depth/total infeed in X axis
- R: Radius difference for taper thread
- Z/W: Thread end point in Z axis
  Z=absolute position
  W=incremental position
- X/U: Thread end point in X axis
  X=absolute position or
  U=incremental position

Fig. 3-17 Description of parameters allowed; running cycles

Example for address P:

G76 P012055 Q4 R0.5

P = 012055

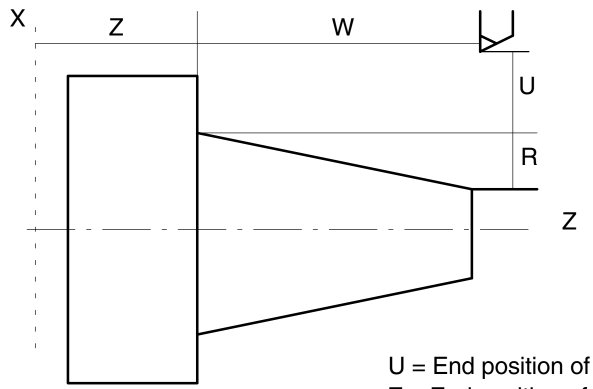
- 55: Angel of tool cutting edge = 55 degrees
- 20: Chamfer at the end of the thread = 2.0 x thread pitch
- 01: 1 finish cut

3.4.2 Turning cycles G77 to G79

Table 3-6 Overview of turning cycles G77 to G79

G command	Description
G77	Longitudinal stock removal
G78	Thread cutting
G79	Transverse stock removal

These cycle calls are modal and are called in every NC block containing axis movements. The machining movements are defined in the call parameters after the G function. The following parameters are allowed in NC blocks with cycle calls via G77–G79:



U = End position of contour in X (incr.)
Z = End position of contour in Z (abs.) or
W = End position of contour in Z (incr.)
R = Incremental radius difference to end position in X for taper turning
R = Machining feed for G77/G79 Thread lead for G78

Fig. 3-18 Parameters for cycle calls via G77 to G79

Longitudinal stock removal G77

G77 X(U).. Z(W).. R.. F..

- F: Machining feed
- R: Incremental offset of end position in X axis for taper turning
  Cylinder turning: R not programmed or R=0
- Z(W): Position in Z axis
  Z=absolute position
  W=incremental position
- X(U): Position in X axis
  X=absolute position
  U=incremental position

Fig. 3-19 Description of parameters allowed; running cycles

Thread cutting G78

G78 X(U).. Z(W).. R.. F..

- F: Lead
- R: Incremental offset of end position in X axis for taper turning
  Cylinder turning: R not programmed or R=0
- Z(W): Position in Z axis
  Z=absolute position
  W=incremental position
- X(U): Position in X axis
  X=absolute position
  U=incremental position

Fig. 3-20 Description of parameters allowed; running cycles

Transverse stock removal G79

G79 X(U).. Z(W).. R.. F..

Machining feed

Incremental offset of end position
in Z axis for taper turning
Cylinder turning: R not programmed or R=0

Position in Z axis
Z=absolute position
W=incremental position

Position in X axis
X=absolute position
U=incremental position

Fig. 3-21 Description of parameters allowed; running cycles

3.4.3 Drilling cycles G80 to G89

Table 3-7 Overview of drilling cycles

G command	Description
G80	Drilling cycle off
G83	Face deep hole drilling
G84	Face tapping
G85	Face drilling
G87	Side deep hole drilling
G88	Side tapping
G89	Side drilling

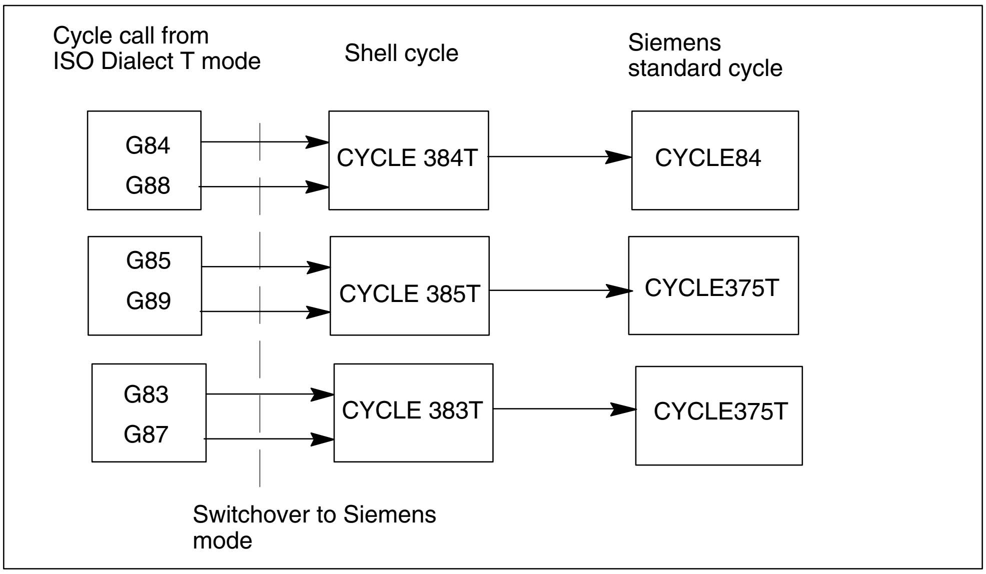


Fig. 3-22 Assignment of the drilling cycle in ISO Dialect T mode via shell cycle for Siemens standard cycle

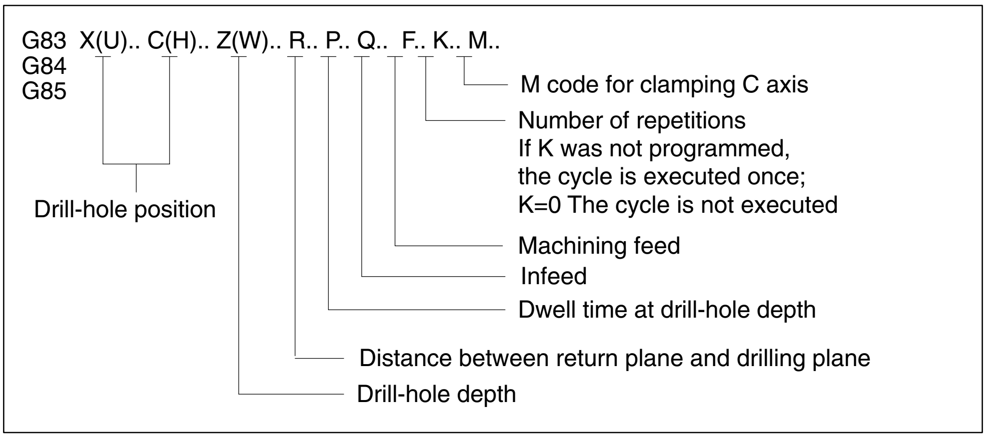


Fig. 3-23 Description of parameters allowed; running cycles

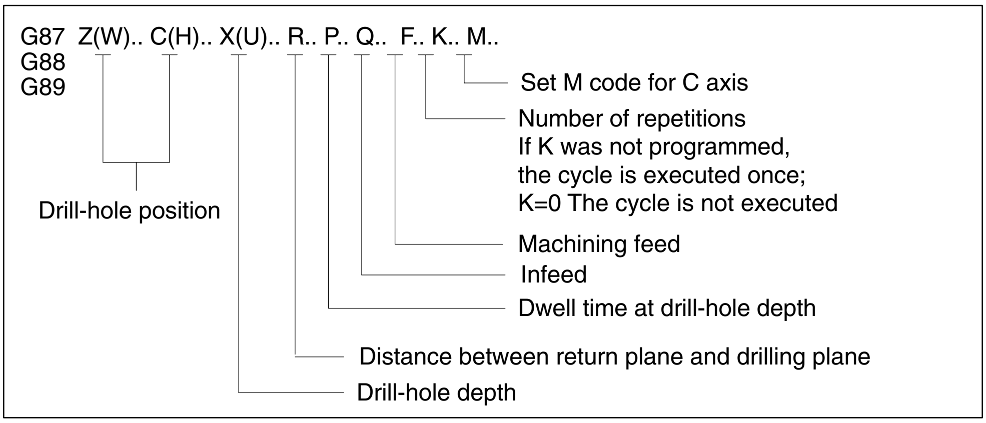


Fig. 3-24 Description of parameters allowed; running cycles

The drilling cycles are modal and are executed in every NC block in which axis movements for axes X, Y, and Z are programmed. While a drilling mode is active, you only have to program the new parameters in order to make parameter modifications. The parameters are stored in system variables $C_xx (xx = Nc address) which are read by the cycles.

The cycle is not executed if a G function of the first G group appears after the cycle G function in the same NC block. Only the axes programmed in the NC block are moved. Addresses R, Q, P, K are not copied into the system variables. The feed programmed in this block is activated.

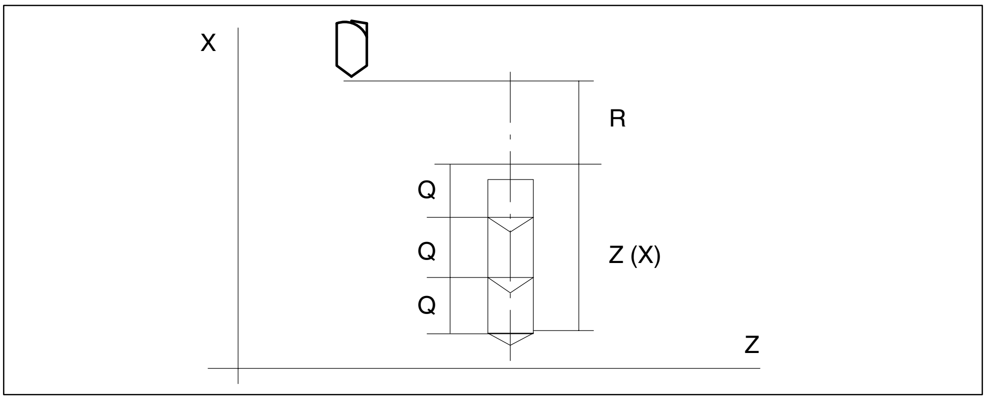


Fig. 3-25 Drilling cycle

Modal cycles

All modal cycles are deselected in ISO Dialect mode with G80 or with a G function of the first G group (G00 – G03, G33, G34, except for G77 – G79).

Cycle parameters can be programmed in the following blocks while a modal cycle is active. These parameters are copied into the system variables so that the shell cycle uses the modified parameters.

Example:

```
N10 G81 X10. Z15. R5 Q4 P10 F1000
N20 X50.                   ;Drilling cycle at position X50
```

3.4.4 Description of shell cycle CYCLE383T

In ISO Dialect T mode, the call is performed with G commands G83 and G87.

Notes

The direction from initial plane to reference plane must be identical to the direction from reference plane to final depth. Otherwise this results in an error message from the Siemens drilling cycle.

In ISO Dialect T, the programmer must take into account the safety clearance when defining the reference plane. In Siemens mode, the safety clearance to the R plane can be specified independently.
This possibility has also been implemented for the ISO cycles. GUD _ZSFR[20] can be used if necessary to enter a safety clearance. If the safety clearance was already allowed for when programming the R plane, the value “NULL” must be entered in GUD_ZSFR[20].

GUD_ZSFR[23] is used to specify whether the dwell time for G95 (only in the case of deep hole drilling) must be executed in seconds or revolutions.

A shell cycle is only permitted to be called from the external G code (G83/G87). A call in Siemens mode (after switching over with G290 and calling CYCLE383T) is not permitted.

If the G83/G87 block contains axis names other than X/Z (U/W), this results in the alarm (61811) “ISO axis name is not permitted”.

GUD_ZSFR[2] is used to determine whether the cycle is executed with chip breaking or chip removal.

In ISO Dialect, the constant individual depth is maintained until the remainder is removed on the floor of the drilled hole with an infeed. In Siemens mode, the remainder (less 2*individual depth) on the floor of the drilled hole is subdivided into two equal infeeds.

Note

Alarms are listed with their alarm number and description in Chapter “Alarms”.

3.4.5 Description of shell cycle CYCLE384T

In ISO Dialect T mode, the call is performed with G commands G84 and G88.

Notes

The direction from initial plane to reference plane must be identical to the direction from reference plane to final depth. Otherwise this results in an error message from the Siemens drilling cycle. ISO Dialect does not monitor this.

In ISO Dialect T, the programmer must take into account the safety clearance when defining the reference plane. In Siemens mode, the safety clearance to the R plane can be specified independently.
This possibility has also been implemented for the ISO cycles. GUD _ZSFR[20] can be used if necessary to enter a safety clearance. If the safety clearance was already allowed for when programming the R plane, the value “NULL” must be entered in GUD_ZSFR[20].

A shell cycle is only permitted to be called from the external G code (G84/G88). A call in Siemens mode (after switching over with G290 and calling CYCLE384T) is not permitted.

If the G84/G88 block contains axis names other than X/Z (U/W), this results in the alarm (61811) “ISO axis name is not permitted”.

The drilling speed during retraction can be controlled via GUD _ZSFI[22] (value in %).
Example: _ZSFI[22]=95, the retraction takes place with 95% of the drilling depth.

Note

Alarms are listed with their alarm number and description in Chapter “Alarms”.

3.4.6 Description of shell cycle CYCLE385T

In ISO Dialect T mode, the call is performed with G commands G85 and G89.

Notes

The direction from initial plane to reference plane must be identical to the direction from reference plane to final depth. Otherwise this results in an error message from the Siemens drilling cycle. ISO Dialect does not monitor this.

In ISO Dialect T, the programmer must take into account the safety clearance when defining the reference plane. In Siemens mode, the safety clearance to the R plane can be specified independently.
This possibility has also been implemented for the ISO cycles. GUD _ZSFR[20] can be used if necessary to enter a safety clearance. If the safety clearance was already allowed for when programming the R plane, the value “NULL” must be entered in GUD_ZSFR[20].

A shell cycle is only permitted to be called from the external G code (G85/G89). A call in Siemens mode (after switching over with G290 and calling CYCLE385T) is not permitted.

If the G85/G89 block contains axis names other than X/Z (U/W), this results in the alarm (61811) “ISO axis name is not permitted”.

If the G84 block contains axis names other than X/Z (U/W), this results in the alarm (61811) “ISO axis name is not permitted”.

Note

Alarms are listed with their alarm number and description in Chapter “Alarms”.

3.5 System variables

The names of the system variables all begin with $C_ xx. The NC address, whose value is stored in the system variable, appears in the name extension xx. The G number used to call a cycle is always stored in variable $C_G.

For all addresses, bit 0 is set in system variables $C_x_PROG if the address is programmed.

In addition, bit 1 is set in variable $C_x_PROG for axis addresses programmed incrementally.

$C_x_PROG is set to FALSE at the end of the subprogram (M17, RET).

Example 1:

```
N10 G01 G81 X100. Z-50. R20 F100
```

Shell cycle CYCLE381M for G81 is called automatically. The calculations are performed in the shell cycle and the Siemens standard drilling cycle CYCLE82 is then called. The G01 command is not required.

The values of the programmed addresses are written into the following system variables:

Address X is written to system variable $C_X;
Address Z is written to system variable $C_Z;
Address R is written to system variable $C_R;
Address F is written to system variable $C_F;

Example 2:

Axis Z is programmed in incremental dimensions (G91) → $C_Z_PROG=3
Axis Z is programmed in absolute dimensions (G90) → $C_Z_PROG =1

Example 3: Siemens shell cycle for Gxy

```
N10 PROC CYCLE377 DISPLOF          ;Block display remains at G77 block,
                                   ;Freeze G code display
N20 DEF REAL DELTA_X, pos_X, pos_Z, FEED
N30 DEF BOOL R_prog, X_prog, Z_prog
N50 DELTA_X = 0
N60 IF $C_R_PROG                   ;Only load DELTA_X if address R
N70 DELTA_X = $C_R                 ;was programmed
N75 ENDIF

N110 CYCLE...(DELTA_X, $C_X, $C_Z, $C_R_PROG, $C_X_PROG, $C_Z_PROG,
$C_F)                              ;Call Siemens cycle
N230 RET                           ;End of shell cycle
```

Table 3-8 List and description of system variables

Identifier	Type	Description
$C_A	REAL	Value of programmed address A in ISO Dialect mode for cycle programming
$C_B	REAL	Value of programmed address B in ISO Dialect mode for cycle programming
....	....	.....
$C_G	INT	G number for cycle calls in external mode
$C_H	REAL	Value of programmed address H in ISO Dialect mode for cycle programming
$C_I[ ]	REAL	Value of programmed address I in ISO Dialect mode for cycle programming and macro programming with G65/G66. Up to 10 items are possible in one block for macro programming. The values are stored in the array in the order in which they are programmed.
$C_I_ORDER[ ]	REAL	For description see $C_I[ ], used to define the programming sequence
$C_J[ ]	REAL	For description see $C_I[ ]
$C_J_ORDER[ ]	REAL	For description see $C_I[ ], used to define the programming sequence
$C_K[ ]	REAL	For description see $C_I[ ]
$C_K_ORDER[ ]	REAL	For description see $C_I[ ], used to define the programming sequence
$C_L	REAL	Value of programmed address L in ISO Dialect mode for cycle programming
....	....	....
$C_Z	REAL	Value of programmed address Z in ISO Dialect mode for cycle programming
$C_TS	STRING	String of tool name programmed at address T
$C_A_PROG	INT	Address A is programmed in a block with a cycle call. 0 = not programmed 1 = programmed (absolute) 3 = programmed (incremental)
$C_B_PROG	INT	Address B is programmed in a block with a cycle call. 0 = not programmed 1 = programmed (absolute) 3 = programmed (incremental)
....	....	....
$C_G_PROG	INT	The shell cycle call is programmed with a G function
$C_Z_PROG	INT	Address Z is programmed in a block with a cycle call. 0 = not programmed 1 = programmed (absolute) 3 = programmed (incremental)
$C_TS_PROG	INT	A tool name was programmed at address T TRUE = programmed, FALSE = not programmed

Table 3-8 List and description of system variables

Identifier	Type	Description
$C_ALL_PROG	INT	Bitmap of all programmed addresses in a block with a cycle call Bit 0 = address A Bit 25 = address Z Bit = 1 address programmed in incremental dimensions Bit = 0 address not programmed
$P_EXTGG[n]	INT	Active G code of the external language
$C_INC_PROG	INT	Bitmap of all programmed incremental addresses in a block with a cycle call Bit 0 = address A Bit 25 = address Z Bit = 1 address programmed in incremental dimensions Bit = 0 address programmed in absolute dimensions
$C_I_NUM	INT	Cycle programming: Value is always 1 if bit 0 set in $C_I_PROG. Macro programming: Number of I addresses programmed in block (max. 10).
$C_J_NUM	INT	For description see $C_I_NUM
$C_K_NUM	INT	For description see $C_I_NUM
$P_AP	INT	Polar coordinates 0 = OFF 1 = ON
$C_TYP_PRO G	INT	Bit map of all programmed addresses in a block with a cycle call Bit 0 = A Bit 25 = Z Bit = 0 axis programmed as INT Bit = 1 axis programmed as REAL
$C_PI	INT	Program number of the interrupt routine that was programmed with M96

3.6 Programming contour definitions (ISO Dialect T)

Contour definitions can be programmed in both ISO Dialect T mode and Siemens mode.

There are 3 basic shapes of contour

- A straight line
  The end point is programmed with a Cartesian coordinate and an angle
- Two straight lines
  The transition is programmed with a rounding or chamfer
- Three straight lines
  The transitions are programmed with a rounding or chamfer

In the descriptions below, indices are occasionally assigned to the address letters X, Z, A, R and C to establish a unique assignment between the NC block and the associated drawing. These indices do not appear in the NC program. The assignment is always derived uniquely from the block containing the address letter. Address letter Q is used as a placeholder for R or C, where either of these letters can appear. Q can also be omitted. In this case, no chamfer or rounding is inserted at the transition between the two linear sections.

Any number of other NC addresses can be used in blocks defining contours, e.g. address letters for further axes (individual axes or an axis perpendicular to the machining plane), auxiliary function parameters, G codes, velocities, etc.

In the following examples, it is assumed that G18 is active. Programming of contours is also possible without restriction with G17 or G19, however.

ISO Dialect mode

Address C is used in ISO Dialect mode both as an axis identifier and as an identifier for a chamfer on the contour.

Address R can be a cycle parameter or an identifier for the radius in a contour.

In order to distinguish between these two options, a "," must be placed in front of the C or R address during contour definition programming (as in ISO Dialect). A comma does not have to be entered if an angle is programmed before C or R. If a radius and a chamfer are programmed together in the same block, e.g. N333 X100 A10 C20 R15, regardless of the programming sequence, a radius will always be inserted in the contour. The chamfer is ignored.

Siemens mode

The identifiers for angle, radius and chamfer are defined by machine data in Siemens mode. This prevents the occurrence of name conflicts. A comma must not be programmed before the identifier for radius or chamfer.

Note

MD 10652 for angle: $MN_CONTEUR_DEF_ANGLE_NAME
MD 10654 for radius: $MN_RADIUS_NAME
MD 10656 for chamfer: $MN_CHAMFER_NAME
(applies in Siemens mode only)

3.6.1 End point programming with angles

If address letter A appears in an NC block, one, both or none of the axes on the active plane can be programmed.

If no axis on the active plane is programmed, the block is either the first or the second block of a contour totaling two blocks. If the block is the second block of such a contour definition, the start and end point in the active plane are identical. In this case, the contour merely consists of a movement perpendicular to the active plane.

If one axis on the active plane is programmed, either a single straight line is being described, whose end point is determined exactly by the angle and the programmed Cartesian coordinate, or we are dealing with the second block of a contour totaling two blocks. In the latter case, the missing coordinate will be set equal to the last (modal) position reached.

If two axes on the active plane are programmed, the block is the second block of a contour comprising two blocks. If the current block was not preceded by a block with angle programming without programmed axes of the active plane, the block is not permitted.

