SIEMENS

SINUMERIK

SINUMERIK 840D sl / 828D ISO Milling

Programming Manual

Valid for

Control
SINUMERIK 840D sl / 840DE sl
SINUMERIK 828D

Software Version
CNC software 4.5

02/2012
6FC5398-7BP40-3BA0

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Order number: 6FC5398-7BP40-3BA0
Ⓟ 10/2012 Technical data subject to change

Table of contents

Principles of programming 1

1.1 Introductory comments

1.1.1 Siemens mode

The following conditions are valid in the Siemens mode:

- The default of the G commands can be defined for each channel via the machine data 20150 $MC_GCODE_RESET_VALUES.
- No language commands from the ISO dialects can be programmed in the Siemens mode.

1.1.2 ISO dialect mode

The following conditions are valid in the active ISO dialect mode:

- The ISO dialect mode can be set with machine data as the default setting of control system. The control system reboots by default in the ISO dialect mode subsequently.
- Only G functions from the ISO dialect can be programmed; the programming of Siemens G functions is not possible in the ISO Mode.
- Mixing of ISO dialect and Siemens language in the same NC block is not possible.
- Switching between ISO Dialect M and ISO Dialect T with a G command is not possible.
- Subroutines that are programmed in the Siemens mode can be called.
- If Siemens functions are to be used, one must first switch to the Siemens mode.

1.1.3 Switching between the modes

The following G functions can be used to switch between the Siemens mode and the ISO dialect mode:

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

The active tool, the tool offsets and work offsets are not influenced by the switchover.

G290 and G291 must be programmed alone in an NC block.

1.1.4 Display of the G code

The G code is displayed in the same language (Siemens or ISO Dialect) as the relevant current block. If the display of the blocks is suppressed with DISPLOF, the G codes continue to be displayed in the language in which the active block is displayed.

Example

The G functions of the ISO dialect mode are used to call the Siemens standard cycles. To do this, DISPLOF is programmed at the start of the relevant cycle; this way the G functions that are programmed in the ISO dialect language continue to be displayed.

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

Procedure

The Siemens shell cycles are called via main programs. The Siemens mode is selected automatically by calling the shell cycle.

With DISPLOF, the block display is frozen on calling the cycle; the display of the G code continues in the ISO Mode.

The G codes that were changed in the shell cycle, are reset to their original status at the end of the cycle with the "SAVE" attribute.

1.1.5 Maximum number of axes/axis identifiers

The maximum number of axes in the ISO dialect mode is 9. The axis identifiers for the first three axes are defined permanently with X, Y and Z. All other axes can be identified with the letters A, B, C, U, V and W.

1.1.6 Decimal point programming

In the ISO dialect mode, there are two notations for evaluating programmed values without decimal point:

- Pocket calculator notation

  Values without decimal points are interpreted as mm, inch or degree.

- Standard notation

  Values without decimal point are multiplied by a conversion factor.

The setting is done over MD10884 $MN_EXTERN_FLOATINGPOINT_PROG.

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

Example:

Linear axis in mm:

- X 100.5

  corresponds to a value with decimal point: 100.5 mm

- X 1000
  - Pocket calculator notation: 1,000 mm
  - Standard notation:

    IS-B: 1,000* 0.001= 1 mm

    IS-C: 1,000* 0.0001= 0.1 mm

ISO dialect milling

Table 1- 1 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	Degree	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 lead	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 IPO parameters	mm inch	0.001 0.0001	0.0001 0.00001
G04 X or U	s	0.001	0.001
A angle contour definition	Degree	0.001	0.0001
G74, G84 tapping cycles $MC_EXTERN_FUNCTION_MASK Bit8 = 0 F as feed such as G94, G95 Bit8 = 1 F as thread lead			

1.1.7 Comments

In the ISO dialect mode, brackets are interpreted as comment signs. In the Siemens mode, ";" is interpreted as comment. To simplify matters, an ";" is also understood as comment in the ISO dialect mode.

If the comment start sign '(' is used inside a comment again, the comment is ended only if all the open brackets are closed again.

Example:

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

X100 Y100 is executed in block N5 and N10, but only Y100 in block N15, because the first bracket is closed only after X100. Everything up to that point is interpreted as comment.

1.1.8 Skip block

The sign of skipping or suppression of blocks "/" can be used at any convenient position in a block, i.e. even in the middle of the block. If the programmed block skip level is active on the date of the compilation, the block is not compiled from this point up to the end of the block. An active block skip level has the same effect as a block end.

Example:

```
N5 G00 X100. /3 YY100 --> Alarm 12080 "Syntax error"
N5 G00 X100. /3 YY100 --> no alarm, if block skip level 3 is active
```

Block skip signs within a comment are not interpreted as block skip signs

Example:

```
N5 G00 X100. ( /3 Part1 ) Y100
;the Y axis is traversed even when the block skip level 3 is active
```

The block skip levels /1 to /9 can be active. Block skip values <1 and >9 lead to alarm 14060 "Impermissible skip level for differential block skip".

The function is mapped to the existing Siemens skip levels. Unlike the ISO Dialect original, "/" and "/1" are separate skip levels that must also be activated separately.

Note

The "0" in "/0" can be omitted.

1.2 Preconditions for the feed

The following Section describes the feed function with which the feedrate (covered path per minute or per rotation) of a cutting tool is defined.

1.2.1 Rapid traverse

Rapid traverse is used for positioning (G00) as well as for manual traverse with rapid traverse (JOG). In rapid traverse, each axis is traversed with the rapid traverse rate set for the individual axes. The rapid traversing rate is defined by the machine manufacturer and it is specified by the machine data for the individual axes. As the axes traverse independently of each other, each axis reaches its target point at a different time. Hence, the resulting tool path is generally not a straight line.

1.2.2 Path feed (F function)

Note

Unless something else is specified, the unit "mm/min" always stands for feedrate of the cutting tool in this documentation.

The feed with which a tool should be traversed in linear interpolation (G01) or circular interpolation (G02, G03) is designated with the address character "F".

The feed of the cutting tool in "mm/min