Angle A must only be programmed with linear or spindle interpolation. (Spline interpolation only in Siemens mode.)

Alarms are generated in the following situations:

- In a contour consisting of two blocks, the active plane was changed during the transition from the first to the second block.
- In a contour consisting of two straight lines, a valid intermediate point cannot be generated from the programmed angles.
- Neither linear nor spline interpolation is active in a block with address A.
- A block with address A without a programmed axis on the active plane is not followed by a block with which the end point of the contour can be determined. This is the case if the block is the last block in a program or if the following block contains a preprocessor stop.

- No angle was programmed in the second block of a contour consisting of two straight lines.
- Both axes on the active plane are programmed in a block with address A which is not the second block of a contour consisting of two straight lines.
- Programmed Cartesian coordinate and programmed angle are incompatible.

3.6.2 Straight line with angle

The end point is defined by specifying the angle A and one of the two coordinates X_2 or Z_2.

Programming syntax:

X_2.. A.. or
Z_2.. A..

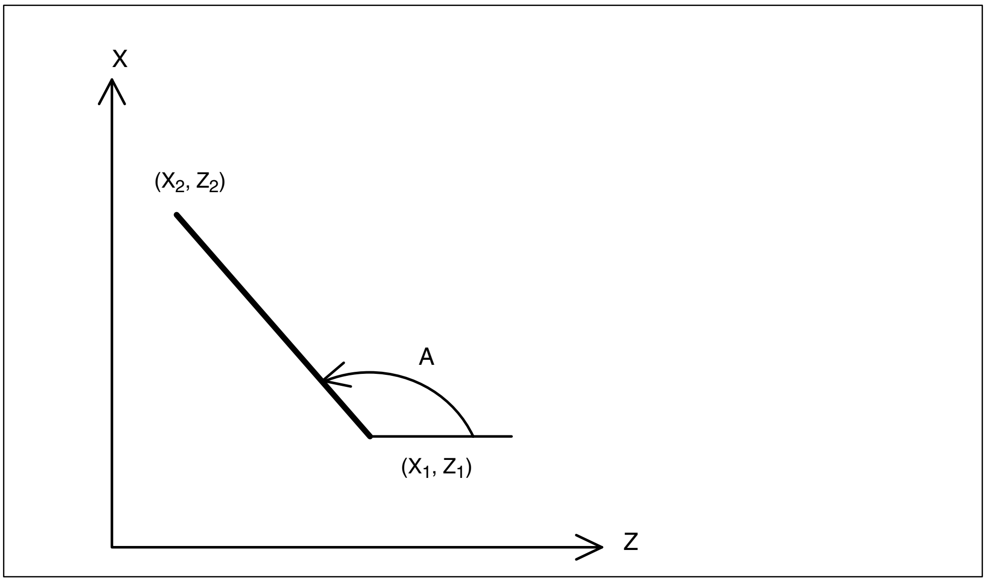


Fig. 3-26 Straight line with angle

Example (Fig. 3-27):

Programming in ISO Dialect T mode:

```
N10 G1 X5. Z70. F1000 G18
N 20 X88.8 A 110 or (Z39.5 A110)
```

Programming in Siemens mode:

```
N10 X5. Z70. F1000 G18
```

```
N20 X88.8 ANG=110 or (Z39.5 ANG=110)
```

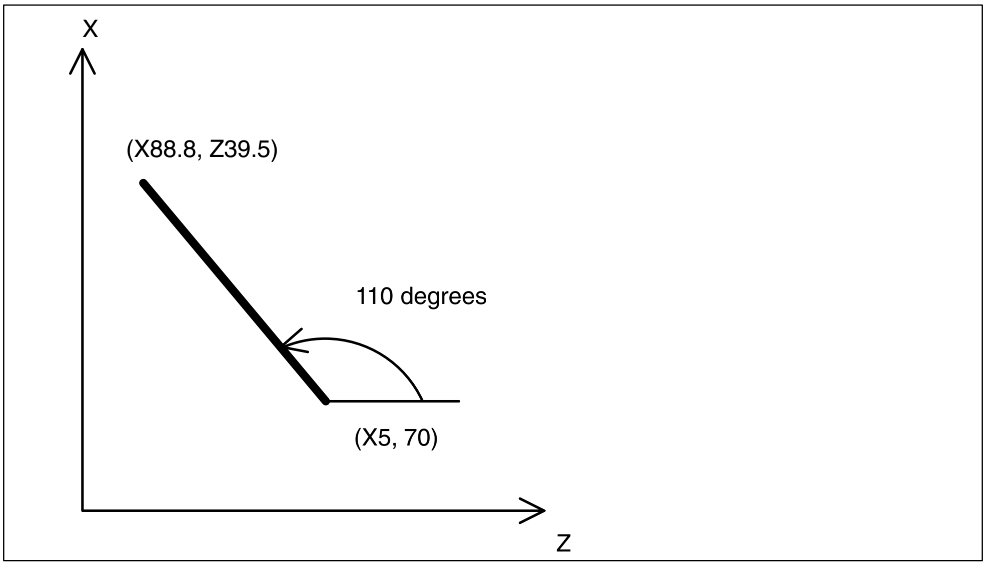


Fig. 3-27 Straight line with angle

3.6.3 Two straight lines

The end point of the first straight line can be programmed either by specifying the Cartesian coordinates or by specifying the angle of the two straight lines relative to the abscissa.

Programming syntax:

N10 A_1..   (Q..)
N20 X_3.. Z_3.. A_2..

or

N10 X_1.. Z_1.. (Q..)
N20 X_3.. Z_3..

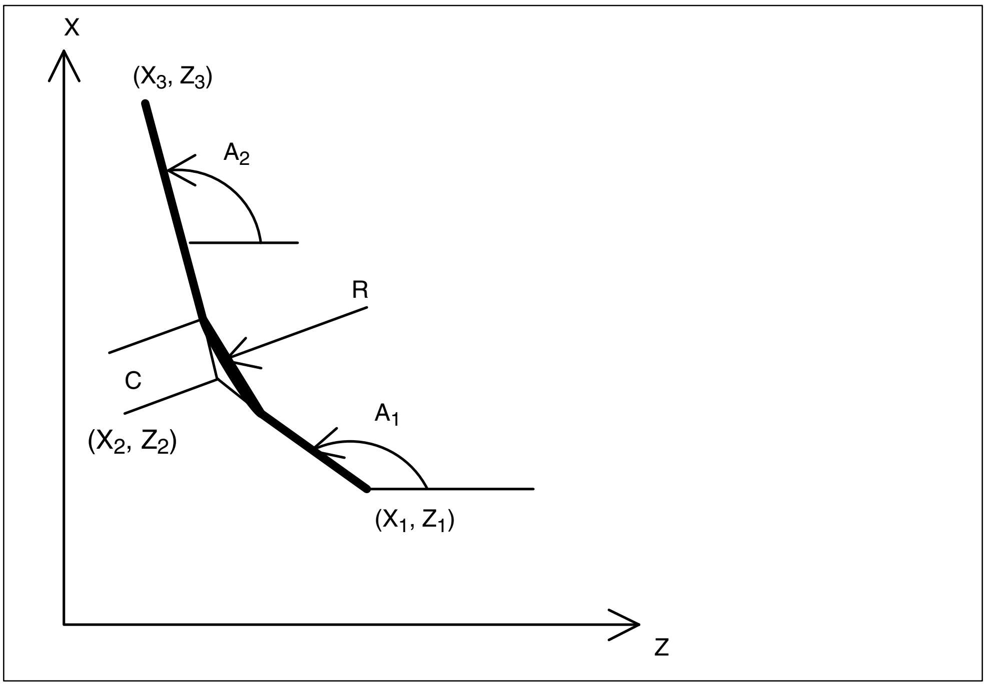


Fig. 3-28 Two straight lines

Example (Fig. 3-29):

Programming in ISO Dialect T mode:

```
N10 G1 X10. Z80. F1000 G18
N20 A 1.48.64 C5.5
N30 X85. Z40. A100
```

Programming in Siemens mode:

```
N10 X10. Z80. F1000 G18
N20 ANG=148.65 CHR=5.5
N30 X85. Z40. ANG=100
```

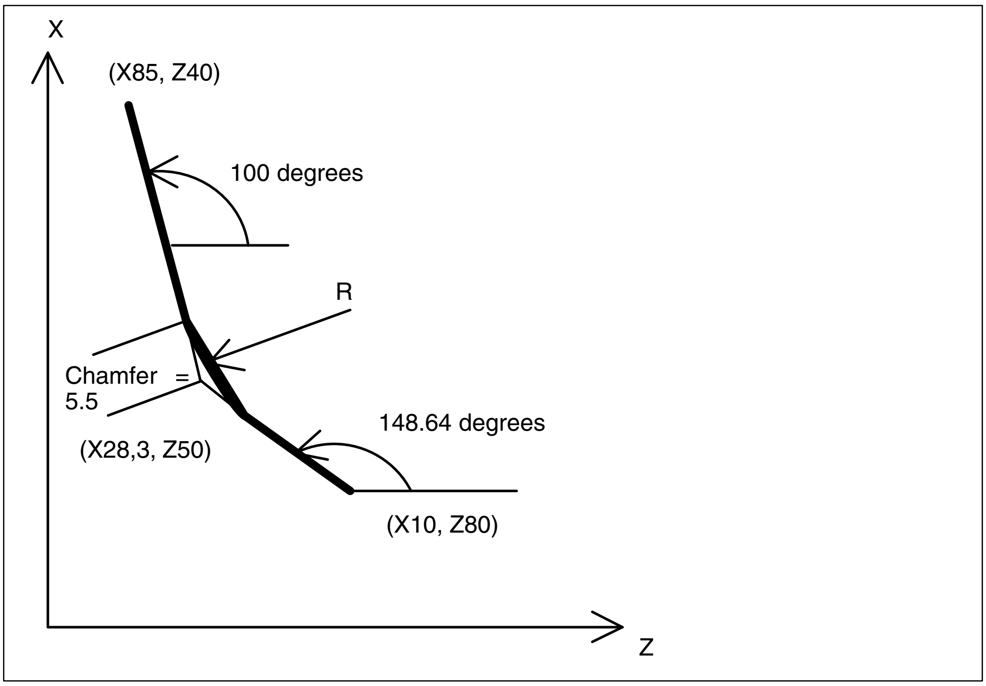


Fig. 3-29 Two straight lines

3.6.4 Three straight lines

The end point of the third straight line must be programmed with Cartesian coordinates. Either a chamfer or a radius can be used for the transition from the second to the third coordinate.

This type of programming can be used for any number of further blocks, i.e. a distinction does not have to be made between contours with two or more blocks.

Programming syntax:

N10 X_2.. Z_2.. (Q_1..)
N20 X_3.. Z_3.. (Q_2..)
N30 X_4.. Z_4..

or

N10 A_1.. (Q_1..)
N20 X_3.. Z_3.. A_2.. (Q_2..)
N30 X_4.. Z_4..

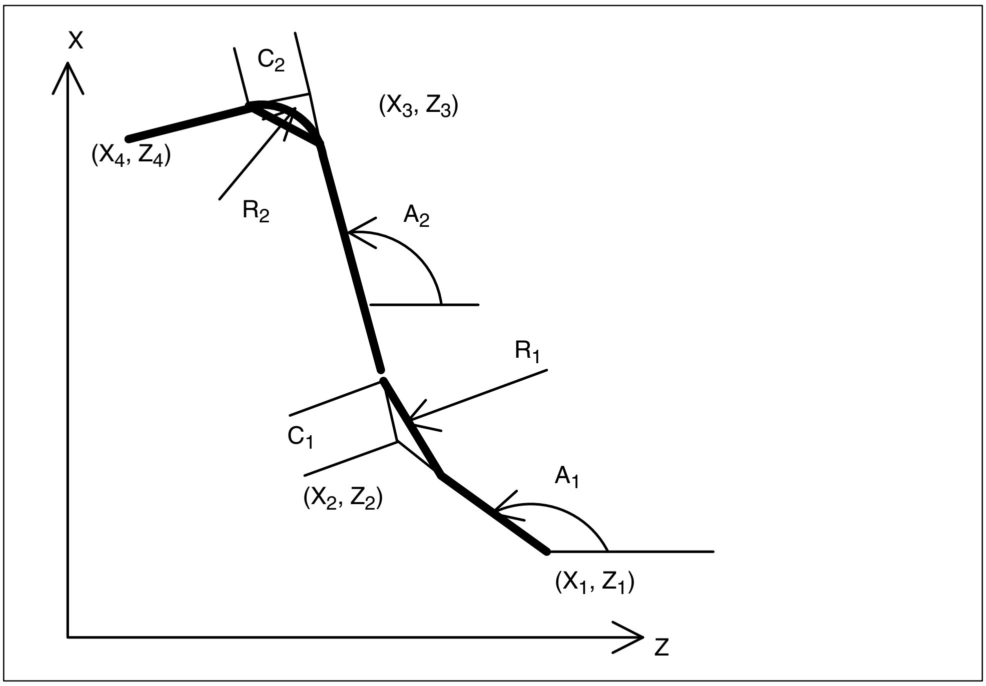


Fig. 3-30 Three straight lines

Example (Fig. 3-31):

Programming in ISO Dialect T mode:

```
N10 G1 X10. Z100. F1000 G18
N20 A140 C7.5
N30 X80. Z70. A95.824, R10
```

Programming in Siemens mode:

```
N10 X10. Z100. F1000 G18
N20 ANG=140 CHR=7,5
N30 X80. Z70. ANG=95.824 RND=10
```

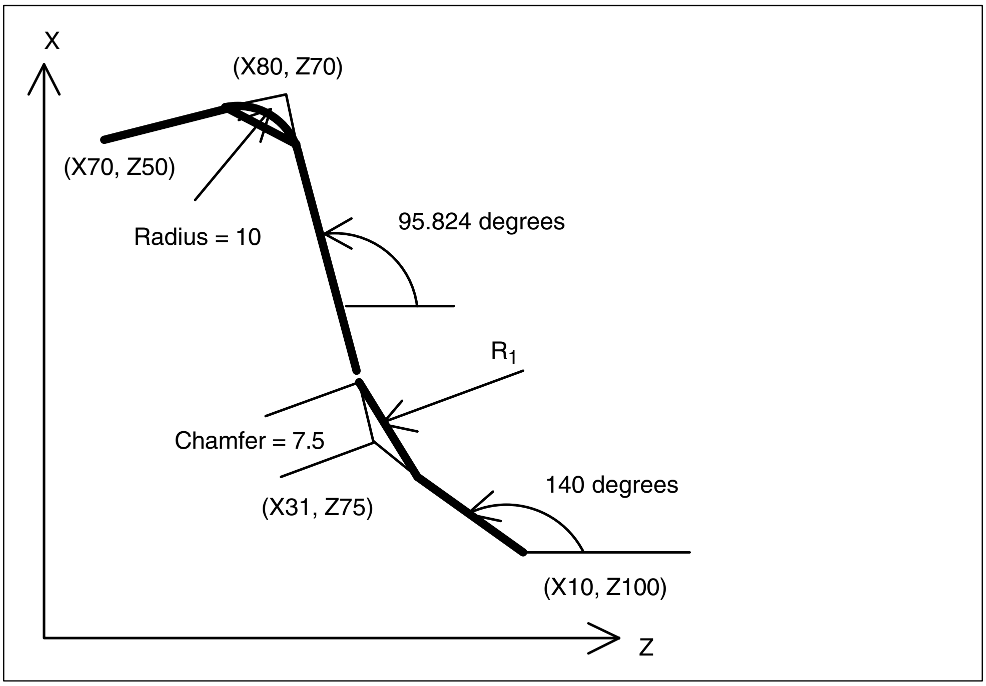


Fig. 3-31 Three straight lines

3.6.5 Polygon turning with G51.2

Function G51.2 allows two spindles to be linked so that multi-edge workpieces can be made. This corresponds to the synchronous spindle function in Siemens mode with a speed ratio other than 1 : 1.

Programming syntax G51.2 Q.. P.. R.. enables synchronous spindle coupling. The speed ratio for leading spindle to following spindle is defined in parameters “Q” and “P”. If it is intended that linking should be activated using an angular offset for the following spindle and leading axis, the angular difference needs to be programmed using the address “R”.

When mapping the function into Siemens language, two parts program commands must always be drawn up in order to activate the synchronous spindle function, and these commands must not be in the same block.
One parts program command is used to define the assignment of leading and following spindles together with the speed ratio and coupling type (COUPDEF(..)). The second parts program command activates coupling complete with the programmed angular offset (COUPON(..)). To execute both these program commands, G51.2 is used to call a cycle (CYCLE3512). The programmed values are transferred in cycle parameters $C_P, $C_Q and $C_R. G50.2 is used to turn coupling off again (also using CYCLE3512).

When programming G51.2 the first spindle in the channel is always defined as the leading spindle and the second spindle is defined as the following spindle. Setpoint linkage is selected as the coupling type.

Example

```
N10 T1234
N20 G0 X10 Z100 M3 S1000
N30 G51.2 P1 Q3      ; Start synchronous spindle using speed ratio 1 : 3 and
                     ; angular offset 0 degrees
Nxx ....
N1000 G51.2 R180     ; Angular offset between leading and following spindle
                     ; to be 180 degrees

N1200 G50.2          ; Deactivate synchronous spindle mode
N2000 M30
```

Please refer to the following documentation for a detailed description of the synchronous spindle function:

/FB2/ SINUMERIK 840D/810D(CCU2)
Description of Functions, Extended Functions, Chapter M1 and
/PGA/ SINUMERIK 840D/810D
Programming Guide, Advanced, Chapter “Synchronous Spindle”

3.6.6 Contour repetition G72.1 / G72.2

A subprogram programmed under the address P.. is called when code G72.x is used. Address L.. is used to define the number of times this subprogram is repeated. If address L is not programmed, the subprogram is executed once. Before each subprogram call, and depending which G code is used, either a coordinate rotation is executed (G72.1) or an incremental path is traversed by reference to the starting point of the contour (G72.2).

G72.1

G72.1 repeatedly calls a subprogram in which the contour to be repeated is programmed. Before each subprogram call the coordinate system is rotated through a defined angle.

This function is performed by calling a cycle (CYCLE3721). The programmed values are transferred to the cycle in the $C_.. cycle parameters. The G function number is held in $C_G. The value 721 is entered in $C_G for G72.1 and the value 722 is entered in $C_G for G72.2. The cycle executes coordinate rotation n times and calls the subprogram n times. The coordinate system is rotated about the vertical axis of the selected plane.

X.. Y.. (Z..) Reference point for coordinate rotation
P.. Subprogram number

L.. Number of times the subprogram is repeated
R.. Angle of rotation

Example:

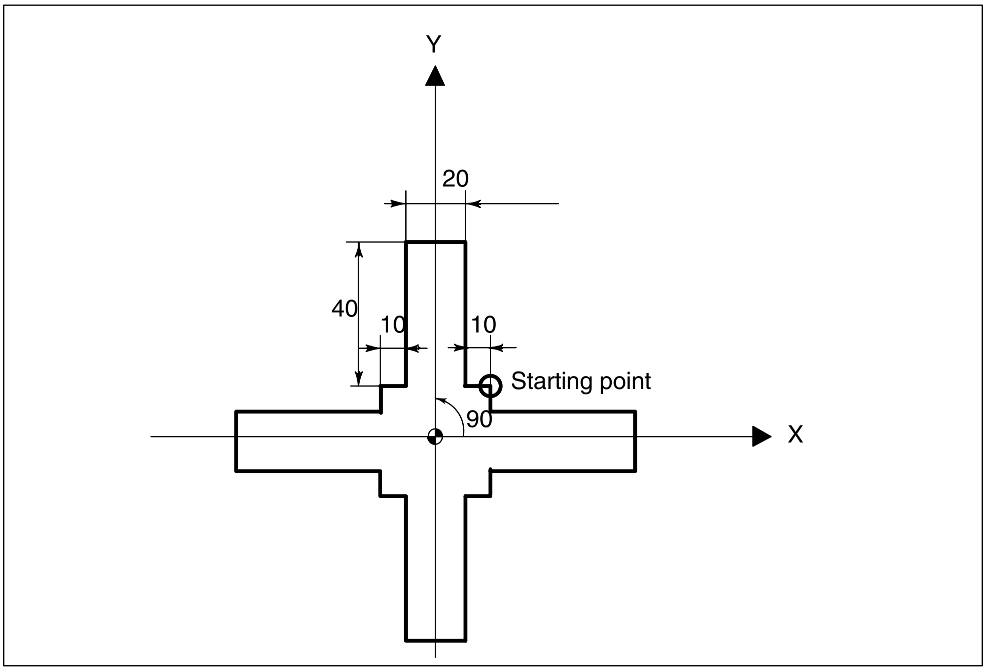


Fig. 3-32 Contour repetition using G72.1

Main program

```
N10 G92 X40.0 Y50.0 ;
N20 G01 G90 G17 G41 20 Y20 D01 F1000
N30 G72.1 P1234 L4 X0 Y0 R90.0
N40 G40 G01 X100 Y50 Z0
N50 G00 X40.0 Y50.0 ;
N60 M30 ;
```

Subprogram 1234.spf

```
N100 G01 X10
N200 Y50
N300 X-10
N400 Y10
N500 X-20
N600 M99
```

G72.2

G72.2 repeatedly calls a subprogram in which the contour to be repeated is programmed. Before each subprogram call, the axes programmed using I, J, K are incrementally traversed. The programmed G function within CYCLE3721 detects whether the contour subprogram will be repeated after a rotation or a linear movement.

This function is performed by calling a cycle. The programmed values are transferred to the cycle in the $C_.. cycle parameters. The cycle calls the subprogram n times. Before each subprogram call, a path programmed using I, J, K is traversed incrementally by reference to the starting point.

I.. J.. K.. Position to which the X, Y, Z axes will traverse before the subprogram is called. The position is calculated by reference to the starting point of the subprogram.

P.. Subprogram number

L.. Number of times the subprogram is repeated

Example:

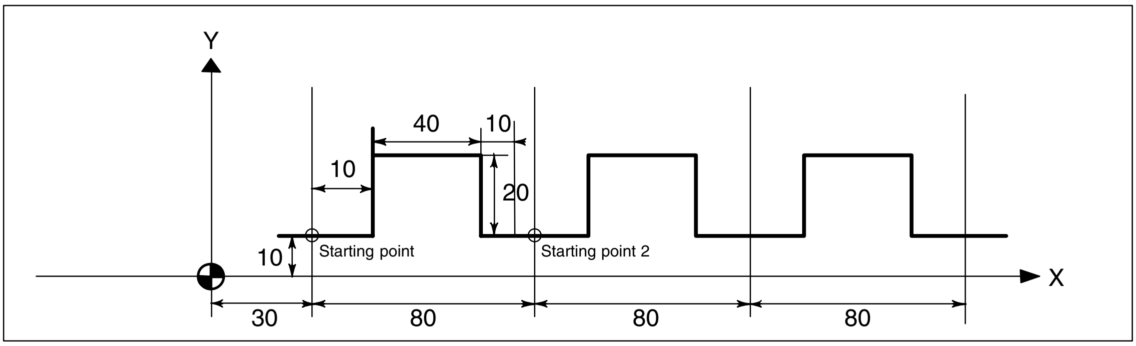


Fig. 3-33 Contour repetition using G72.2

```
N10 G00 G90 X0 Y0
N20 G01 G17 G41 X30. Y0 D01 F1000
N30 Y10.
N40 X30.
N50 G72.2 P2000 L3 I80. J0

O2000 G90 G01 X40.
N100 Y30.
N200 G01 X80.
N300 G01 Y10.
N400 X90.
N500 M99
```

■

Start-Up

4

4.1 Machine data

Activating ISO Dialect mode

MD 18800: $MN_MM_EXTERN_LANGUAGE

Note

Switchover to an external programming language is an option.

Selection of ISO Dialect M or T

MD 10880: $MN_MM_EXTERN_CNC_SYSTEM = 1 ISO Dialect M
MD 10880: $MN_MM_EXTERN_CNC_SYSTEM = 2 ISO Dialect T

Axis names in ISO Dialect T

The maximum number of possible axes is 8. The axis identifiers for the first 2<NBS>axes are fixed as X and Z. Subsequent axes can have the identifiers Y, A, B, C, U, V and W.

If G code system A is active (G91 is not available in this case), incremental values for X, Z and Y must be programmed with U, V and W. It is not possible to use U, V and W as axis designations, so the maximum number of axes is reduced to 6. In the case of auxiliary function H, the C axis traverses incrementally.

If B is not used as an axis identifier, B can be used as an extended auxiliary function. B is then output as auxiliary function H with the address extension 1 (H1=).

Contour definition:
Machine data 20734: $MC_EXTERN_FUNCTION_MASK, Bit 0 is used to specify programming of the contour definition.

0: Contour definition is programmed with ,C ,R ,A (with commas).
C and A can be axis identifiers.
1: Contour definition is programmed with C R A (without commas).
C and A cannot be axis identifiers.

Axis names in ISO Dialect M

The maximum number of possible axes is 8. The axis identifiers for the first 3 axes are fixed as X, Y and Z. Subsequent axes can have the identifiers A, B, C, U, V and W.

If B is not an auxiliary function, B can be used as an axis identifier.

Axis interpolation

All programmable axes interpolate with each other by default in ISO Dialect.

This is equivalent to FGROUP for ISO Dialect M: X, Y, Z (A, B, C, U, V, W).
This is equivalent to FGROUP for ISO Dialect T: X, Z, Y (C).

This behavior is achieved with machine data
$MC_FGROUP_DEFAULT_AXES[0] if the machine has 4 axes:

$MC_FGROUP_DEFAULT_AXES[0] = 1
$MC_FGROUP_DEFAULT_AXES[1] = 2
$MC_FGROUP_DEFAULT_AXES[2] = 3
$MC_FGROUP_DEFAULT_AXES[3] = 4

Zero offsets (ISO Dialect M only)

If only zero offsets G54 to G59 are to be used, MD
28080: $MC_MM_NUM_USER_FRAMES or, for global frames, MD
18601: $MC_MM_NUM_GLOBAL_USER_FRAMES >= 7 must be set.

If G54 is to be active after a reset, the following MDs must be set to 1:
20154: $MC_EXTERN_GCODE_RESET_VALUES[13]
20150: $MC_GCODE_RESET_VALUES[7]

If the extended zero offsets G54 P1 to P48 are used, MD
28080: $MC_MM_NUM_USER_FRAMES or, for global frames, MD
18601: $MC_MM_NUM_GLOBAL_USER_FRAMES >= 55 must be set.

If the following MDs are set to 7
20154: $MC_EXTERN_GCODE_RESET_VALUES[13]
20150: $MC_GCODE_RESET_VALUES[7]
G54 P1 is active after a reset. G54 P1 is mapped onto Siemens G507.

If a selected zero offset is not to be traversed with G91,
the following setting data must be set to 0
42440: $SC_FRAME_OFFSET_INCR_PROG

Suppress tool length and radius compensation with G53:
10760: $MN_G53_TOOLCORR = 1

Metric/inch switchover

The handwheel and increment weighting are not switched over with G20 and G21. This switchover must be initiated from the PLC: see MD $MA_JOG_INCR_WEIGHT

In ISO Dialect mode, the zero offsets are converted on switchover. In ISO Dialect Original, only a decimal point shift is performed.

Programming diameter or radius

20150: MD $MC_GCODE_RESET_VALUES[28] = 2
activates diameter programming for the transverse axis.

Decimal point programming

With machine data $MN_EXTERN_FLOATINFPOINT_PROG you can choose between standard notation and pocket calculator notation.

Input resolution IS-B and IS-C is selected with machine data 10886: $MN_EXTERN_INCREMENT_SYSTEM.

Please make sure that the settings in machine data MD 10200 $MN_INT_INCR_PER_MM and 10210 $MN__INT_INCR_PER_DEG allow the IS-B or IS-C resolution to be calculated.
Otherwise the programmed values are rounded.

Example: IS-C mm $MN_INT_INCR_PER_MM = 10000.

Scaling

22910: $MC_WEIGHTING_FACTOR_FOR_SCALE=0 0.001
22910: $MC_WEIGHTING_FACTOR_FOR_SCALE=1 0.00001

Axial scale factor: 43120: $MA_DEFAULT_SCALE_FACTOR_AXIS

Scale factor P: 42140: $MC_DEFAULT_SCALE_FACTOR_P

Enable axial scaling: 22914: $MC_AXES_SCALE_ENABLE = 1
(axial scaling is not possible when = 0)

Position in machine coordinate system G53

The axis velocity for positioning with G53 and with G00 without interpolation is defined in MD 32060: $MA_POS_AX_VELO.

With G53 X.. Y... a position in the machine coordinate system is approached. The axes do not interpolate with each other, but each axis travels separately at maximum speed to the programmed position.

Incremental positions are skipped in block G53. As long as tool radius compensation or tool length compensation is active (G41/G42, G43/G44) the axes **do not** traverse separately, they interpolate with each other.

Set actual value G92

Delete G92 offset on Power ON:
24004: $MC_CHBFRAME_POWERON_MASK=1

G92 is retained on reset (M30, channel reset):
20110: $MC_RESET_MODE_MASK bit0 and bit14=1

Resetting the tool coordinate system G92.1

With G92.1 X.. (G code system A: G50.3 P0) an offset coordinate system can be reset to its position before resetting. That resets the workpiece coordinate system to the coordinate system that is defined by the active settable zero offsets (G54–G59). If no settable zero offset is active, the workpiece coordinate system will be set to the reference position. G92.1 resets offsets set with G92 or G52. But only axes that have been programmed are reset.

Example 1:

```
N10 G0 X100 Y100    ;display: WCS: X100 Y100   MCS: X100 Y100
N20 G92 X10 Y10     ;display: WCS: X10 Y10     MCS: X100 Y100
N30 G0 X50 Y50      ;display: WCS: X50 Y50     MCS: X140 Y140
N40 G92.1 X0 Y0     ;display: WCS: X140 Y140   MCS: X140 Y140
```

Example 2:

```
N10 G10 L2 P1 X10 Y10
N20 G0 X100 Y100    ;display: WCS: X100 Y100   MCS: X100 Y100
N30 G54 X100 Y100   ;display: WCS: X100 Y100   MCS: X110 Y110
N40 G92 X50 Y50     ;display: WCS: X50 Y50     MCS: X110 Y110
N50 G0 X100 Y100    ;display: WCS: X100 Y100   MCS: X160 Y160
N60 G92.1 X0 Y0     ;display: WCS: X150 Y150   MCS: X160 Y160
```

Delete distance to go

In ISO Dialect Original, delete distance to go is enabled with G31. The distance to go is only deleted in this block if the PLC signal is active on the channel. The signal is not evaluated without G31.

In ISO Dialect mode, the PLC signals are evaluated in every block, irrespective of G31. G31 activates probe1.

In ISO Dialect, the deleted distance to go can be calculated via the PLC Var selector.

The function G31 P1 (..P4) only differs from G31 by the fact that different inputs can be selected for the measurement signal with P1 – P4. That way it is also possible to monitor several inputs simultaneously for a rising edge of a measurement signal. Assignment of inputs to addresses P1 –P4 is defined in machine data 10810: $MN_EXTERN_MEAS_G31_P_SIGNAL[0 .. 3].

Note

On 840D only two measurement inputs are available.

Spindle position

The spindle position for M19 is set via setting data 43240: $SC_M19_SPOS.

Protection zone

A protection zone must exist if G commands G22 and G23 are used:

18190: $MN_NUM_PROTECT_AREA_NCK = 1
28210: $MC_NUM_PROTECT_AREA_ACTIVE = 1

Auxiliary functionoutput

If the H value is to be output as an integer value to the PLC,
MD 22110: $MC_AUXFU_H_TYPE_INT must be set to 1.

The timing of auxiliary function output (M, S, T, H) to the PLC can be set in MD:

0 = Auxiliary function output before motion

1 = Auxiliary function output during motion

2 = Auxiliary function output after motion

3 = No output of motion to PLC

22200: $MC_AUXFU_M_SYNC_TYPE for M functions
22210: $MC_AUXFU_S_SYNC_TYPE for S functions
22220: $MC_AUXFU_T_SYNC_TYPE for T functions
22230: $MC_AUXFU_H_SYNC_TYPE for H functions

1st reference point approach G28

The following machine data must be set:

20050: $MC_AXCONF_GEOAX_ASSIGN_TAB[0–2]
Axis 1 to 3

20060: $MC_AXCONF_GEOAX_NAME_TAB[0–2]
Axis names for milling: X, Y, Z
Axis names for turning: X, Z, Y

200070: $MC_AXCONF_MACHAX_USED[0–3]
Axis 1 to 4

20080: $MC_AXCONF_CHANAX_NAME_TAB[0–3]
4th axis name for milling: X, Y, Z are permanently defined; A, B, C, U, V or W can also be selected.
4th axis name for turning: X, Y, Z are permanently defined; C can also be selected

20100: $MC_DIAMETER_AX_DEF
For turning only: X axis (available soon)

20150: $MC_GCODE_RESET_VALUES[28]
Radius or diameter programming
1= DIAMOF (radius for G90/G91)
2 = DIAMON (diameter for G90/G91)
3 = DIAM90 (diameter for G90, radius for G91)
Note: DIAM90 is handled like DIAMON within the cycle.

34100: $MA_REFP_SET_POS[0]
0 = 1st reference point
Enter a value for each axis

35000: $MA_SPIND_ASSIGN_TO_MACHAX
0 = Axis is not a spindle
1 = Axis is a spindle

$2^{nd}/3^{rd}/4^{th}$ reference point approach G30

The following machine data must be set:

20050: $MC_AXCONF_GEOAX_ASSIGN_TAB[0–2]
Axis 1 to 3

20060: $MC_AXCONF_GEOAX_NAME_TAB[0–2]
Axis names for milling: X, Y, Z
Axis names for turning: X, Z, Y

20070: $MC_AXCONF_MACHAX_USED[0–3]
Axis 1 to 4

20080: $MC_AXCONF_CHANAX_NAME_TAB[0–3]
4th axis name for milling: X, Y, Z are permanently defined; A, B, C, U, V or W can also be selected.
4th axis name for turning: X, Y, Z are permanently defined; C can also be selected

20100: $MC_DIAMETER_AX_DEF
For turning only: X axis (available soon)

20150: $MC_GCODE_RESET_VALUES[28]
Radius or diameter programming
1 = DIAMOF (radius for G90/G91)
2 = DIAMON (diameter for G90/G91)
3 = DIAM90 (diameter for G90, radius for G91)

Note: DIAM90 is handled like DIAMON within the cycle.

34100: $MA_REFP_SET_POS[1,2,3]
1,2,3 = 2nd, 3rd, 4th reference point
Enter a value for each axis

35000: $MA_SPIND_ASSIGN_TO_MACHAX
0 = Axis is not a spindle
1 = Axis is a spindle

G30.1 floating reference position

The reference point approach is executed in the CYCLE328 cycle. The position of the reference point is defined in the setting data
43340: $SC_EXTERN_REF_POSITION_G30_1.

4.1.1 Active G command to PLC

The user can select the G groups of an external language with MD 22512: $MC_EXTERN_GCODE_GROUPS_TO_PLC; the active G command is then signaled to the PLC for these groups.

$MC_EXTERN_GCODE_GROUPS_TO_PLC[0..7]=0

4.1.2 Tool change, tool data

A cutting edge is not selected on a tool change.
20270: $MC_CUTTING_EDGE_DEFAULT = 0

Setting data: The offset is not traversed on a tool selection with G91
42442: $SC_TOOL_OFFSET_INCR_PROG = 0

Tool length offsets are permanently assigned to geometry axes:
Length 1: Z
Length 2: Y
Length 3: X
42940: $SC_TOOL_LENGTH_CONST = 17

Tool length compensation remains active after reset:
20110: $MC_RESET_MODE_MASK = ’B1000000’

Tool offset takes effect on programming of T/H/D, not with M6
22550: $MC_TOOL_CHANGE_MODE = 0

4.1.3 G00 always with exact stop

At high velocities when contouring is active in G00 mode, collisions can occur due to approximate positioning. Machine data 20734: $MC_EXTERN_FUNCTION_MASK, bit 4 is used to specify the exact stop response for G00.

20734: $MC_EXTERN_FUNCTION_MASK, bit 4=0, G00 traverses with the currently active exact stop function. If G64 is active, the G00 blocks will also be traversed with G64.

20734: $MC_EXTERN_FUNCTION_MASK, bit 4=1, every G00 block with a traversing motion will be traversed with G09 (non-modal exact stop). Even if G64 is active, non-modal exact stop will be effective in every G00 block.

4.1.4 Response to syntax errors

Machine data 20734: $MC_EXTERN_FUNCTION_MASK, bit 3 is used to specify the response to errors that are detected in the first part of the ISO translator. The complete ASCII block is checked here.

If bit3==0, when unknown addresses are found, an NC alarm will be output and further processing is stopped.

If bit3==1, an alarm is not output and the ASCII block is transferred to the Siemens translator. In the Siemens translator, an attempt is made to compile the block and the subsequent NC block is then sent to the ISO translator first.
This allows the user to program unique Siemens blocks while ISO mode is active without having to switch to Siemens mode using G290.

In the G code window, the current ISO G code is displayed. There is no switchover to Siemens mode.
If a G function is activated in a block of this type in the Siemens translator that can be directly converted to an ISO G code, the G code will be updated here.

Example

20734: $MC_EXTERN_FUNCTION_MASK, bit3==0

```
N5 G291        ;ISO mode
N10 WAIT       ;Alarm 12080 "WAIT unknown"
N15 G91 G500   ;Alarm 12080 "G500 unknown"
```

20734: $MC_EXTERN_FUNCTION_MASK, bit3==1

```
N5 G291        ;ISO mode
N10 WAIT       ;Block will be processed by the Siemens
               ;translator
N15 G91 G500   ;Block will be processed by the Siemens
               ;translator
N20 X Y        ;Block will be processed by the ISO translator
               ;due to G291 G91 from N15 is active
```

Note

Programming mistakes in ISO mode can result in undesirable reactions.
Example for ISO M:

Programming required	G90 G76	;modal cycle call
But	G90 G75	;is entered instead

G75 does not exist in ISO M mode, so the block is transferred to the Siemens translator where it causes G75 "Traverse to fixed point" without any prompt or alarm being output.

4.1.5 Selection of code system A, B, C (ISO Dialect T)

In ISO Dialect T, a distinction is made between G code systems A, B, and C. G<NBS>code system B is active by default. Switchover was performed until now using a cycle that used the function "Rename G codes" by means of machine data 10712: $MN_NC_USER_CODE_CONF_NAME_TAB.
The disadvantage of this method is that the cycle has to be adapted when extensions are made to the G code. Apart from this, the function "Rename G codes" is no longer available to the user.

Software 6

The default G code system continues to be G code system B.

Machine data 10881: $MN_MM_EXTERN_GCODE_SYSTEM is used to select code system A, B or C. The function "Rename G codes" is no longer used by this function so the user is now able to apply this function without any restrictions.

Switching over using a cycle as before is still possible.

$MN_MM_EXTERN_CNC_SYSTEM = 1:	ISO Dialect M
$MN_MM_EXTERN_CNC_SYSTEM = 2:	ISO Dialect T

$MN_MM_EXTERN_GCODE_SYSTEM = 0:	G code system B
$MN_MM_EXTERN_GCODE_SYSTEM = 1:	G code system A
$MN_MM_EXTERN_GCODE_SYSTEM = 2:	G code system C

To ensure that the shell cycles work in the correct G code system, you must enter the corresponding system in the GUD variable _ZSFI[39]:

_ZSFI[39]: Setting data for G code system with ISO-T
0 = G code system B (default)
1 = G code system A
2 = G code system B
3 = G code system C

Inch/metric switchover

In ISO Dialect Original there is an MD that defines how inch/metr. switchover is programmed; either with G20/21 or G70/71. This MD is not available for ISO Dialect mode and it cannot be selected by means of $MN_MM_EXTERN_GCODE_SYSTEM.

G20/G21 is active by default. Switchover to G70/71 can be implemented with the machine data 10712: $MN_USER_CODE_CONF_NAME_TAB.

$MN_USER_CODE_CONF_NAME_TAB[0]= G20
$MN_USER_CODE_CONF_NAME_TAB[1]= G70
$MN_USER_CODE_CONF_NAME_TAB[2]= G21
$MN_USER_CODE_CONF_NAME_TAB[3]= G71

Whether G20/21 or G70/71 is used, a 1 will be read into system variable $P_GG[6] for G20/G70 and a 2 will be read in for G21/G71.

4.1.6 Fixed feedrates F0 – F9

F0 to F9 can be used to activate ten different feedrate values preset by means of setting data.
To activate rapid traverse with F0, it is necessary to enter the appropriate velocity in the setting data
42160: $SC_FIXED_FEEDRATE_F1_9[0].

The feedrate values for F0 – F9 are entered in the setting data as real values. The entered values are not evaluated.

The function is activated in machine data
22920: $MC_FIXED_FEEDRATE_F1_F9_ON. If this MD is set to FALSE, F1–F9 will be interpreted as normal feedrate programming, e.g. F2 = 2mm/min, F0 = 0mm/min.
If it is set to TRUE, for F1–F9, the feedrate values are fetched from the setting data 42160: $SC_FIXED_FEEDRATE_F1_F9[ ]. If one of these setting data contains the value 0, when the corresponding address extension is programmed, a feedrate of 0 is effective.

Example

$SC_FIXED_FEEDRATE_F1_F9[0] = 5000
$SC_FIXED_FEEDRATE_F1_F9[1] = 1000
$SC_FIXED_FEEDRATE_F1_F9[2] = 500

```
N10 X10 Y10 Z10 F0 G94      ;Approach position with 5000 mm/min
N20 G01 X150 Y30 F1         ;Feedrate 1000mm/min. active
N30 Z0 F2                   ;Position is approached at 500 mm/min
N40 Z10 F0                  ;Position is approached at 5000 mm/min
```

If the function is activated using machine data $MC_FIXED_FEEDRATE_F1_F9_ON and the feedrate value from the setting data must not become active with F1–F9, the feedrate value must be programmed as a real value. If a feedrate value of, for example, 1 mm/min is to be programmed, the must be programmed with F1.0 instead of F1.

Note

In the case of macro programming with G65/66, for address F, the programmed value is always stored in the cycle system variable. For F1–F9, value 1 – 9 is entered, for example, in the cycle system $C_F. In this case, the address is handled as a transferred parameter and has no direct connection with the feedrate.

The same applies to programming the thread pitch for G33 – G34 with the address F. In this case, a feedrate is not programmed with F, but instead the distance between 2 thread starts with one spindle revolution.

In the case of cycle programming (e.g. G81 X.. Y.. Z.. R.. P.. Q.. F..), the feed-rate is always programmed under address F. In a parts program block with a cycle call via a G function (G81–G87 etc.) therefore when F1 – F9 is programmed, the appropriate feedrate value from the corresponding setting data is written in the variable $C_F.

Restriction

In ISO Dialect mode, the feedrate values in the setting data are changed using a handwheel. In Siemens mode, the feedrates can only be influenced in the same manner as a directly programmed feedrate, e.g. using the override switch.

4.1.7 Parallel axes G17<axis name>.. (G18 / G19)

Function G17 (G18, G19)<axis name>.. can be used to activate an axis that is parallel to the basic axis in the coordinate system.

The basic axes are, for example, X, Y and Z. With the programming

```
G17 U0 Y0      ;Activation of the parallel axis U
```

instead of the basic axis X, axis U is active in the G17 plane.

This function can be simulated with the 840D function GEOAX (..,..). GEOAX() can be used to replace a geometry axis with any channel axis. In this case, all frames (with the exception of handwheel offset and external offset), working area limitation and the protection zones are deleted. Deletion of the frames can be prevented with machine data 10602: $MN_FRAME_GEOAX_CHANGE_MODE; deletion of the protection zones can be prevented with machine data $MC_PROTAREA_GEOAX_CHANGE_MODE; and deactivation of working area limitation can be prevented with the new machine data 10604: $MN_WALIM_GEOAX_CHANGE_MODE.

For each of the 3 geometry axes, one parallel axis can be defined with machine data 22930: $MC_EXTERN_PARALLEL_GEOAX[ ]. When the G function for plane selection (G17–G19) is programmed as well as the axis name of the parallel axis, geometry axis replacement will be performed in a similar manner to the function GEOAX().

On plane selection, the axes (geometry axis replaced with parallel axis) will be traversed to their programmed position.

In a plane selection record, if a basic axis of the coordinate system is programmed in combination with its parallel axis, alarm 12726 “Illegal plane selection with parallel axes” is output.

4.1.8 Insertion of chamfers and radii

The insertion of chamfers and radii is mapped onto the corresponding Siemens functionality. The two blocks between which a radius or a chamfer is to be inserted must always be programmed. If more than one address is programmed in a block, the radius programmed last is always effective.

In ISO Dialect mode, the name of the radius is always “R” and for the chamfer it is always “C”. As soon as a comma is programmed into a block, addresses R and C to the right of the comma are always interpreted as the radius and chamfer respectively. If a circle with radius R or the axis C is to be programmed, the two addresses must be to the left of the comma. A radius or a chamfer can be inserted between linear blocks, circular blocks or a mixture of the two.

In ISO Dialect T mode, the name for the radius is always “R”, but addresses “C”, “I” and “K” can be used for the chamfer. Address C is only permitted to be used when it is not defined as an axis name. Radii and chamfers can only be inserted between linear blocks. The linear blocks must not be vertically one above the other. A sign programmed in front of the radius or chamfer is of no significance here because the direction of the radius or chamfer is defined by the second linear block.

Programming in ISO Dialect M

Chamfers and radii are always marked in the block with a comma. The address for a chamfer is “C” and for a radius “R”. Radii and chamfers can be inserted between linear blocks and circular blocks.

```
N10 X100. ,R10                ;Insert radius of 10mm
N20 Y30. ,C5                  ;Insert chamfer of 5mm
N30 X150. Y40.
N40 G03 X180. Y65. R30 ,R8
N50 G01 X150. ,R8
```

Programming in ISO Dialect T

Chamfers and radii are not marked in the block with a comma. The address for a radius is always “R”, but the address for a chamfer can be “I”, “K” or “C”. C is only permitted to be used when the address is not defined as an axis name.
In ISO Dialect T mode, chamfers and radii can only be inserted between 2 linear blocks.

```
N10 X100. R10                 ;Insert radius of 10mm
N20 Z30. C5                   ;Insert chamfer of 5mm
N30 X150. Z40.
N40 X180. Z65. I8
N50 G01 X150. K8
```

4.1.9 Rotary axis function

If an axis is defined as a rotary axis, this axis can be traversed as follows:

- The axis is traversed as if it were a linear axis
  No modulo conversion is active
  Positive and negative values can be programmed
  30300: $MA_IS_ROT_AX = TRUE
  30310: $MA_ROT_IS_MODULO = FALSE

- Example:
  ```
  N5 G90 B0
  N10 B370       ; Axis travels 370 degrees
  N15 B–10       ; Axis travels –10 degrees
  N20 G91 B–20   ; Axis travels –20 degrees
  ```

- The axis traverses the shortest path
  Modulo conversion is active
  Values <0 and > modulo value can be programmed
  30300: $MA_IS_ROT_AX = TRUE
  30310: $MA_ROT_IS_MODULO = TRUE
  20754: $MC_EXTERN_FUNCTION_MASK, bit5 = 0

30455: $MA_MISC_FUNCTION_MASK, bit0 = 1

Example:	N5 B0	; Axis travels 0 degrees
	N10 B10	; Axis travels 10 degrees, positive ; direction of rotation
	N15 B350	; Axis travels 350 degrees, negative ; direction of rotation
	N20 B–5	; Axis travels 355 degrees, positive ; direction of rotation
	N25 G91 B–10	; Axis travels 345 degrees, negative ; direction of rotation

- The axis traverses in a positive or negative direction of travel depending on the programmed sign
  Modulo conversion is active
  Values <0 and > modulo value can be programmed
  30300: $MA_IS_ROT_AX = TRUE
  30310: $MA_ROT_IS_MODULO = TRUE
  20754: $MC_EXTERN_FUNCTION_MASK, bit5 = 1
  30455: $MA_MISC_FUNCTION_MASK, bit0 = 1

Example:	N5 B0	; Axis travels 0 degrees
	N10 B10	; Axis travels 10 degrees, positive ; direction of rotation
	N15 B350	; Axis travels 350 degrees, positive ; direction of rotation
	N20 B–5	; Axis travels 355 degrees, negative ; direction of rotation
	N25 G91 B–10	; Axis travels 345 degrees, negative ; direction of rotation

In this case the sign has two meanings: it is taken into account in the modulo conversion and defines the direction of rotation.
The response is the same for all interpolation types.
Incremental movements are always executed in accordance with the plus/minus sign.

4.1.10 Program coordination between two channels and M functions

In order to synchronize the program run between two channels, it is possible to program M functions which act as WAIT marks. If the parts program on one channel reaches such an M function, the program run is halted until the other channel has reached the same M function. Processing then resumes in the parts programs in both channels.

The M function numbers for the wait marks are defined with the aid of two machine data. In the process a range of M numbers is defined exclusively for use with this function.

Machine data 10800: $MN_EXTERN_CHAN_SYNC_M_NO_MIN indicates the smallest M number and 10802: $MN_EXTERN_CHAN_SYNC_M_NO_MAX indicates the highest M number in the range reserved for program coordination. Machine data $MN_EXTERN_CHAN_SYNC_M_NO_MAX must not be greater than ($MN_EXTERN_CHAN_SYNC_M_NO_MIN + 10 * number of channels).

In order to avoid conflicts with standard M functions, the smallest M number (MD $MN_EXTERN_CHAN_SYNC_M_NO_MIN) must not be lower than 100. The default machine data –1 means that program coordination in ISO Dialect T/M mode is not possible.

If a value <100 is entered in $MN_EXTERN_CHAN_SYNC_M_NO_MIN or a value <$MN_ETERN_CHAN_SYNC_M_NO_MIN is entered in $MN_EXTERN_CHAN_SYNC_M_NO_MAX, this produces the alarm “Invalid number for channel synchronization”.

The function is mapped onto the WAITM command in Siemens language (WAITM(<Mark>, <Channel number>, <Channel number>)). In this case channel synchronization for channels 1 and 2 is always carried out in ISO Dialect T/M mode. All other channels can only be synchronized in Siemens mode.

The M numbers are not output to the PLC.

The M functions for channel synchronization must be in a block of their own. If further addresses other than “M” are programmed in the block, alarm 12080 (syntax error) is output.

For a more detailed description see

/PGA/ SINUMERIK 840D/810D/FM–NC Programming Guide, Advanced, Chapter “Program Coordination”.

Restriction

Only 10 M functions (WAIT marks) can be set per channel. The difference between $MN_EXTERN_CHAN_SYNC_M_NO_MAX and $MN_EXTERN_CHAN_SYNC_M_NO_MIN must therefore not exceed 20 in a 2-channel system. In the case of ISO Dialect Original, 99999899 M numbers can be defined as wait marks.

4.2 Default assignment of machine data for ISO Dialect

ISO Dialect M

Default settings of MD $MC_EXTERN_GCODE_RESET_VALUES[ ]:

$MC_EXTERN_GCODE_RESET_VALUES[0]=1	G00
$MC_EXTERN_GCODE_RESET_VALUES[1]=1	G17
$MC_EXTERN_GCODE_RESET_VALUES[2]=1	G90
$MC_EXTERN_GCODE_RESET_VALUES[3]=2	G23
$MC_EXTERN_GCODE_RESET_VALUES[4]=1	G94
$MC_EXTERN_GCODE_RESET_VALUES[5]=1	G20
$MC_EXTERN_GCODE_RESET_VALUES[6]=1	G40
$MC_EXTERN_GCODE_RESET_VALUES[7]=3	G49
$MC_EXTERN_GCODE_RESET_VALUES[8]=4	G80
$MC_EXTERN_GCODE_RESET_VALUES[9]=1	G98
$MC_EXTERN_GCODE_RESET_VALUES[10]=1	G50
$MC_EXTERN_GCODE_RESET_VALUES[11]=2	G67
$MC_EXTERN_GCODE_RESET_VALUES[12]=2	G97
$MC_EXTERN_GCODE_RESET_VALUES[13]=1	G54
$MC_EXTERN_GCODE_RESET_VALUES[14]=3	G64
$MC_EXTERN_GCODE_RESET_VALUES[15]=2	G69
$MC_EXTERN_GCODE_RESET_VALUES[16]=1	G15
$MC_EXTERN_GCODE_RESET_VALUES[17]=0	Modal
$MC_EXTERN_GCODE_RESET_VALUES[21]=1	G50.1
$MC_EXTERN_GCODE_RESET_VALUES[24]=1	G12.1
$MC_EXTERN_GCODE_RESET_VALUES[30]=1	G290

ISO Dialect T

Default settings of MD $MC_EXTERN_GCODE_RESET_VALUES[ ]:

Several optional G code systems are available for ISO Dialect T. The same function is called up via different G commands. G code system B is implemented by default. Machine data 10882: §MN_NC_USER_EXTERN_GCODES_TAB is introduced to use another G code system (see 4.2.1)

$MC_EXTERN_GCODE_RESET_VALUES[0]=1	G00
$MC_EXTERN_GCODE_RESET_VALUES[1]=2	G97
$MC_EXTERN_GCODE_RESET_VALUES[2]=1	G90
$MC_EXTERN_GCODE_RESET_VALUES[3]=2	G69
$MC_EXTERN_GCODE_RESET_VALUES[4]=2	G95
$MC_EXTERN_GCODE_RESET_VALUES[5]=1	G21
$MC_EXTERN_GCODE_RESET_VALUES[6]=1	G40
$MC_EXTERN_GCODE_RESET_VALUES[8]=2	G23
$MC_EXTERN_GCODE_RESET_VALUES[9]=1	G80
$MC_EXTERN_GCODE_RESET_VALUES[10]=1	G98
$MC_EXTERN_GCODE_RESET_VALUES[11]=2	G67
$MC_EXTERN_GCODE_RESET_VALUES[15]=2	G18
$MC_EXTERN_GCODE_RESET_VALUES[17]=0	Modal
$MC_EXTERN_GCODE_RESET_VALUES[19]=1	G50.2
$MC_EXTERN_GCODE_RESET_VALUES[20]=1	G12.1
$MC_EXTERN_GCODE_RESET_VALUES[31]=1	G290

■

Boundary Conditions 5

Availability of the “ISO Dialect” function

The function is an option and available for

- SINUMERIK 810D with CCU1 and CCU2
- SINUMERIK 840D with NCU 572.2 and NCU 573.2

5.1 Restrictions

The following section describes functions for which the SINUMERIK 840D exhibits **non-compatibility** with the ISO Dialect Original when in ISO Dialect mode.

Mode switchover

The standard set of machine data implements “Siemens” mode only. No external NC language is generated as a 2nd G code table.

Machine data 10712: $MN_NC_USER_CODE_CONF_NAME_TAB is only valid for NC language commands in Siemens mode.

For compatibility with Siemens mode (with reference to machine data input, OPI interface: “data array”[0] = 1st G group), the zero G groups defined in ISO Dialect Original are converted as follows in ISO Dialect mode:

ISO Dialect M: G group 0 → G group 18
ISO Dialect T: G group 0 → G group 17

Implicit mode change

Asynchronous subprograms (ASUBs), INI files and macro/GUD definition files always run in Siemens mode. If necessary, an implicit change to Siemens mode is executed. When the file has finished running, the original mode of the external CNC system is restored.

5.1.1 Program commands

F value

ISO Dialect M has fixed F values which can be selected with F1 to F9. These fixed values are not available in ISO Dialect mode. F1 to F9 are interpreted as value F1 to F9.

G02/G03

Programming G02/G03 without a radius parameter results in G01 in ISO Dialect T Original, and in a full circle with an undefined radius in ISO Dialect M Original. In both cases, an NC alarm is output in ISO Dialect mode.

G04 X..

With ISO Dialect Original, the dwell time in the X axis is displayed as the distance to go. This dwell time is not displayed in ISO Dialect mode; the message “dwell time running” is displayed.

G16

Polar coordinate programming is terminated with G15.
In ISO Dialect Original, the polar radius and polar angle are retained. The next time G16 is programmed, an incremental movement can be superimposed on the angle. The result is a traversing movement which cannot be evaluated. The angle and the radius are deleted on a reset or M30.

In ISO Dialect mode, the polar radius and polar angle are deleted with G15. With G16, the traversing movement is always performed with an angle and a radius of 0.

Example: Axis U is parallel to axis X
G17 U Y ; Plane U Y is selected instead of X Y.

In ISO Dialect mode, a parallel axis **cannot** be programmed with G17/G18/G19.

G20/G21

In ISO Dialect Original mode, the zero offsets are not converted on switchover. The decimal point is merely shifted by one decimal place. In ISO Dialect mode, the zero offset values are converted completely.

G22

Protection zone 4 is activated with G22 in ISO Dialect M Original. This is not available in phase 1. Protection zones 1/2 and 3 are implemented. Protection zone 4 is entered permanently in the machine data settings for ISO Dialect. This is not possible with the 840D. In ISO Dialect mode there is only one protection zone.

G40

In ISO Dialect T mode, a vector can be programmed in a linear block with I, J, K; the vector controls the behavior at the end of the block. This function is not possible in ISO Dialect Original mode. An alarm is output if I, J and K are programmed with G40.

G41/G42

The cutter radius compensation functions are not compatible between ISO Dialect Original and 840D.

G53

If G53 (position in machine coordinate system approached) is called while G41/G42 is active, the axes are not traversed separately, but interpolated instead.

G63

With ISO Dialect, G63 can appear in every block. The override is disabled in that block. Override 0 also triggers a stop at the start of a G53 block. This function is practical in combination with rigid tapping in G01 interpolation. In ISO Dialect mode, G63 is only effective in a G01 block. If G63 is selected in a G00 block, this has no effect on the block.

G94/G95

Each time you switch from revolutional feedrate (G95) to linear feedrate (G94) or vice versa, feed F must be programmed again. If the feed is missing, alarm 10860 “No feedrate programmed” is output. ISO Dialect Original mode interprets the feedrate which has been programmed once as either a revolutional or a linear feedrate when the feedtype is changed.

G96

With ISO Dialect M the axis to which G96 refers can be programmed with P. This generates an alarm in ISO mode. Only P0 transverse axis is possible from machine data.

M06

Tool-changing cycle: If an M function is assigned to a cycle via 10715: MD $MN_M_NO_FCT_CYCLE and 10716: $MN_M_NO_FCT_CYCLE_NAME, not all of the parameters programmed in the block can be accessed in the cycle. Only the programmed T number can be read with $C_T/$C_T_PROG. Traversing movements in the M block are executed before the call.

Syntax-governing G functions

With ISO Dialect Original, several syntax-governing G functions can be programmed in the same block. A distinction is made between the following:

syntax governing → non-modal

syntax governing → modal

non syntax governing → modal

non syntax governing → non-modal

In Siemens mode, more than one syntax-governing G function in a block causes an NC alarm.

5.1.2 Tool management

Tool management, tool life and workpiece count monitoring can be reproduced with the Siemens tool management system.

Tool data

Milling: Only tool compensation memory C is supported, i.e. multi-column structure of the tool compensation memory (D == H applies).
The variant whereby the tool and offset number are generated from the same value for milling is not supported.

Read-out/archiving of the current tool data in ISO Dialect mode (G10) is not possible. It is not possible to modify the tool data via G10 until the tool offsets have been set up by an operator action.

Tool length compensation

If tool length compensation input is active in diameter mode, the input for geometry and wear can be configured in diameters with ISO Dialect Original. In ISO Dialect mode, only the wear is specified as a diameter. The geometry is always specified in radius mode.

With ISO Dialect Original, the tool length is calculated as a diameter or radius in the transverse axis, depending on a machine parameter.
In Siemens mode, the length offset is always calculated as a diameter, it is not possible to switch over.

Compensation of the tool length offset by shifting the coordinate system is not possible.

The geometry and wear cannot be taken from different compensation memories for the "Turning" technology.

Modified tool offsets are valid when the next T, H or D value is programmed.

5.1.3 Control system response to Power ON, Reset and block search

Power On

On Power ON, the Siemens G code list is created for all NC channels with the possible code changes from MD 10712:
$MN_NC_USER_CODE_CONF_NAME_TAB.

Startup, Reset

MD 20150: $MC_GCODE_RESET_VALUES[46] defines the startup and reset behavior depending on MD 10880: $MN_EXTERN_CNC_SYSTEM.

This decides between the G codes of ISO Dialect M and ISO Dialect T.

The change between Siemens and ISO Dialect mode has no effect on the modal G functions while the program is running.

Block search

The "search to end of block" block search type available in Siemens mode is the equivalent of the search behavior with ISO Dialect.

In "block search without calculation" mode, the user must provide an appropriate search target, especially in NC programs with a mode switchover (e.g. an NC block with a command from G group 47).

■

Notes

Data Descriptions (MD, SD) 6

6.1 General machine data

10604 MD number	**WALIM_GEOAX_CHANGE_MODE** Working area limitation on changeover from geometry axes				
Default setting: 0		Minimum input limit: 0		Maximum input limit: 1	
Changes effective after Power ON			Protection level: 2/7		Unit: –
Data type: BYTE				Applies with effect from SW version: 6.2	
Meaning:	This machine data is used to specify whether a working area limitation that is active should stay active or be deactivated on geometry axis changeover. The MD is bit-coded as follows: Bit 0= =0: Working area limitation deactivated on geometry axis changeover =1: Active working area limitation remains activated on geometry axis changeover				

10615 MD number	**NCBFRAME_POWERON_MASK** Delete global base frames on Power ON				
Default setting: 0		Minimum input limit: 0		Maximum input limit: 0	
Changes effective after Power ON			Protection level: 2/7		Unit: –
Data type: DWORD				Applies with effect from SW version: 5.2	
Meaning:	This machine data defines whether global base frames are deleted on a Power ON reset. The selection can be made separately for the individual base frames. Bit 0 corresponds to base frame 0, bit 1 to base frame 1, etc. 0: Base frame is retained on Power ON 1: Base frame is deleted on Power ON.				

10652 MD number	**CONTOUR_DEF_ANGLE_NAME** Definable name for angle in the contour short description	
Default setting: “ANG”	Minimum input limit: –	Maximum input limit: –
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: STRING	Applies with effect from SW version: 5	
Meaning:	The setting is effective for Siemens G code programming only, i.e. G290. The name used to program the angle in the contour short description is definable. This allows, for example, identical programming in different language modes: If the angle is named “A”, it is programmed in the same way with Siemens and ISO Dialect. The name must be unique, i.e. axes, variables, macros, etc. must not exist with the same name.	

10654 MD number	**RADIUS_NAME** Definable name for radius non-modally in the contour short description	
Default setting: “RND”	Minimum input limit: –	Maximum input limit: –
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: STRING	Applies with effect from SW version: 5	
Meaning:	The name used to program the radius in the contour short description is definable. This allows, for example, identical programming in different language modes: If the radius is named “R”, it is programmed in the same way with Siemens and ISO Dialect. The name must be unique, i.e. axes, variables, macros, etc. must not exist with the same name. The setting is effective for Siemens G code programming, i.e. G290.	

10656 MD number	**CHAMFER_NAME** Definable name for chamfer in the contour short description	
Default setting: “CHR”	Minimum input limit: –	Maximum input limit: –
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: STRING	Applies with effect from SW version: 5	
Meaning:	The name used to program the chamfer in the contour short description is definable. This allows, for example, identical programming in different language modes: If the chamfer is named “C”, it is programmed in the same way with Siemens and ISO Dialect. The name must be unique, i.e. axes, variables, macros, etc. must not exist with the same name. The setting is effective for Siemens G code programming, i.e. G290. The chamfer in the original direction of movement. Alternatively, the chamfer length can be programmed with the name CHF.	

10704 MD number	DRYRUN_MASK Activating dry run feed	
Default setting:	Minimum input limit: –	Maximum input limit: –
Changes effective after:	Protection level:	Unit: –
Data type: BYTE	Applies with effect from SW version:	
Meaning:	DRYRUN_MASK == 0 Dry run must only be activated or deactivated at the end of a block. DRYRUN_MASK == 1 Dry run feed may be activated or deactivated even during program execution Note: Once dry run feed has been activated, the axes are stopped for the duration of the reorganization. DRYRUN_MASK == 2 Dry run can be activated or deactivated in any phase and the axes are not stopped. **Note:** However, the function is only effective upon using a block which comes "later" in the program run. The function takes effect on the next (implicit) Stop Reset.	

10706 MD number	SLASH_MASK Activating the block skip function	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 2
Changes effective after:	Protection level:	Unit: –
Data type: BYTE	Applies with effect from SW version:	
Meaning:	SLASH_MASK == 0 The block skip function can only be switched over at the end of a block. SLASH_MASK == 1 When SLASH_MASK == 1 the block skip function may be activated even during program execution. Note: Once block skip has been activated, the axes are stopped for the duration of the reorganization. SLASH_MASK == 2 Block switchover is possible in any phase. **Note:** However, the function is only effective upon using a block which comes "later" in the program run. The function takes effect on the next (implicit) Stop Reset.	

10715 MD number	**M_NO_FCT_CYCLE[0]** M function number for cycle call				
Default setting: –1		Minimum input limit: –1		Maximum input limit: –	
Changes effective after Power ON			Protection level: 2/7		Unit: –
Data type: DWORD				Applies with effect from SW version: 5.2	
Meaning:	M number with which a subprogram is called. The name of the subprogram is stored in $MN_M_NO_FCT_CYCLE_NAME. If the M function defined by $MN_M_NO_FCT_CYCLE is programmed in a parts program, the subprogram defined in M_NO_FCT_CYCLE_NAME is started at the end of the block. If the M function is programmed again in the subprogram, the substitution no longer takes place by means of a subprogram call. $MN_M_NO_FCT_CYCLE is effective both in Siemens mode G290 and in external language mode G291. A subprogram call may not be superimposed on M functions with fixed meanings. In the event of a conflict, alarm 4150 is output: – M0 to M5, – M17, M30, – M40 to M45, – M function for spindle/axis mode switchover according to $MC_SPIND_RIGID_TAPPING_M_NR (default M70) – M functions for nibbling/punching according to configuration via $MC_NIBBLE_PUNCH_CODE if activated via $MC_PUNCHNIB_ACTIVATION. – With applied external language ($MN_MM_EXTERN_LANGUAGE) M19, M96–M99. Exception: The M functions defined for the tool change with $MC_TOOL_CHANGE_M_CODE. $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be active in the same block (parts program line), i.e. only one M/T function substitution can be active per block. Neither an M98 call nor a modal subprogram call can be programmed in the block with the M function substitution. A subprogram return jump or end of parts program is not allowed. Alarm 14016 is output in the event of a conflict.				

10716 MD number	**M_NO_FCT_CYCLE_NAME[0]** Name of tool-changing cycle for M functions from MD $MN_MFCT_CYCLE				
Default setting: –		Minimum input limit: –		Maximum input limit: –	
Changes effective after Power ON			Protection level: 2/7		Unit: –
Data type: STRING				Applies with effect from SW version: 5.2	
Meaning:	The name of the cycle is stored in the machine data. This cycle is called when the M function from machine data $MN_M_NO_FCT_CYCLE is programmed. If the M function is programmed in a motion block, the cycle is executed after the movement. $MN_M_NO_FCT_CYCLE is effective both in Siemens mode G290 and in external language mode G291. If a T number is programmed in the calling block, the programmed T number can be scanned in the cycle in variable $P_TOOL. $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be active in the same block, i.e. only one M/T function substitution can be active per block. Neither an M98 call nor a modal subprogram call can be programmed in the block with the T function substitution. A subprogram return jump or end of parts program is not allowed. Alarm 14016 is output in the event of a conflict. You can query the address expansion of the M function during the cycle using the system variable $C_ME. Example: $MN_M_NO_FCT_CYCLE = 6 $MN_M_NO_FCT_CYCLE_NAME = "MSUB" PROC MAIN ... N100 M[2]=6 ;address expansion 2 ... M30 PROC MSUB ... N200 M[$C_ME]=$C_M ... You can query the address expansion of the T number to be programmed during the cycle using the system variable $C_TE. Example: $MN_T_NO_FCT_CYCLE_NAME = "TSUB" PROC MAIN ... N100 T[1]=6 ;address expansion 1 ... M30 PROC TSUB ... N200 T[$C_ME]=$C_M ... System variable $C_T or $C_T_PROG can be used in the cycle to scan the programmed T no. as a decimal value, and $C_TS or $C_TS_PROG as a string (only with tool management). If a T number is programmed with the D number, it can also be scanned in the cycle in system variable $C_D or $C_D_PROG.				

10716 MD number	**M_NO_FCT_CYCLE_NAME[0]** Name of tool-changing cycle for M functions from MD $MN_MFCT_CYCLE
Meaning:	System variable $C_DL_PROG can be used to scan the cycle to ascertain whether address DL (additive offset) has also been programmed with the T function. The programmed value can then be read in via system variable $C_DL. Example: $MN_T_NO_FCT_CYCLE_NAME = “TSUB” PROC MAIN ... ;Tool change with N100 T5 D1 DL=2 ;cutting edge selection and additive offset ... M30 PROC TSUB ... N190 IF $SC_DL_PROG == 1 N200 DL=$SC_DL N210 ENDIF

10717 MD number	T_NO_FCT_CYCLE_NAME Name for tool-changing cycle with T number		
Default setting: –	Minimum input limit: –	Maximum input limit: –	
Changes effective after Power ON	Protection level: 2/7	Unit: –	
Data type: STRING	Applies with effect from SW version: 5.2		
Meaning:	Cycle name for tool-changing routine when called via T function. If a T function is programmed in a parts program block, the subprogram defined in T_NO_FCT_CYCLE_NAME is called at the end of the block. System variable $C_T / $C_T_PROG can be used in the cycle to scan the programmed T no. as a decimal value, and $C_TS / $C_TS_PROG as a string (only with tool management). If a T number is programmed with the D number, it can be scanned in the cycle in system variable $C_D/$C_D_PROG. System variable $C_T_PROG or $C_D_PROG can be used in the subprogram to check whether the T or D command was programmed. The values can be read out with system variable $C_T or $C_D. If another T command is programmed in the subprogram, no substitution takes place, but the T word is output to the PLC. $MN_T_NO_FCT_CYCLE_NAME and system variables $C_T / $C_TS_PROG are effective both in Siemens mode G290 and in external language mode G291. $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be active in the same block, i.e. only one M/T function substitution can be active per block. Neither an M98 call nor a modal subprogram call can be programmed in the block with the T function substitution. A subprogram return jump or end of parts program is not allowed. Alarm 14016 is output in the event of a conflict. You can query the address expansion of the T number to be programmed during the cycle using the system variable $C_TE. Example: $MN_T_NO_FCT_CYCLE_NAME = “TSUB” PROC MAIN ... N100 T[1]=5 ;address expansion 2 ... M30 PROC TSUB ... N200 M[$C_TE]=$C_T ...		

10718 MD number	M_NO_FCT_CYCLE_PAR M function substitution with parameters	
Default setting: –1	Minimum input limit: –	Maximum input limit: –
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies with effect from SW version: 6.3	
Meaning:	If an M function substitution has been configured with MD 10715: M_NO_FCT_CYCLE[n] / MD 10716: M_NO_FCT_CYCLE_NAME[n], a parameter transfer for each system variable as for the T function substitution can be specified for one of these M functions with MD 10718: M_NO_FCT_CYCLE_PAR. The parameters stored in the system variables always refer to the parts program line in which the M function to be substituted was programmed. The following system variables are available: $C_ME : Address expansion of the substituted M function $C_T_PROG : TRUE if address T was programmed $C_T : Value of address T (integer) $C_TE : Address expansion of address T $C_TS_PROG : TRUE if address TS was programmed $C_TS : Value of address TS (string, with tool management only) $C_D_PROG : TRUE if address D was programmed $C_D : Value of address D $C_DL_PROG : TRUE if address DL was programmed $C_DL : Value of address DL	

10719 MD number	T_NO_FCT_CYCLE_MODE Parameterization of T function substitution	
Default setting: 0	Minimum input limit: –	Maximum input limit: –
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies with effect from SW version: 6.4	
Meaning:	This machine data is used to set whether D or DL is transferred as a parameter to the T substitution cycle when D or DL and T are programmed in a single block (default) or whether it is to be executed before the T substitution cycle is called. Value 0: as previously, the D or DL number is transferred to the cycle (default value) Value 1: the D or DL number is calculated directly in the block This function is only active if tool change has been configured with the M function (MD 22550: TOOL_CHANGE_MODE = 1), otherwise the D or DL values are always transferred.	

10760	**G53_TOOLCORR**			
MD number	Method of operation of G53, G153 and SUPA			
Default setting: 2		Minimum input limit: 2		Maximum input limit: 4
Changes effective after Power ON			Protection level: 2/7	Unit: –
Data type: BYTE			Applies with effect from SW version: 5.2	
Meaning:	The MD is effective in both Siemens mode and in external language mode. This machine data defines whether tool length compensation and tool radius compensation are suppressed with language commands G53, G153 and SUPA. 0 = G53/G153/SUPA is non-modal suppression of zero offsets, tool length compensation and tool radius compensation remain active. 1 = G53/G153/SUPA is non-modal suppression of zero offsets, and active tool length and tool radius compensation.			

10800	**EXTERN_CHAN_SYNC_M_NO_MIN**			
MD number	First M number for channel synchronization			
Default setting: –1		Minimum input limit: 100		Maximum input limit:
Changes effective after Power ON			Protection level: 2/7	Unit: –
Data type: DWORD			Applies with effect from SW version: 6.2	
Meaning:	Lowest M number of the M number range that is reserved for channel synchronization.			

10802	**EXTERN_CHAN_SYNC_M_NO_MAX**			
MD number	Last M number for channel synchronization			
Default setting: –1		Minimum input limit: 100		Maximum input limit:
Changes effective after Power ON			Protection level: 2/7	Unit: –
Data type: DWORD			Applies with effect from SW version: 6.2	
Meaning:	Highest M number of the M number range that is reserved for channel synchronization. The M number range is permitted to be up to 10 times the channel number (2 channels = 20 M numbers). If a wider range is specified, alarm 4170 is output.			

10804	**EXTERN_M_NO_SET_INT**			
MD number	M function for activating asynchronous subprogram			
Default setting: 96		Minimum input limit: 0		Maximum input limit:
Changes effective after Power ON			Protection level: 2/7	Unit: –
Data type: DWORD			Applies with effect from SW version: 6.2	
Meaning:	M function number that is used to activate an interrupt program in ISO_T/M mode (asynchronous subprogram, ASUB).			

10806	**EXTERN_M_NO_DISABLE_INT**			
MD number	M function for deactivating asynchronous subprogram			
Default setting: 97	Minimum input limit: 0		Maximum input limit:	
Changes effective after Power ON		Protection level: 2/7		Unit: –
Data type: DWORD			Applies with effect from SW version: 6.2	
Meaning:	M function number that is used to deactivate an interrupt program in ISO_T/M mode (asynchronous subprogram, ASUB).			

10808	**EXTERN_INTERRUPT_BITS_M96**			
MD number	Interrupt program processing (M96)			
Default setting: 0	Minimum input limit: 0		Maximum input limit: 8	
Changes effective after Power ON		Protection level: 2/7		Unit: –
Data type: WORD			Applies with effect from SW version: 6.2	
Meaning:	When the various bits are set, the execution of the interrupt routine activated with M96 P.. can be influenced. Bit 0: =0, interrupt program not possible, M96/M97 are normal M functions =1, activation of interrupt program with M96/M97 is permitted Bit 1: =0, execution of parts program continues from the final position of the block following the interrupted block =1, execution of the parts program continues from the point of interruption Bit 2: =0, the interrupt signal interrupts the current block immediately and starts the interrupt routine =1, the interrupt routine is not started until the block has been completed Bit 3: =0, the machining cycle is interrupted on an interrupt signal =1, the interrupt program is not started until the machining cycle has been completed			

10810	**EXTERN_MEAS_G31_P_SIGNAL**			
MD number	Assignment of measuring inputs for G31 P..			
Default setting: 1	Minimum input limit: 0		Maximum input limit: 3	
Changes effective after Power ON		Protection level: 2/7		Unit: –
Data type: BYTE			Applies with effect from SW version: 6.2	
Meaning:	This machine data is used to assign measuring inputs 1 and 2 to the P numbers programmed with G31 P1 (–P4). The MD is bit-coded. Only bit 0 and bit 1 are evaluated. If, for example, bit 0=1 in $MN_EXTERN_MEAS_G31_P_SIGNAL[1], the first measuring input will be activated with G31 P2. If $MN_EXTERN_MEAS_G31_P_SIGNAL[3] = 2, the second measuring input will be activated with G31 P4. Bit 0: =0: Measuring input 1 is not evaluated for G31 P1 (–P4) =1 Measuring input is activated for G31 P1 (–P4) Bit 1: =0 Measuring input 2 is not evaluated for G31 P1 (–P4) =1 Measuring input 2 is activated for G31 P1 (–P4)			

<table>
<tr><td colspan="2">10812
MD number</td><td colspan="4">EXTERN_DOUBLE_TURRET_ON
Double turret with G68</td></tr>
<tr><td colspan="2">Default setting:</td><td colspan="2">Minimum input limit:</td><td colspan="2">Maximum input limit:</td></tr>
<tr><td colspan="3">Changes effective after:</td><td colspan="2">Protection level:</td><td>Unit: –</td></tr>
<tr><td colspan="3">Data type: BOOLEAN</td><td colspan="3">Applies with effect from SW version: 6.2</td></tr>
<tr><td>Meaning:</td><td colspan="5">The machine data is only effective with $MN_EXTER_CNC_SYSTEM = 2.
This MD defines whether G68 should activate double slide machining (channel synchronization for the first and second channel) or whether the second tool of a double revolver should be activated (= 2, with the spacing defined in setting data $SC_EXTERN_DOUBLE_TURRET_DIST, fully interconnected tool).
FALSE: Channel synchronization for double slide machining
TRUE: Load the second tool of a double turret (=$SC_EXTERN_DOUBLE_TURRET_DISTANCE as an additive zero offset and activate mirroring about the Z axis)</td></tr>
</table>

10814 MD number	EXTERN_M_NO_MAC_CYCLE Macro call with M function		
Default setting:	Minimum input limit:	Maximum input limit:	
Changes effective after Power ON	Protection level: 2/7	Unit: –	
Data type: DWORD	Applies with effect from SW version:		
Meaning:	M number with which a macro is called. The name of the subprogram is specified in $MN_EXTERN_M_NO_MAC_CYCLE_NAME[n]. If the M function defined with $MN_EXTERN_M_NO_MAC_CYCLE[n] is programmed in a parts program block, the subprogram defined in EXTERN_M_NO_MAC_CYCLE_NAME[n] is started, all the addresses programmed in the block are written to the associated variables. If the M function is programmed again in the subprogram, the substitution no longer takes place by means of a subprogram call. $MN_EXTERN_M_NO_MAC_CYCLE_NAME[n] is only effective in external language mode G291. A subprogram call may not be superimposed on M functions with fixed meanings. In the event of a conflict, alarm 4150 is output: – M0 to M5, – M17, M30, – M19, – M40 to M45, – M function for spindle mode/axis mode switchover according to $MC_SPIND_RIGID_TAPPING_M_NR (default: M70), – M functions for nibbling/punching according to configuration with $MC_NIBBLE_PUNCH_CODE if activated with $MC_PUNCHNIB_ACTIVATION. – with external language ($MN_MM_EXTERN_LANGUAGE) applied additionally M96 to M99 – M functions defined with $MN_M_NO_FCT_CYCLE. <u>Exception:</u> The M function defined with $MC_TOOL_CHANGE_M_CODE for tool change. The subprograms activated with $MN_EXTERN_M_NO_MAC_CYCLE_NAME[n] may not be active in the same block (parts program line), i.e. only one M function substitution can be active per block. Neither an M98 call nor a modal subprogram call can be programmed in the block with the T function substitution. A subprogram return jump or end of parts program is not allowed. Alarm 14016 is output in the event of a conflict.		

10815 MD number	EXTERN_M_NO_MAC_CYCLE_NAME Subprogram name for M function macro call		
Default setting:	Minimum input limit:	Maximum input limit:	
Changes effective after Power ON	Protection level:	Unit: –	
Data type: STRING	Applies with effect from SW version:		
Meaning:	Cycle name if called with M function defined with $MN_EXTERN_M_NO_MAC_CYCLE[n].		

10816 MD number	**EXTERN_G_NO_MAC_CYCLE** Macro call with G function	
Default setting:	Minimum input limit:	Maximum input limit:
Changes effective after Power ON	Protection level:	Unit: –
Data type: DOUBLE	Applies with effect from SW version: 6.3	
Meaning:	G number with which a macro is called. The name of the subprogram is specified in $MN_EXTERN_G_NO_MAC_CYCLE_NAME[n]. If the G function defined with $MN_EXTERN_G_NO_MAC_CYCLE[n] is programmed in a parts program block, the subprogram defined in EXTERN_M_NO_MAC_CYCLE_NAME[n] is started, all the addresses programmed in the block are written to the associated $C_xx variables. If a subprogram call is already active via a M/G macro or an M substitution, no subprogram call will be executed. If a standard G function is programmed in this case, it will be executed, otherwise alarm 12470 is output. $MN_EXTERN_G_NO_MAC_CYCLE[n] is only effective in external language mode G291. A block can only contain one subprogram call, i.e. only one M/G function substitution may be programmed in a block and the block must not contain any additional subprogram (M98) or cycle call. A subprogram return jump or end of parts program is not allowed in the same block. Alarm 14016 is output in the event of a conflict.	

10817 MD number	**EXTERN_G_NO_MAC_CYCLE_NAME** Subprogram name for G function macro call	
Default setting:	Minimum input limit:	Maximum input limit:
Changes effective after Power ON	Protection level:	Unit: –
Data type: STRING	Applies with effect from SW version: 6.3	
Meaning:	Cycle name if called with G function defined with $MN_EXTERN_G_NO_MAC_CYCLE[n].	

10818 MD number	**EXTERN_INTERRUPT_NUM_ASUP** Interrupt number for ASUP start (M96)	
Default setting:	Minimum input limit:	Maximum input limit:
Changes effective after Power ON	Protection level:	Unit: –
Data type: BYTE	Applies with effect from SW version: 6	
Meaning:	Number of the interrupt input with which an asynchronous subprogram activated in ISO mode is started. (M96< program number>)	

10820 MD number	**EXTERN_INTERRUPT_NUM_RETRAC** Interrupt number for rapid retraction (G10.6)	
Default setting:	Minimum input limit:	Maximum input limit:
Changes effective after Power ON	Protection level:	Unit: –
Data type: BYTE	Applies with effect from SW version: 6	
Meaning:	Number of the interrupt input with which a rapid retraction to the position programmed with G10.6 is triggered in ISO mode.	

10880	**MM_EXTERN_CNC_SYSTEM**	
MD number	External control system whose programs are executed	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 2
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: WORD	Applies with effect from SW version: 5	
Meaning:	Selection of the external language 1 = ISO–2: System Fanuc0 Millling (5.1 and higher) 2 = ISO–3: System Fanuc0 Turning (5.2 and higher) The functional scope defined in the current Siemens documentation is valid. This data is only evaluated if machine data $MN_MM_EXTERN_LANGUAGE is set.	

10881	**MM_EXTERN_GCODE_SYSTEM**	
MD number	ISO mode T: G code system	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 2
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies with effect from SW version: 6.2	
Meaning:	Definition of the G code system that is to be actively executed in ISO Dialect T mode: Value = 0: ISO_T: Code system B Value = 1: ISO_T: Code system A Value = 2: ISO_T: Code system C To make sure that the shell cycles are executed in the correct G code system, the relevant system must be entered in the GUD variable _ZSFI[39].	

10882	**NC_USER_EXTERN_GCODES_TAB [n]:0...59**	
MD number	List of user-specific G commands of an external NC language	
Default setting: –	Minimum input limit: –	Maximum input limit: –
Changes effective after Power ON	Protection level: 2/2	Unit: –
Data type: STRING	Applies with effect from SW version: 5	
Meaning:	Code B is implemented by default for external programming language ISO Dialect T. Code A and Code C have different G function names. $MN_NC_USER_EXTERN_GCODES_TAB can be used to rename the G functions. The G command codes can be changed for external NC languages. The G group and the position within the G group remain the same. Only the G command codes can be changed. Up to 30 code changes are possible. Example: $MN_NC_USER_EXTERN_GCODES_TAB[0]="G20" $MN_NC_USER_EXTERN_GCODES_TAB[1]="G70" —> G20 is reassigned to G70; If G70 already exists, an error message appears on NCK reset.	

10884	**EXTERN_FLOATINGPOINT_PROG**		
MD number	Evaluation of programmed values without a decimal point		
Default setting: 1	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after Power ON	Protection level: 2/7	Unit: –	
Data type: BOOLEAN	Applies with effect from SW version: 5.2		
Meaning:	This machine data is effective for external programming languages, i.e. if MD 18800: MM_EXTERN_LANGUAGE = 1. The machine data defines how programmed values without decimal points are evaluated. 0: Standard notation: Values without decimal points are interpreted in internal units IS-B, IS–C (see MD EXTERN_INCREMENT_SYSTEM). Values without decimal points are interpreted in internal units e.g. X1000 = 1mm (for 0.001mm input resolution) X1000.0 = 1000mm 1: Pocket calculator notation: Values without decimal points are interpreted as mm, inch or degrees. Values without decimal points are interpreted as mm, inch or degrees e.g. X1000 = 1000mm X1000.0 = 1000mm		

10886	**EXTERN_INCREMENT_SYSTEM**		
MD number	Increment system		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after Power ON	Protection level: 2/7	Unit: –	
Data type: BOOLEAN	Applies with effect from SW version: 5.2		
Meaning:	This machine data is effective for external programming languages, i.e. if MD 18800: MM_EXTERN_LANGUAGE = 1. This machine data defines which increment system is active 0: Increment system IS-B = 0.001 mm/degrees = 0.0001 inch 1: Increment system IS-C = 0.0001 mm/degrees = 0.00001 inch		

10888	**EXTERN_DIGITS_TOOL_NO**		
MD number	Number of digits for T number in external language mode		
Default setting: 2	Minimum input limit: 2	Maximum input limit: 4	
Changes effective after Power ON	Protection level: 2/7	Unit: –	
Data type: BYTE	Applies with effect from SW version: 5.2		
Meaning:	The machine data is only effective with $MN_EXTERN_CNC_SYSTEM = 2. Number of digits for tool number in programmed T value. The number of leading digits specified in $MN_EXTERN_DIGITS_TOOL_NO is interpreted as the tool number from the programmed T value. The trailing digits address the compensation memory.		

10890	**EXTERN_TOOLPROG_MODE**			
MD number	Tool change programming with external programming language			
Default setting: 0	Minimum input limit: 0		Maximum input limit: 1	
Changes effective after Power ON		Protection level: 2/7		Unit: –
Data type: BYTE			Applies with effect from SW version: 5.2	
Meaning:	Configuration of tool change programming for external programming language: Bit0 = 0: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 2: The tool number and offset number are programmed in the T value. $MN_DIGITS_TOOLNO determines the number of leading digits representing the tool number. Example: $MN_DIGITS_TOOL_NO = 2 T = 1234 ; tool number 12, ; offset number 34 Bit0 = 1: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 2: Only the tool number is programmed in the T value. Offset number = tool number. $MN_DIGITS_TOOL_NO is irrelevant. Example: T = 12 ; tool number 12 ; offset number 12 Bit1 = 0: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 2: If the number of digits programmed in the T value is equal to the number in $MN_EXTERN_DIGITS_TOOL_NO, leading zeroes are added. Bit1 = 1: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 2: If the number of digits programmed in the T value is equal to the number of digits specified in $MN_EXTERN_DIGITS_TOOL_NO, the programmed number is used as the offset number and the tool number Bit2 = 0: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 2: ISO T offset selection only with D (Siemens cutting edge number) Bit2 = 1: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 2: ISO T offset selection only with H ($TC_DPH[t,d]) Bit3 = 0: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 2: Each H number is allowed only once per TOA, except for H=0. If Bit3 1 –> 0 is set, no H number is not allowed to occur several times in a TO unit. Otherwise an alarm will be output on the next restart. Bit3 = 1: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 2: Each H number is allowed several times per TOA. Bit6 = 0: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 1: Selection of tool length not possible under the address H. Bit6 = 1: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 1: Selection of tool length under address H. Bit7 = 0: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 1: Selection of tool length not possible under address D. Bit7 = 1: Effective only with $MN_MM_EXTERN_CNC_LANGUAGE = 1: Selection of tool length under address D. If Bit6 and Bit7 is set, the selection is possible under address D or H.				

18800 MD number	**MM_EXTERN_LANGUAGE** External language active in the PLC	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 1
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies with effect from SW version: 5	
Meaning:	To execute parts programs generated by controller manufacturers other than Siemens, the appropriate NC language must be activated. Only one external language can be selected. The available instruction set will be described in the applicable documentation. Bit 0 (LSB): Execution of parts programs ISO_2 or ISO_3. For codes, see $MN_MM_EXTERN_CNC_SYSTEM (10880)	

6.2 Channel-specific machine data

20094 MD number	**SPIND_RIGID_TAPPING_M_NR** M number for switchover to controlled spindle mode (Siemens mode)	
Default setting: 70	Minimum input limit: 0	Maximum input limit: 0xFF
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: BYTE	Applies with effect from SW version: 5.2	
Meaning:	The machine data is effective in Siemens mode and external language mode. This machine data defines the M function number used to switch the spindle to controlled spindle mode (axis mode). This number replaces M70 in Siemens mode and M29 in external language mode. Only M numbers which have not already been defined as defaults are allowed. M<NBS>numbers M1, M2, M3, M4, M5 M30 etc. are not allowed, for example.	

20095 MD number	**EXTERN_RIGID_TAPPING_M_NR** M number for switchover to controlled spindle mode (external language mode)	
Default setting: 29	Minimum input limit: 6	Maximum input limit: 0xFF
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: BYTE	Applies with effect from SW version:	
Meaning:	This machine data defines the M function number used to switch the spindle to controlled spindle mode (axis mode) in external language mode. This number can be used in external language mode to substitute M29 with another M function. Only M numbers which have not already been defined as defaults are allowed. M numbers M0, M1, M3, M4, M5, M30, M99 etc. are not allowed, for example.	

20152 MD number	GCODE_RESET_MODE Reset response in the G groups	
Default setting:	Minimum input limit: 0	Maximum input limit: 1
Changes effective after Reset	Protection level: 2/7	Unit: –
Data type: BYTE	Applies with effect from SW version:	
Meaning:	This machine data is only evaluated if bit 0 is set in $MC_RESET_MODE_MASK. For every entry in machine data $MN_GCODE_RESET_VALUES (and thus for every G group) this MD defines whether the setting corresponding to $MC_GCODE_RESET_VALUES will be resumed upon the occurrence of a reset/parts program end (MD = 0), or if the setting valid at that moment will be retained (MD = 1). Example: In this case whenever there is a reset/parts program end, the initial setting for the sixth G group (current plane) will be read from machine data $MC_GCODE_RESET_VALUES: $MC_GCODE_RESET_VALUE(5)=1; Reset value of sixth G group is M17 $MC_GCODE_RESET_MODE(5)=0; Initial setting for sixth G group after a reset/parts program end is as in $MC_GCODE_RESET_VALUES(5) If it is required that the current setting for the sixth G group (current plane) be retained in the event of a reset/parts program end, the setting is as follows: $MC_GCODE_RESET_VALUE(5)=1; Reset value of sixth G group is M17 $MC_GCODE_RESET_MODE(5)=1; Current setting for the sixth G group is retained even after a reset/part program end	

20154 MD number	EXTERN_GCODE_RESET_VALUES[n]: 0, ..., 30 Defines the G codes which are activated on startup if the NC channel is not running in Siemens mode.	
Default setting: –	Minimum input limit: –	Maximum input limit: –
Changes effective after Power ON	Protection level: 2/2	Unit: –
Data type: BYTE	Applies with effect from SW version: 5	
Meaning:	The following external programming languages are possible: – ISO Dialect Milling – ISO Dialect Turning The G group classification to be used is specified in the current SINUMERIK documentation. The following groups can be defined within MD EXTERN_GCODE_RESET_VALUES: **ISO Dialect M**: G group 2: G17/G18/G19 G group 3: G90/G91 G group 5: G94/G95 G group 6: G20/G21 G group 13: G96/G97 G group 14: G54–G59 **ISO Dialect T**: G group 2: G96/G97 G group 3: G90/G91 G group 5: G94/G95 G group 6: G20/G21 G group 16: G17/G18/G19	

20380 MD number	**TOOL_CORR_MODE_G43/G44** Handling for tool length offset G43/G44		
Default setting: 0	Minimum input limit: 1	Maximum input limit: 2	
Changes effective after Reset	Protection level: 2/7	Unit: –	
Data type: BYTE		Applies with effect from SW version: 5.2	
Meaning:	The machine data is only effective with $MN_MM_EXTERN_CNC_LANGUAGE = 1; When G43/G44 is active, it determines how length offsets programmed with H are processed. 0: Mode A The tool length H always acts on the Z axis, independent of the current plane 1: Mode B Depending on the active plane, tool length H acts on one of the three geometry axes, namely with G17 on the third geometry axis (usually Z) with G18 on the second geometry axis (usually Y) with G19 on the first geometry axis (usually X) In this mode, offsets can be programmed for all three geometry axes, i.e. the activation of a component in one axis does not cancel a length offset which is already active in another axis. 2: Mode C The tool length is applied independent of the active plane to the axis programmed simultaneously with H. In all other respects the behavior is the same as variant B		

20382 MD number	**TOOL_CORR_MOVE_MODE** Traversing the tool length offset		
Default setting: FALSE	Minimum input limit: –	Maximum input limit: –	
Changes effective after Reset	Protection level: 2/7	Unit: –	
Data type: BOOLEAN		Applies with effect from SW version: 5.2	
Meaning:	The machine data determines how the tool length offsets are applied. FALSE: A tool length component is only applied if the associated axis was programmed. (same behavior as previous software versions) TRUE: Tool lengths are always applied immediately, regardless of whether the associated axes were programmed.		

20732 MD number	**EXTERN_G0_LINEAR_MODE** Interpolation behavior for G00		
Default setting: 1	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after Power ON	Protection level: 2/4	Unit: –	
Data type: BOOLEAN		Applies with effect from SW version:	
Meaning:	This MD defines the interpolation behavior for G00. 0: Axes traverse as positioning axes 1: Axes interpolate with each other		

20734 MD number	**EXTERN_FUNCTION_MASK** Function mask for external language	
Default setting:	Minimum input limit: 0	Maximum input limit: 16
Changes effective after Reset	Protection level: 2/7	Unit: –
Data type: DWORD	Applies with effect from SW version: 6.2	
Meaning:	This machine data is used to influence functions in ISO mode. Bit 0 =0: ISO mode T: "A" and "C" are interpreted as axes. If contour definition is programmed, a comma must precede "A" or "C". =1: "A" and "C" in the parts program are always interpreted as contour definition. Neither axis A nor axis C is permitted to exist. Bit 1 =0: ISO mode T G10 P<100 tool geometry >100 tool wear =1: G10 P<10 000 tool geometry >10 000 tool wear Bit 2 =0: G04 dwell time: always either [s] or [ms] =1: if G95 is active, dwell time in spindle revolutions Bit 3 =0 ISO scanner errors result in an alarm Example: N5 G291 ; ISO Dialect mode N10 WAIT ; Alarm 12080 "WAIT unknown" N15 G91 G500 ; Alarm 12080 "G500 unknown" =1: ISO scanner errors are not output. The block will be transferred to the Siemens translator Example: N5 G291 ; ISO Dialect mode N10 WAIT ; The Siemens translator will process ; the block N15 G91 G500 ; The Siemens translator will process ; the block N20 X Y ; Block processed by ISO translator due to ; G291, G91 off N15 is active Bit 4 =0: G00 is traversed into the exact stop function. Example: In G64, G00 blocks are also traversed with G64 =1 G00 blocks are always traversed with G09, even when G64 is active Bit 5 =0: Movements of the rotary axis are carried out along the shortest path =1: Depending on the sign, movements of the rotary axis are carried out in the positive or negative direction of rotation Bit 6 =0: Only 4-digit program number allowed Bit 6 =1: 8-digit program number allowed. Numbers shorter than 4 digits are expanded to 4 digits . Bit 7 =0: Axis programming with geo axis replacement/parallel axes is compatible in ISO mode =1: Axis programming with geo axis replacement/parallel axes is compatible with Siemens mode in ISO mode Bit8 =0: In cycles, the F value is always interpreted as a feedrate for transfer =1: In thread cycles, the F value is interpreted as a pitch for transfer Bit9 =0: In ISO Mode T for G84, G88 and in standard mode F for G95, multiplication is by 0.01 mm or 0.0001 inch =1: In ISO Mode T for G84, G88 and in standard mode F for G95, multiplication is by 0.01 mm or 0.0001 inch Bit 10 = 0: In M96 Pxx the Pxx program is called when interrupted. = 1: In M96 Pxx CYCLE396.spf is always called when interrupted. Bit 11 = 0: When G54 Pxx is programmed, G54.1 is displayed. = 1: When G54 Pxx or G54.1 Px is programmed, G54Px is always displayed. Bit 12 = 0: When the UP defined by M96 Pxx is called, $P_ISO_STACK is not changed. =1: When the UP defined by M96 Pxx is called, $P_ISO_STACK is incremented.	

<table>
<tr><td>22420
MD number</td><td colspan="5">FGROUP_DEFAULT_AXES[n]: 0, ..., 7
Default value for FGROUP command</td></tr>
<tr><td colspan="2">Default setting: 0</td><td colspan="2">Minimum input limit: 0</td><td colspan="2">Maximum input limit: 8</td></tr>
<tr><td colspan="3">Changes effective after Power ON</td><td colspan="2">Protection level: 7/7</td><td>Unit: –</td></tr>
<tr><td colspan="4">Data type: BYTE</td><td colspan="2">Applies with effect from SW version: 5.2</td></tr>
<tr><td>Meaning:</td><td colspan="5">You can specify up to 8 channel axes whose resulting velocity corresponds to the programmed path feed. If all 8 values are set to zero (default), the geometry axes entered in $MC_AXCONF_GEOAX_ASSIGN_TAB are activated as the default setting for the FGROUP command.

Example: The first 4 axes in the channel are relevant for the path feed:
$MC_FGROUP_DEFAULT_AXES[0] = 1
$MC_FGROUP_DEFAULT_AXES[2] = 2
$MC_FGROUP_DEFAULT_AXES[3] = 3
$MC_FGROUP_DEFAULT_AXES[4] = 4</td></tr>
</table>

<table>
<tr><td>22512
MD number</td><td colspan="5">EXTERN_GCODE_GROUPS_TO_PLC[n]: 0, ..., 7
Specifies the G groups which are output to the NCK/PLC interface when an external NC language is active</td></tr>
<tr><td colspan="2">Default setting: –</td><td colspan="2">Minimum input limit: –</td><td colspan="2">Maximum input limit: –</td></tr>
<tr><td colspan="3">Changes effective after Power ON</td><td colspan="2">Protection level: 2/7</td><td>Unit: –</td></tr>
<tr><td colspan="4">Data type: BYTE</td><td colspan="2">Applies with effect from SW version: 5</td></tr>
<tr><td>Meaning:</td><td colspan="5">The user can select the G groups of an external NC language with channel MD $MC_EXTERN_GCODE_GROUPS_TO_PLC. The active G command is then signaled from the NCK to the PLC for these groups.

Default 0: No output
The NCK/PLC interface is updated on every block change and after a Reset.
A block-synchronous relationship does not always exist between the NC block and the signaled G functions (e.g. if very short blocks are used in continuous-path mode).
The same applies to $MC_GCODE_GROUPS_TO_PLC.</td></tr>
</table>

<table>
<tr><td>22515
MD number</td><td colspan="5">GCODE_GROUPS_TO_PLC_MODE
Behavior of G group to PLC</td></tr>
<tr><td colspan="2">Default setting: –</td><td colspan="2">Minimum input limit: –</td><td colspan="2">Maximum input limit: –</td></tr>
<tr><td colspan="3">Changes effective after Power ON</td><td colspan="2">Protection level: 2/7</td><td>Unit: –</td></tr>
<tr><td colspan="4">Data type: DWORD</td><td colspan="2">Applies with effect from SW version: 6.3</td></tr>
<tr><td>Meaning:</td><td colspan="5">For setting how the G groups are to be interpreted in the PLC as data. The current behavior (bit 0=0) was for the G group to be the array index of a 64 byte field (DBB 208 – DBB 271). That way, up to the 64th G group can be reached.
The new behavior (bit 0=1)) is for the data storage in the PLC to be up to 8 bytes (DBB 208 – DBB 215). With this behavior, the array index of this byte array is identical with the index of the MD $MC_GCODE_GROUPS_TO_PLC[Index] and $MC_EXTERN_GCODE_GROUPS_TO_PLC[Index]. Each index (0–7) must only be entered in one of the two machine data, the other must contain the value 0.
Bit 0(LSB = 0: Behavior as before, the 64 byte array is used for the G code.
Bit 0(LSB = 1: The user sets for which G groups the first 8 bytes will be used</td></tr>
</table>

22900 MD number	**STROKE_CHECK_INSIDE** Direction (internal/external) in which the protection zone acts			
Default setting: 0		Minimum input limit: 0		Maximum input limit: 1
Changes effective after Power ON			Protection level: 2/7	Unit: –
Data type: BYTE			Applies with effect from SW version: 5.2	
Meaning:	This machine data applies in combination with external programming languages. It is effective with $MN_MM_EXTERN_LANGUAGE = 1. It defines whether protection zone 3 is an internal or external protection zone. Meaning: 0: Protection zone 3 is an internal protection zone, i.e. the protection zone must not be overtraveled in the internal direction 1: Protection zone 3 is an external protection zone			

22910 MD number	**WEIGHTING_FACTOR_FOR_SCALE** Input resolution for scaling factor			
Default setting: 0		Minimum input limit: 0		Maximum input limit: 1
Changes effective after Power ON			Protection level: 2/7	Unit: –
Data type: BOOLEAN			Applies with effect from SW version: 5.2	
Meaning:	This machine data applies in combination with external programming languages. It is active with $MN_MM_EXTERN_LANGUAGE = 1. It defines the unit for the scale factor P and the axial scale factors I, J, K Meaning: 0: Scale factor in 0.001 1: Scale factor in 0.00001			

22914 MD number	**AXES_SCALE_ENABLE** Activation for axial scaling factor (G51)			
Default setting: 0		Minimum input limit: 0		Maximum input limit: 1
Changes effective after Power ON			Protection level: 2/7	Unit: –
Data type: BOOLEAN			Applies with effect from SW version: 5.2	
Meaning:	This MD enables axial scaling. Meaning: 0: Axial scaling not possible 1: Axial scaling is possible, i.e. MD DEFAULT_SCALE_FACTOR_AXIS is active			

22920 MD number	EXTERN_FIXED_FEEDRATE_F1_ON Activation of fixed feedrates F1 – F9	
Default setting: FALSE	Minimum input limit:	Maximum input limit:
Changes effective after Power ON	Protection level: 2/7	Unit:
Data type: BOOLEAN	Applies with effect from SW version: 6.2	
Meaning:	This MD enables the fixed feedrates from the setting data $SC_EXTERN_FIXED_FEEDRATE_F1_F9 [ ]. 0: no fixed feedrates with F1 – F9 1: the feedrates from the setting data $SC_EXTERN_FIXED_FEEDRATE_F1_F9 are activated by programming F1 –F9	

22930 MD number	EXTERN_PARALLEL_GEOAX Assignment of parallel channel geometry	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 3
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: BYTE	Applies with effect from SW version: 6.2	
Meaning:	Assignment table for the axes that run parallel to the geometry axes. This table can be used to assign the channel axes that run parallel to the geometry axes. The parallel axes can then be activated in ISO Dialect with the G functions for plane selection (G17–G19) and the axis names for the parallel axis as a geometry axis. An axis replacement is then performed with the axis defined by means of $MC_AXCONF_GEOAX_ASSIGN_TAB[ ]. Prerequisite: The channel axes used must be active (with an occupied list slot in AXCONF_MACHAX_USED). If a zero is entered, the corresponding parallel geometry axis is deactivated.	

24004 MD number	CHBFRAME_POWERON_MASK Reset channel-specific base frame following Power ON	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 0xFF
Changes effective after Power ON	Protection level: 2/7	Unit: –
Data type: DWORD	Applies with effect from SW version: 5.2	
Meaning:	This machine data is used to specify whether channel-specific base frames should be reset in the data management on Power ON Reset. This means that displacements and rotations are set to 0 and scaling will be set to 1. Mirroring will be disabled. The selection can be made separately for the individual base frames. Bit 0 corresponds to base frame 0, bit 1 to base frame 1, etc. 0: Base frame is retained on Power ON 1: Base frame is reset in data management on Power ON.	

6.3 Axis-specific setting data

43120	**DEFAULT_SCALE_FACTOR_AXIS**			
MD number	Axial default scaling factor with active G51			
Default setting: 1		Minimum input limit: –99999999		Maximum input limit: 99999999
Changes effective immediately			Protection level: 7/7	Unit: –
Data type: DWORD			Applies with effect from SW version: 5.2	
Meaning:	This machine data applies in combination with external programming languages. It is effective with $MN_MM_EXTERN_LANGUAGE = 1. If no axial scale factor I, J or K is programmed in the G51 block, DEFAULT_SCALEFACTOR_AXIS is effective. For the scale factor to be effective, the machine data AXES_SCALE_ENABLE must be set.			

43240	**M19_SPOS**			
MD number	Spindle position in degrees for spindle positions with M19			
Default setting: 0		Minimum input limit: –359.999		Maximum input limit: 359.999
Changes effective immediately			Protection level: 7/7	Unit: –
Data type: DOUBLE			Applies with effect from SW version: 5.2	
Meaning:	The setting data is also effective in Siemens mode.			

6.4 Channel-specific setting data

42110	**DEFAULT_FEED**			
MD number	Default value for path feed			
Default setting: 0		Minimum input limit: 0		Maximum input limit: –
Changes effective immediately			Protection level: 7/7	Unit: –
Data type: DOUBLE			Applies with effect from SW version: 5.2	
Meaning:	If no path feed is programmed in the parts program, the value stored in $SC_DEFAULT_FEED is used. The setting data is evaluated at the start of the parts program allowing for the feed type active at the time (see $MC_GCODE_RESET_VALUES and/or $MC_EXTERN_GCODE_RESET_VALUES).			

42140	**DEFAULT_SCALE_FACTOR_P**			
MD number	Default scaling factor for address P			
Default setting: 0		Minimum input limit: –99999999		Maximum input limit: 99999999
Changes effective immediately			Protection level: 7/7	Unit: –

42140 MD number	DEFAULT_SCALE_FACTOR_P Default scaling factor for address P
Data type: DWORD	Applies with effect from SW version: 5.2
Meaning:	This machine data applies in combination with external programming languages. It is effective with $MN_MM_EXTERN_LANGUAGE = 1. If no scale factor P is programmed in the block, the value in this machine data is applied.

42150 MD number	DEFAULT_ROT_FACTOR_R Default selection for rotation angle R	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 360
Changes effective immediately	Protection level: 2/7	Unit: Degrees
Data type: DOUBLE	Applies with effect from SW version:	
Meaning:	If a factor has not been programmed for rotation R during selection of rotation G68, the value from this setting data becomes effective.	

42160 MD number	EXTERN_FIXED_FEEDRATE_F1_F9 Allow fixed feedrates with F1–F9	
Default setting: 0	Minimum input limit: 0	Maximum input limit:
Changes effective immediately	Protection level: 2/7	Unit: VELO
Data type: DOUBLE	Applies with effect from SW version:	
Meaning:	Fixed feedrates that are selected by programming F1–F9 when G01 is active.	

42162 MD number	EXTERN_DOUBLE_TURRET_DIST Tool spacing on the double turret	
Default setting:	Minimum input limit:	Maximum input limit:
Changes effective	Protection level:	Unit:
Data type: DOUBLE	Applies with effect from SW version:	
Meaning:	The machine data is only effective with $MN_EXTER_CNC_SYSTEM = 2. Turning Spacing of both the tools on a double slide turret. The spacing is activated as an additive zero offset when code G68 is used, if $MN_EXTERN_DOUBLE_TURRET_ON = TRUE is set.	

42520 MD number	CORNER_SLOWDOWWN_START Beginning of feedrate reduction with G62	
Default setting: 0	Minimum input limit: 0	Maximum input limit: any
Changes effective immediately	Protection level: 7/7	Unit: POSN_LIN

42520 MD number	**CORNER_SLOWDOWWN_START** Beginning of feedrate reduction with G62			
Data type: DOUBLE			Applies with effect from SW version: 6	
Meaning:	Path length from which the feedrate is reduced before the corner with G62			

42522 MD number	**CORNER_SLOWDOWN_END** End of feedrate reduction with G62			
Default setting: 0		Minimum input limit: 0		Maximum input limit: any
Changes effective immediately		Protection level: 7/7		Unit: POSN_LIN
Data type: DOUBLE			Applies with effect from SW version: 6	
Meaning:	Path length up to which the feedrate will remain reduced after a corner for G62.			

42524 MD number	**CORNER_SLOWDOWN_OVR** Override for feedrate reduction with G62			
Default setting: 0		Minimum input limit: 0		Maximum input limit: any
Changes effective immediately		Protection level: 7/7		Unit: PERCENT
Data type: DOUBLE			Applies with effect from SW version: 6	
Meaning:	Override with which the feedrate is multiplied at the corner for G62.			

42526 MD number	**CORNER_SLOWDOWN_CRIT** Corner recognition in G62, G21			
Default setting: 0		Minimum input limit: 0		Maximum input limit: any
Changes effective immediately		Protection level: 7/7		Unit: POSN_ROT
Data type: DOUBLE			Applies with effect from SW version: 6	
Meaning:	Angle from which a corner is considered to be a corner for the purposes of feedrate reduction with G62, G21.			

43340 MD number	**EXTERN_REF_POSITION_G30_1** Reference point position for G30.1			
Default setting:		Minimum input limit:		Maximum input limit:
Changes effective immediately		Protection level:		Unit:
Data type: DOUBLE			Applies with effect from SW version:	
Meaning:	Setting data Reference point position for G30.1. This setting data is evaluated in CYCLE328.			

■

Signal Descriptions

7

None

8

Example

None

Notes

Data Fields, Lists 9

9.1 Machine data

Number	Identifier	Name	Refer-, ence
General ($MN_ ...)			
10604	WALIM_GEOAX_CHANGE_MODE	Working area limitation on changeover from geometry axes	
10615	NCFRAME_POWERON_MASK	Delete global base frames following Power ON	K2
10652	CONTOUR_DEF_ANGLE_NAME	Definable name for angle in the contour short description	
10654	RADIUS_NAME	Definable name for radius non-modally in the contour short description	
10656	CHAMFER_NAME	Definable name for chamfer in the contour short description	
10704	DRYRUN_MASK	Activating dry run feed	
10706	SLASH_MASK	Activating the block skip function	
10715	M_NO_FCT_CYCLE[n]: 0, ..., 0	M function number for tool-changing cycle call	K1
10716	M_NO_FCT_CYCLE_NAME[ ]	Name of tool-changing cycle for M functions from MD $MN_MFCT_CYCLE	K1
10717	T_NO_FCT_CYCLE_NAME	Name of tool-changing cycle for T function	K1
10718	M_NO_FCT_CYCLE_PAR	M function substitution with parameters	K1
10719	T_NO_FCT_CYCLE_MODE	Parameterization of T function substitution	K1
10740	EXTER_M_NO_MAC_CYCLE	Macro call with M function	
10741	EXTER_M_NO_MAC_CYCLE_NAME	Subprogram name for M function macro call	
10760	G53_TOOLCORR	Method of operation of G53, G153 and SUPA	
10800	EXTERN_CHAN_SYNC_M_NO_MIN	First M number for channel synchronization	
10802	EXTERN_CHAN_SYNC_M_NO_MAX	Last M number for channel synchronization	
10804	EXTERN_M_NO_SET_INT	M function for activating asynchronous subprogram	
10806	EXTERN_M_NO_DISABLE_INT	M function for deactivating asynchronous subprogram	
10808	EXTERN_INTERRUPT_BITS_M96	Interrupt program processing (M96)	
10810	EXTERN_MEAS_G31_P_SIGNAL	Assignment of measuring inputs for G31 P..	
10812	EXTERN_DOUBLE_TURRET_ON	Double turret with G68	
10814	EXTERN_M_NO_MAC_CYCLE	Macro call with M function	
10815	EXTERN_M_NO_MAC_CYCLE_NAME	Subprogram name for M function macro call	

General ($MN_ ...)			
10816	EXTERN_G_NO_MAC_CYCLE	Macro call with G function	
10817	EXTERN_G_NO_MAC_CYCLE_NAME	Subprogram name for G function macro call	
10818	EXTERN_INTERRUPT_NUM_ASUP	Interrupt number for ASUB start (M96)	
10820	EXTERN_INTERRUPT_NUM_RETRAC	Interrupt number for rapid retraction (G10.6)	
10870	EXTERN_CHAN_SYNC_M_NO_MIN	First M number for channel synchronization	
10872	EXTERN_CHAN_SYNC_M_NO_MAX	Last M number for channel synchronization	
10880	EXTERN_CNC_SYSTEM	External control system whose programs are to be executed	
10881	EXTERN_GCODE_SYSTEM	ISO mode T: G code system	
10882	NC_USER_EXTERN_GCODES_TAB[n]: 0–59	List of user-specific G commands of an external NC language	
10884	EXTERN_FLOATINGPOINT_PROG	Evaluation of programmed values without a decimal point	
10886	EXTERN_INCREMENT_SYSTEM	Defines the increment system	
10888	EXTERN_DIGITS_TOOL_NO	Number of digits for T number in external language mode	
10890	EXTERN_TOOLPROG_MODE	Tool change programming with external programming language	
18190	MM_NUM_PROTECT_AREA_NCK	Number of files for machine-based protection zones (SRAM)	S7
18800	MM_EXTERN_LANGUAGE	Activation of external NC languages	
Channel-specific ($MC_ ...)			
20050	AXCONF_GEOAX_ASSIGN_TAB[ ]	Assignment of geometry axis to channel axis	K2
20060	AXCONF_GEOAX_NAME_TAB[ ]	Geometry axis in channel	K2
20070	AXCONF_MACHAX_USED[ ]	Machine axis number applicable in channel	K2
20080	AXCONF_CHANAX_NAME_TAB[ ]	Channel axis name in channel	K2
20094	SPIND_RIGID_TAPPING_M_NR	M function for changeover to controlled axis mode	
20095	EXTERN_RIGID_TAPPING_M_NR	M function number in external language mode for spindle switchover to controlled spindle mode	
20100	DIAMETER_AX_DEF	Geometry axis with transverse axis function	P1
20150	GCODE_RESET_VALUES[n]: 0 to max. number of G codes	Initial setting of the G groups	K1
20152	GCODE_RESET_MODE	Reset response in the G group	
20154	EXTERN_GCODE_RESET_VALUES[n]: 0–30	Initial setting of the G groups	
20380	TOOL_CORR_MODE_G43G44	Handling for tool length offset G43/G44	
20382	TOOL_CORR_MOVE_MODE	Traversing the tool length offset	
20732	EXTERN_G0_LINEAR_MODE	Define interpolation behavior for G00	
20734	EXTERN_FUNCTION_MASK	Function mask for external language	
22420	FGROUP_DEFAULT_AXES[ ]	Default value for FGROUP command	
22512	EXTERN_GCODE_GROUPS_TO_PLC[n]: 0–7	Send G command of an external language to PLC	

Channel-specific ($MC_ ...)			
22515	GCODE_GROUPS_TO_PLC_MODE	Behavior of G group to PLC	
22900	STROKE_CHECK_INSIDE	Direction (internal/external) in which the protection zone acts	
22910	WEIGHTING_FACTOR_FOR_SCALE	Unit of scale factor	
22914	AXES_SCALE_ENABLE	Activation for axial scaling factor (G51)	
22920	EXTERN_FEEDRATE_F1_F9_ACTIV	Allow fixed feedrates with F0–F9	
22930	EXTERN_PARALLEL_GEOAX	Assignment of parallel channel geometry	
24004	CHBFRAME_POWERON_MASK	Reset channel-specific base frame following Power ON	
28080	NUM_USER_FRAMES	Number of zero offsets	
29210	NUM_PROTECT_AREA_ACTIVE	Activate protection zone	
34100	REFP_SET_POS[0]	Reference point value/No meaning in distance-coded system	
35000	SPIND_ASSIGN_TO_MACHAX	Assignment of spindle to machine axis	

9.2 Setting data

Number	Identifier	Name	Reference
Axis-specific			
42150	DEFAULT_ROT_FACTOR_R	Default selection for rotation angle R	
43120	DEFAULT_SCALE_FACTOR_AXIS	Axial default scaling factor with active G51	
43240	M19_SPOS	Position of spindle when programming M19	
42890	M19_SPOSMODE	Positioning mode of spindle when programming M19	
Channel-specific			
42110	DEFAULT_FEED	Default value for path feed	V1
42140	DEFAULT_SCALE_FACTOR_P	Default scaling factor for address P	
42150	DEFAULT_ROT_FACTOR_R	Default selection for rotation angle R	
42160	EXTERN_FIXED_FEEDRATE_F1_F9	Allow fixed feedrates with F1–F9	
42162	EXTERN_DOUBLE_TURRET_DIST	Tool spacing on the double turret	
42520	CORNER_SLOWDOWN_START	Beginning of feedrate reduction with G62	
42522	CORNER_SLOWDOWN_END	End of feedrate reduction with G62	
42524	CORNER_SLOWDOWN_OVR	Override for feedrate reduction with G62	

Channel-specific			
42526	CORNER_SLOWDOWN_CRIT	Corner recognition in G62, G21	
43340	EXTERN_REF_POSITION_G30_1	Reference point position for G30.1	

■

Alarms 10

If error states are detected in cycles, an alarm is generated and cycle execution is interrupted.

The cycles continue to output messages in the dialog line of the control. These messages do not interrupt execution.

Alarms with numbers between 61000 and 62999 are generated in the cycles (see /DA/, Diagnostics Guide and /PGZ/, Cycles, Programming Guide). This number range is subdivided further according to alarm reactions and cancelation criteria.

The table below also describes error messages for the cycles described in Chapter 3.

Table 10-1 Alarm number and alarm description

Alarm no.	Brief description	Source	Explanation/remedy
Alarms – general			
61001	Pitch incorrectly specified	CYCLE376T	Pitch incorrectly specified
61003	No feed programmed in cycle	CYCLE371T, CYCLE374T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	No feed “F” was programmed in the calling block before the cycle call, see standard Siemens cycles
61004	Configuration of geometry axis not correct	CYCLE328	The order of the geometry axes is incorrect, see standard Siemens cycles
61101	Reference plane improperly defined	CYCLE375T, CYCLE81, CYCLE83, CYCLE84, CYCLE87	See standard Siemens cycles
61102	No spindle direction programmed	CYCLE371T, CYCLE374T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Spindle direction M03 or M04 missing, see standard Siemens cycles
61107	First drilling depth incorrectly defined		First drilling depth counter to total drilling depth
61603	Recess shape incorrectly specified	CYCLE374T	Recess depth is set to 0
61607	Starting point incorrectly specified	CYCLE376T	The starting point achieved before the cycle was called does not lie outside the area to be machined
61610	No infeed depth programmed	CYCLE374T	Infeed depth is set to 0

Table 10-1 Alarm number and alarm description

Alarm no.	Brief description	Source	Explanation/remedy
ISO alarms			
61800	External CNC system missing	CYCLE300, CYCLE328, CYCLE330, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Machine data for external language MD18800: $MN_MM_EXTERN_LANGUAG E or option bit 19800 $ON_EXTERN_LANGUAGE is not set
61801	Incorrect G code selected	CYCLE300, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T	A numerical value that is invalid for the specified CNC system has been programmed in the program call CYCLE300<value>; or an incorrect value for the G code system has been entered in cycle setting data memory.
61802	Incorrect axis type	CYCLE328, CYCLE330	The programmed axis is assigned to a spindle
61803	Programmed axis does not exist	CYCLE328, CYCLE330	The programmed axis does not exist in the system. Check MD20050–20080
61804	Programmed position beyond reference point	CYCLE328, CYCLE330	The programmed intermediate position or current position is located behind the reference point
61805	Value programmed in absolute and incremental dimensions	CYCLE328, CYCLE330, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T	The intermediate position is programmed using both absolute and incremental dimensions
61806	Incorrect axis assignment	CYCLE328	The order of the axis assignment is incorrect
61807	Incorrect spindle direction programmed (active)	CYCLE384M	The programmed spindle direction conflicts with the spindle direction used for the cycle
61808	Final drilling depth or single drilling depth missing	CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Total depth “Z” or single drilling depth “Q” missing from G8x block (first call of cycle)
61809	Drill position not permitted	CYCLE383T, CYCLE384T, CYCLE385T	
61810	ISO G code not possible	CYCLE383T, CYCLE384T, CYCLE385T	
61811	ISO axis name not permitted	CYCLE328, CYCLE330. CYCLE371T, CYCLE374T, CYCLE376T, CLE383T, CYCLE384T, CYCLE385T	An invalid ISO axis name was programmed in the calling block
61812	Value(s) incorrectly specified in the external cycle call	CYCLE371T, CYCLE376T,	An invalid numerical value was programmed in the calling block
61813	GUD value incorrectly defined	CYCLE376T	An invalid numerical value was entered in cycle setting data memory

Table 10-1 Alarm number and alarm description

Alarm no.	Brief description	Source	Explanation/remedy
61814	Polar coordinates not possible	CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	
61815	G40 not active	CYCLE374T, CYCLE376T	Before the cycle call, G40 was not active
61816	Axes not in reference point		
61817	Axis coordinates within protection zone		
61818	Limit values for axis area are identical		

■

Notes

References

A

General Documentation

/BU/ SINUMERIK & SIMODRIVE, Automation Systems for Machine Tools
Catalog NC 60
Order No.: E86060–K4460–A101–B1–7600

/IKPI/ Industrial Communications and Field Devices
Catalog IK PI
Order No.: E86060–K6710–A101–B2–7600

/ST7/ SIMATIC
Products for Totally Integrated Automation and Micro Automation
Catalog ST 70
Order No.: E86060–K4670–A111–A8–7600

/Z/ MOTION-CONNECT
Connection Systems & System Components for SIMATIC, SINUMERIK, MASTERDRIVES and SIMOTION
Catalog NC Z
Order No.: E86060–K4490–A001–B1–7600

Safety Integrated Application Manual
The Safety Program for World Industry
Order No.: 6ZB5000–0AA02–0BA0

Electronic Documentation

/CD1/ The SINUMERIK System (03.04 Edition)
DOC ON CD
(includes all SINUMERIK 840D/840Di/810D/802 and SIMODRIVE publications)
Order No.: 6FC5298–7CA00–0BG0

User Documentation

/AUK/ SINUMERIK 840D/810D (09.99 Edition)
Brief Instructions **AutoTurn Operation**
Order No.: 6FC5298–4AA30–0BP2

/AUP/ SINUMERIK 840D/810D (02.02 Edition)
Operator's Guide **AutoTurn Graphical Programming System**
Programming/Setup
Order No.: 6FC5298–4AA40–0BP3

/BA/ SINUMERIK 840D/810D (10.00 Edition)
MMC Operator's Guide
Order No.: 6FC5298–6AA00–0BP0

/BAD/ SINUMERIK 840D/840Di/810D (03.04 Edition)
HMI Advanced Operator's Guide
Order No.: 6FC5298–6AF00–0BP3

/BAH/ SINUMERIK 840D/840Di/810D (03.04 Edition)
HT 6 Operator's Guide
Order No.: 6FC5298–0AD60–0BP3

/BAK/ SINUMERIK 840D/840Di/810D (02.01 Edition)
Brief Operating Instructions
Order No.: 6FC5298–6AA10–0BP0

/BAM/ SINUMERIK 810D/840D (08.02 Edition)
Operating/Programming **ManualTurn**
Order No.: 6FC5298–6AD00–0BP0

/BAS/ SINUMERIK 840D/840Di/810D (11.03 Edition)
Operating/Programming **ShopMill**
Order No.: 6FC5298–6AD10–0BP2

/BAT/ SINUMERIK 840D/810D (06.03 Edition)
Operating/Programming **ShopTurn**
Order No.: 6FC5298–6AD50–0BP2

/BEM/ SINUMERIK 840D/810D (03.04 Edition)
HMI Embedded Operator's Guide
Order No.: 6FC5298–6AC00–0BP3

/BNM/ SINUMERIK 840D/840Di/810D (03.04 Edition)
Measuring Cycles User's Guide
Order No.: 6FC5298–6AA70–0BP3

/BTDI/ SINUMERIK 840D/840Di/810D (06.04 Edition)
Motion Control Information System (MCIS)
Tool Data Information User Manual
Order No.: 6FC5297–6AE01–0BP1

/CAD/	SINUMERIK 840D/840Di/810D **CAD Reader Operator's Guide** Order No.: (part of the online help)	(03.02 Edition)
/DA/	SINUMERIK 840D/840Di/810D **Diagnostics Guide** Order No.: 6FC5298–7AA20–0BP0	(03.04 Edition)
/KAM/	SINUMERIK 840D/810D Brief Instructions **ManualTurn** Order No.: 6FC5298–5AD40–0BP0	(04.01 Edition)
/KAS/	SINUMERIK 840D/810D Brief Instructions **ShopMill** Order No.: 6FC5298–5AD30–0BP0	(04.01 Edition)
/KAT/	SINUMERIK 840D/810D Brief Instructions **ShopTurn** Order No.: 6FC5298–6AF20–0BP0	(07.01 Edition)
/PG/	SINUMERIK 840D/840Di/810D Programming Guide **Fundamentals** Order No.: 6FC5298–7AB00–0BP0	(03.04 Edition)
/PGA/	SINUMERIK 840D/840Di/810D Programming Guide **Advanced** Order No.: 6FC5298–7AB10–0BP0	(03.04 Edition)
/PGA1/	SINUMERIK 840D/840Di/810D List Manual **System Variables** Order No.: 6FC5298–7AE10–0BP0	(03.04 Edition)
/PGK/	SINUMERIK 840D/840Di/810D Brief Instructions **Programming** Order No.: 6FC5298–7AB30–0BP0	(03.04 Edition)
/PGM/	SINUMERIK 840D/840Di/810D Programming Guide **ISO Milling** Order No.: 6FC5298–6AC20–0BP2	(11.02 Edition)
/PGT/	SINUMERIK 840D/840Di/810D Programming Guide **ISO Turning** Order No.: 6FC5298–6AC10–0BP2	(11.02 Edition)
/PGZ/	SINUMERIK 840D/840Di/810D Programming Guide **Cycles** Order No.: 6FC5298–7AB40–0BP0	(03.04 Edition)

/PI / PCIN 4.4
Software for Data Transfer to/from **MMC Module**
Order No.: 6FX2060–4AA00–4XB0 (German, English, French)
Order from: WK Fürth

/SYI/ SINUMERIK 840Di (02.01 Edition)
System Overview
Order No.: 6FC5298–6AE40–0BP0

Manufacturer/Service Documentation

a) Lists

/LIS/ SINUMERIK 840D/840Di/810D (03.04 Edition)
SIMODRIVE 611D
Lists
Order No.: 6FC5297–7AB70–0BP0

b) Hardware

/ASAL/ SIMODRIVE 611, MASTERDRIVES VC/MC (10.03 Edition)
Planning Guide General Part for **AC Induction Motors**
Order No.: 6SN1197–0AC62–0BP0

/APH2/ SIMODRIVE 611 (10.03 Edition)
Planning Guide **AC Induction Motors 1PH2**
Order No.: 6SN1197–0AC63–0BP0

/APH4/ SIMODRIVE 611 (10.03 Edition)
Planning Guide **AC Induction Motors 1PH4**
Order No.: 6SN1197–0AC64–0BP0

/APH7M/ MASTERDRIVES MC (04.04 Edition)
Planning Guide **AC Induction Motors 1PH7**
Order No.: 6SN1197–0AC66–0BP0

/APH7S/ SIMODRIVE 611 (01.04 Edition)
Planning Guide **AC Induction Motors 1PH7**
Order No.: 6SN1197–0AC65–0BP0

/APL6/ MASTERDRIVES VC/MC (01.04 Edition)
Planning Guide **AC Induction Motors 1PL6**
Order No.: 6SN1197–0AC67–0BP0

/BH/ SINUMERIK 840D/840Di/810D (11.03 Edition)
Operator Component Manual
Order No.: 6FC5297–6AA50–0BP3

/BHA/ SIMODRIVE Sensor (03.03 Edition)
User Manual (HW) **Absolute Value Encoder with Profibus-DP**
Order No.: 6SN1197–0AB10–0YP2

/EMC/ SINUMERIK, SIROTEC, SIMODRIVE (06.99 Edition)
Planning Guide **EMC Guidelines**
Order No.: 6FC5297–0AD30–0BP1

The current Declaration of Conformity is on the Internet under
http://www4.ad.siemens.de

Please enter the ID no. 15257461 in the "Search" field (top right) and click "go".

/GHA/ SINUMERIK/ SIMOTION (02.03 Edition)
ADI4 – Analog Drive Interface for 4 Axes
Equipment Manual
Order No.: 6FC5297–0BA01–0BP1

/PFK6/ SIMODRIVE 611, MASTERDRIVES MC (05.03 Edition)
Planning Guide **Synchronous Servomotors 1FK6**
Order No.: 6SN1197–0AD05–0BP0

/PFK7/ SIMODRIVE 611, MASTERDRIVES MC (01.03 Edition)
Planning Guide **Synchronous Servomotors 1FK7**
Order No.: 6SN1197–0AD06–0BP0

/PFS6/ MASTERDRIVES MC (04.04 Edition)
Planning Guide **Synchronous Servomotors 1FS6**
Order No.: 6SN1197–0AD08–0BP1

/PFT5/ SIMODRIVE (05.03 Edition)
Planning Guide **Synchronous Servomotors 1FT5**
Order No.: 6SN1197–0AD01–0BP0

/PFT6/ SIMODRIVE 611, MASTERDRIVES MC (01.04 Edition)
Planning Guide **Synchronous Servomotors 1FT6**
Order No.: 6SN1197–0AD02–0BP0

/PFU/ SINAMICS, MASTERDRIVES (09.03 Edition)
MICROMASTER
SIEMOSYN Motors 1FU8
Order No.: 6SN1197–0AC80–0BP0

/PHC/ SINUMERIK 810D (11.02 Edition)
Configuring (HW) Manual
Order No.: 6FC5297–6AD10–0BP1

/PHD/ SINUMERIK 840D (11.03 Edition)
Configuring (HW) Manual
Order No.: 6FC5297–6AC10–0BP3

/PJAL/ SIMODRIVE 611, SIMOVERT MASTERDRIVES MC (01.04 Edition)
Planning Guide **Synchronous Servomotors**
General Part for 1FT/1FK Motors
Order No.: 6SN1197–0AD07–0BP1

/PJAS/ SIMODRIVE 611, MASTERDRIVES VC/MC (06.04 Edition)
Planning Guide for Asynchronous Motors
Contents: General Section, 1PH2, 1PH4, 1PH7, 1PL6
Order No.: 6SN1197–0AC61–0BP0

/PJFE/ SIMODRIVE (03.04 Edition)
Planning Guide **Synchronous Build-In Motors 1FE1**
AC Motors for Main Spindle Drives
Order No.: 6SN1197–0AC00–0BP5

/PJF1/ SIMODRIVE (12.02 Edition)
Mounting Guide **Synchronous Build-In Motors 1FE1 051.–1FE1 147**.
AC Motors for Main Spindle Drives
Order No.: 610.43000.02

/PJLM/ SIMODRIVE (06.02 Edition)
Planning Guide **Linear Motors 1FN1, 1FN3**
ALL General Information about Linear Motors
1FN1 Three-Phase AC Linear Motor 1FN1
1FN3 Three-Phase AC Linear Motor 1FN3
CON Connections
Order No.: 6SN1197–0AB70–0BP4

/PJM2/ SIMODRIVE 611, MASTERDRIVES MC (06.04 Edition)
Planning Guide for Synchronous Servomotors
Contents: General Section, 1FT5, 1FT6, 1FK6, 1FK7, 1FS6
Order No.: 6SN1197–0AC20–0BP0

/PJTM/ SIMODRIVE (05.03 Edition)
Planning Guide **Build-in Torque Motors 1FW6**
Order No.: 6SN1197–0AD00–0BP1

/PJU/ SIMODRIVE 611 (02.03 Edition)
Planning Guide **Converters**
Order No.: 6SN1197–0AA00–0BP6

/PKTM/ SIMOVERT MASTERDRIVES (03.04 Edition)
Planning Guide **Complete Torque Motors 1FW3**
Order No.: 6SN1197–0AC70–0BP0

/PMH/ SIMODRIVE Sensor (07.02 Edition)
Planning/Mounting Guide
Hollow-Shaft Measuring System SIMAG H
Order No.: 6SN1197–0AB30–0BP1

/PMH2/
SIMODRIVE Sensor (03.04 Edition)
Planning/Mounting Guide
Hollow-Shaft Measuring System SIMAG H2
Order No.: 6SN1197–0AB31–0BP0

/PMHS/
SIMODRIVE (12.00 Edition)
Mounting Guide **Measuring System for Main Spindle Drives**
Gearwheel encoder SIZAG2
Order No.: 6SN1197–0AB00–0YP3

/PMS/
SIMODRIVE (03.04 Edition)
Planning Guide
ECO Motor Spindle for Main Spindle Drives 2SP1
Order No.: 6SN1197–0AD04–0BP1

/PPH/
SIMODRIVE (12.01 Edition)
Planning Guide **1PH2/1PH4/1PH7 Motors**
AC Induction Motors for Main Spindle Drives
Order No.: 6SN1197–0AC60–0BP0

/PPM/
SIMODRIVE (11.01 Edition)
Planning Guide
Hollow-Shaft Motors for Main Spindle Drives **1PM4 and 1PM6**
Order No.: 6SN1197–0AD03–0BP0

c) Software

/FB1/
SINUMERIK 840D/840Di/810D/FM-NC (03.04 Edition)
Description of Functions **Basic Machine (Part 1)**
(the various sections are listed below)
Order No.: 6FC5297–6AC20–0BP0
A2 Various Interface Signals
A3 Axis Monitoring, Protection Zones
B1 Continuous Path Mode, Exact Stop and Look Ahead
B2 Acceleration
D1 Diagnostic Tools
D2 Interactive Programming
F1 Travel to Fixed stop
G2 Velocities, Setpoint/Actual Value Systems, Closed-Loop Control
H2 Output of Auxiliary Functions to PLC
K1 Mode Group, Channels, Program Operation
K2 Axis Types, Coordinate Systems, Frames,
Actual-Value System for Workpiece, External Zero Offset
K4 Communication
N2 EMERGENCY STOP
P1 Transverse Axes
P3 Basic PLC Program
R1 Reference Point Approach
S1 Spindles
V1 Feeds
W1 Tool Compensation

/FB2/ SINUMERIK 840D/840Di/810D (03.04 Edition)
Description of Functions **Extended Functions (Part 2)**
including FM-NC: Turning, Stepper Motor
(the various sections are listed below)
Order No.: 6FC5297–7AC30–0BP0
A4 Digital and Analog NCK I/Os
B3 Several Operator Panels and NCUs
B4 Operation via PC/PG
F3 Remote Diagnostics
H1 Jog with/without Handwheel
K3 Compensations
K5 Mode Groups, Channels, Axis Exchange
L1 FM-NC Local Bus
M1 Kinematic Transformation
M5 Measurements
N3 Software Cams, Position Switching Signals
N4 Punching and Nibbling
P2 Positioning Axes
P5 Oscillation
R2 Rotary Axes
S3 Synchronous Spindles
S5 Synchronized Actions (SW 3 and earlier/subsequently /FBSY/)
S6 Stepping Motors
S7 Memory configuration
T1 Indexing Axes
W3 Tool Change
W4 Grinding

/FB3/ SINUMERIK 840D/840Di/810D (03.04 Edition)
Description of Functions **Special Functions (Part 3)**
(the various sections are listed below)
Order No.: 6FC5297–7AC80–0BP0
F2 3 to 5 Axis Transformation
G1 Gantry Axes
G3 Cycle Times
K6 Contour Tunnel Monitoring
M3 Coupled Axes and ESR
S8 Constant Workpiece Speed for Centerless Grinding
S9 Setpoint Switching (S9)
T3 Tangential Control
TEO Installation and Activation of Compile Cycles
TE1 Clearance Control
TE2 Analog Axis
TE3 Master/Slave for Drives
TE4 Handling Transformation Package
TE5 Setpoint Exchange
TE6 Machine (MCS) Coupling
TE7 Retrace Support
TE8 Clock-Independent Path-Synchronous Switching Signal Output
V2 Preprocessing
W5 3D Tool Radius Compensation

/FBA/	SIMODRIVE 611D/SINUMERIK 840D/810D Description of Functions **Drive Functions** (the various sections are listed below) Order No.: 6SN1197–0AA80–1BP1 DB1 Operational Messages/Alarm Reactions DD1 Diagnostic Functions DD1 Speed Control Loop DE1 Extended Drive Functions DF1 Enables DG1 Encoder Parameterization DL1 MD of Linear Motor DM1 Calculation of Motor/Power Section Parameters and Controller Data DS1 Current Control Loop DÜ1 Watchdogs/Limitations	(03.04 Edition)
/FBAN/	SINUMERIK 840D/SIMODRIVE 611 DIGITAL Description of Functions **ANA MODULE** Order No.: 6SN1197–0AB80–0BP0	(02.00 Edition)
/FBD/	SINUMERIK 840D Description of Functions **Digitizing** Order No.: 6FC5297–4AC50–0BP0 DI1 Installation DI2 Scanning with Tactile Sensors (scancad scan) DI2 Scanning with Lasers (scancad laser) DI4 Milling Program Generation (scancad mill)	(07.99 Edition)
/FBDM/	SINUMERIK 840D/840Di/810D Description of Functions NC Program Management DNC Machines Order No.: 6FC5297–1AE81–0BP0	(09.03 Edition)
/FBDN/	SINUMERIK 840D/840Di/810D Motion Control Information System (MCIS) Description of Functions **NC Program Management DNC** Order No.: 6FC5297–1AE80–0BP0 DN1 DNC Plant/DNC Cell DN2 DNC IFC SINUMERIK, NC Data Transfer via Network	(03.03 Edition)
/FBFA/	SINUMERIK 840D/840Di/810D Description of Functions **ISO Dialects for SINUMERIK** Order No.: 6FC5297–6AE10–0BP3	(11.02 Edition)
/FBFE/	SINUMERIK 840D/810D Description of Functions **Remote Diagnosis** Order No.: 6FC5297–0AF00–0BP3 FE1 Remote Diagnosis (ReachOut) FE3 RCS Host/RCS Viewer (pcAnywhere)	(03.04 Edition)

/FBH/ SINUMERIK 840D/840Di/810D
HMI Configuring Package (11.02 Edition)
Order No.: (included with the software)

Part 1 User's Guide
Part 2 Description of Functions

/FBH1/ SINUMERIK 840D/840Di/810D
HMI Configuring Package (03.03 Edition)
ProTool/Pro Option SINUMERIK
Order No.: (included with the software)

/FBHL/ SINUMERIK 840D/SIMODRIVE 611 digital (10.03 Edition)
Description of Functions **HLA Module**
Order No.: 6SN1197–0AB60–0BP3

/FBIC/ SINUMERIK 840D/840Di/810D (06.03 Edition)
Motion Control Information System (MCIS)
Description of functions **TDI ID Connection**
Order No.: 6FC5297–1AE60–0BP0

/FBMA/ SINUMERIK 840D/810D (08.02 Edition)
Description of Functions **ManualTurn**
Order No.: 6FC5297–6AD50–0BP0

/FBO/ SINUMERIK 840D/810D (09.01 Edition)
Description of Functions Configuring **OP 030 Operator Interface**
(the various sections are listed below)
Order No.: 6FC5297–6AC40–0BP0
BA Operator's Guide
EU Development Environment (Configuring Package)
PSE Introduction to Configuring of Operator Interface
(IK Screen Kit: Software Update and Configuration)

/FBP/ SINUMERIK 840D (03.96 Edition)
Description of Functions**C PLC Programming**
Order No.: 6FC5297–3AB60–0BP0

/FBR/ SINUMERIK 840D/840Di/810D (01.04 Edition)
Description of Functions **RPC SINUMERIK Computer Link**
Order No.: 6FC5297–6AD61–0BP0
NFL Interface to Host Computer
NPL Interface to PLC/NCK

/FBSI/ SINUMERIK 840D/SIMODRIVE (11.03 Edition)
Description of Functions SINUMERIK **Safety Integrated**
Order No.: 6FC5297–6AB80–0BP2

/FBSP/ SINUMERIK 840D/840Di/810D (11.03 Edition)
Description of Functions **ShopMill**
Order No.: 6FC5297–6AD80–0BP2

/FBST/ SIMATIC (01.01 Edition)
Description of Functions **FM STEPDRIVE/SIMOSTEP**
Order No.: 6SN1197–0AA70–0YP4

/FBSY/ SINUMERIK 840D/810D (03.04 Edition)
Description of Functions **Synchronized Actions**
Order No.: 6FC5297–7AD40–0BP2

/FBT/ SINUMERIK 840D/810D (03.04 Edition)
Description of Functions **ShopTurn**
Order No.: 6FC5297–6AD70–0BP2

/FBTC/ SINUMERIK 840D/810D (01.02 Edition)
IT Solutions
Description of Functions **Tool Data Communication SinTDC**
Order No.: 6FC5297–5AF30–0BP0

/FBTD/ SINUMERIK 840D/810D (02.01 Edition)
IT Solutions
Description of Functions **Tool Requirements Planning (SinTDI)**
with Online Help
Order No.: 6FC5297–6AE00–0BP0

/FBTP/ SINUMERIK 840D/840Di/810D (01.03 Edition)
Motion Control Information System (MCIS)
Description of Functions **Preventive Maintenance TPM**
Order No.: Document is included with the software

/FBU/ SIMODRIVE 611 universal/universal E (07.03 Edition)
Description of Functions **Control Component for**
Speed Control and Positioning
Order No.: 6SN1197–0AB20–0BP8

/FBU2/ SIMODRIVE 611 **universal** (04.02 Edition)
Mounting Instructions
(supplied with every SIMODRIVE 611 universal)

/FBW/ SINUMERIK 840D/810D (11.02 Edition)
Description of Functions **Tool Management**
Order No.: 6FC5297–6AC60–0BP1

/HBA/ SINUMERIK 840D/840Di/810D (03.02 Edition)
Manual **@Event**
Order No.: 6AU1900–0CL20–0BA0

/HBI/ SINUMERIK 840Di (09.03 Edition)
Manual **SINUMERIK 840Di**
Order No.: 6FC5297–6AE60–0BP2

/INC/ SINUMERIK 840D/840Di/810D (06.03 Edition)
System Description **Start-up Tool SINUMERIK SinuCOM NC**
Order No.: (part of the Online Help for the Startup Tool)

/PJE/ SINUMERIK 840D/810D (08.01 Edition)
Description of Functions **HMI Embedded Configuring Package**
Software Update, Configuration, Installation
Order No.: 6FC5297–6EA10–0BP0

/PS/ SINUMERIK 840D/810D (09.03 Edition)
Planning Guide Configuring Syntax
This document is supplied with the software and is available in PDF format

/POS1/ SIMODRIVE **POSMO A** (08.03 Edition)
User Manual **Distributed Positioning Motor on PROFIBUS DP**
Order No.: 6SN2197–0AA00–0BP6

/POS2/ SIMODRIVE **POSMO A** (05.03 Edition)
Installation Guide (included in every POSMO A)

/POS3/ SIMODRIVE **POSMO SI/CD/CA** (07.03 Edition)
User Manual **Distributed Servo Drives**
Order No.: 6SN2197–0AA20–0BP5

/POS4/ SIMODRIVE **POSMO SI** (04.02 Edition)
Installation Guide (included in every POSMO SI)

/POS5/ SIMODRIVE **POSMO CD/CA** (04.02 Edition)
Installation Guide (included in every POSMO CD/CA)

/S7H/ SIMATIC S7–300 (2002 Edition)
Installation Manual **Technological Functions**
Order No.: 6ES7398–8AA03–8BA0
– Reference Manual: CPU Data (HW Description)
– Reference Manual: Module Data

/S7HT/ SIMATIC S7–300 (03.97 Edition)
Manual **STEP 7, Basic Know-How, V. 3.1**
Order No.: 6ES7810–4CA02–8BA0

/S7HR/ SIMATIC S7–300 (03.97 Edition)
Manual **STEP 7, Reference Manuals, V. 3.1**
Order No.: 6ES7810–4CA02–8BR0

/S7S/ SIMATIC S7–300 (04.02 Edition)
Positioning Module **FM 353 for Stepper Drives**
Order in conjunction with Configuring Package

/S7L/ SIMATIC S7–300 (04.02 Edition)
Positioning Module **FM 354 for Servo Drives**
Order in conjunction with Configuring Package

/S7M/ SIMATIC S7–300 (01.03 Edition)
Multi-Axis Module FM 357.2 for Servo and Stepper Drives
Order in conjunction with Configuring Package

/SP/ SIMODRIVE 611–A/611–D
SimoPro 3.1
Program for Configuring Machine Tool Drives
Order No.: 6SC6111–6PC00–0BA☐
Order from: WK Fürth

d) Start-up

/BS/ SIMODRIVE 611 analog (10.00 Edition)
Description **Installation Software for**
Main Spindle and Asynchronous Motor Modules Version 3.20
Order No.: 6SN1197–0AA30–0BP1

/IAA/ SIMODRIVE 611A (10.00 Edition)
Installation & Start-up Guide
Order No.: 6SN1197–0AA60–0BP6

/IAC/ SINUMERIK 810D (11.02 Edition)
Installation & Start-up Guide
(including description of start-up software
SIMODRIVE 611D)
Order No.: 6FC5297–6AD20–0BP1

/IAD/ SINUMERIK 840D/SIMODRIVE 611D (03.04 Edition)
Installation & Start-up Guide
(including description of start-up software
SIMODRIVE 611D)
Order No.: 6FC5297–7AB10–0BP0

/IAM/ SINUMERIK 840D/840Di/810D (03.04 Edition)
Installation & Start-up Guide **HMI**
Order No.: 6FC5297–6AE20–0BP3
AE1 Updates/Extensions
BE1 Supplement Operator Interface
HE1 Online Help
IM2 Installation and Start-Up HMI Embedded
IM4 Installation and Start-Up HMI Advanced
TX1 Creating Foreign Language Texts with Windows 95 / NT
TX1 Creating Foreign Language Texts with Windows 2000 / XP

Notes

Index

Z

Notes

Commands

Characters

Numbers

B

C

D

E

F

G

H

L

M

P

S

T

<table>
<tr><td>To
SIEMENS AG
A&D MC BMS
P.O. Box 3180
D-91050 Erlangen, Germany
(Phone: ++49-(0)180 / 5050-222 [Hotline]
Fax: ++49-(0)9131 / 98-2176 [Documentation]
Email: motioncontrol.docu@erlf.siemens.de)</td><td>Suggestions

Corrections
For Publication/Manual:

SINUMERIK 840D/840Di/810D
ISO Dialects for SINUMERIK
Description of Functions

Manufacturer Documentation</td></tr>
<tr><td>From

Name
Company/Dept.
Address

Phone: /
Fax: /</td><td>Manual

Order No.: 6FC5 297-6AE10-0BP4
Edition: 07.04

Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.</td></tr>
</table>

Suggestions and/or corrections

Overview of SINUMERIK 840D/840Di/810D Documentation (04.2004)

General Documentation

- *SINUMERIK* 840D/810D — Brochure
- *SINUMERIK* 840D/840Di/810D/ — Catalog Ordering info. NC 60 ***)**
- *Safety Integrated* Application Manual — **Safety Integrated** Application Manual

User Documentation

- *SINUMERIK* 840D/810D/FM-NC — **AutoTurn**
  – Short Guide
  – Programming/Setup
- *SINUMERIK* 840D/840Di 810D — Operator's Guide **– HT 6**
- *SINUMERIK* 840D/840Di/810D — **Diagnostics Guide *)**
- *SINUMERIK* 840D/840Di/810D — **Operator's Guide *)**
  – Short Guide
  – HMI Embedded
  – HMI Advanced

User Documentation

- *SINUMERIK* 840D/840Di/810D — **Program. Guide**
  – Short Guide
  – Fundamentals ***)**
  – Advanced ***)**
  – Cycles
  – Measuring Cycles
  – ISO Turning/Milling
- *SINUMERIK* 840D/810D — Operator's Guide
  – **ManualTurn**
  – Short Guide ManualTurn
  – **ShopMill**
  – Short Guide ShopMill
  – **ShopTurn**
  – Short Guide ShopTurn
- *SINUMERIK* 840Di — **System Overview**
- *SINUMERIK* 840D/840Di/810D — Lists **System Variables**

Manufacturer/Service Documentation

- *SINUMERIK* — **Configuring** (HW) ***)**
  – 810D
  – 840D
- *SINUMERIK* 840D/840Di 810D — **Operator Components** (HW) ***)**
- *SINUMERIK* 840D/840Di/810D — Description of Functions
  – **ManualTurn**
  – **ShopMill**
  – **ShopTurn**

Manufacturer/Service Documentation

- *SINUMERIK SIMODRIVE* 611D 840D/810D — Description of Functions **Synchronized Actions**
- *SINUMERIK* 840D/840Di/810D — Desription of Functions **Drive Functions *)**
- *SINUMERIK* 840D/840Di/810D — **Desciption of Functions**
  – Basic Machine ***)**
  – Extended Functions
  – Special Functions
- *SINUMERIK* 840D/810D — Desription of Functions **Tool Management**
- *SINUMERIK* 840D/810D — **Configuring Kit** HMI Embedded
- *SINUMERIK* 840D/810D — Description of Functions Operator Interface **OP 030**
- *SINUMERIK* 840D/810D — **MCIS**
  –Computer Link
  –Tool Data Information System
  –NC Data Management
  –NC Data Transfer
  –Tool Data Communication

Manufacturer/Service Documentation

- *SINUMERIK SIMODRIVE* — Description of Functions SINUMERIK **Safety Integrated**
- *SINUMERIK* 840D — Description of Functions **Digitizing**
- *SINUMERIK SIMODRIVE* — **Installation & Start-Up Guide *)**
  – 810D
  – 840D/611D
  – HMI
- *SINUMERIK SIMODRIVE* 840D/840Di 810D 611D — **Lists*)**
- *SINUMERIK SIMODRIVE* 840D 611D — Description of Functions **Linear Motor**
- *SINUMERIK SIMODRIVE* 840D 611D — Description of Functions
  – Hydraulics Modules
  – Analog Modules
- *MOTION CONTROL SYSTEMS* — **EMC** Guidelines

Electronic Documentation

- *SINUMERIK SIMODRIVE* 840D/840Di/810D 611, Motors — **DOC ON CD *)** The SINUMERIK System

Manufacturer/Service Documentation

- *SINUMERIK* 840D/840Di/810D — Description of Functions **ISO** Dialects for SINUMERIK
- *SINUMERIK* 840Di — **Manual** (HW + Installation and Start-Up)
- *SINUMERIK* 840D/810D — Desription of Functions **Remote Diagnosis**
- *SINUMERIK* 840D/840Di/810D — **Manual @ Event**

***)** These documents are a minimum requirement

Siemens AG
Automation & Drives
Motion Control Systems
P. O. Box 3180, D – 91050 Erlangen
Germany

www.siemens.com/motioncontrol

Order No.: 6FC5297-6AE10-0BP4

Printed in Germany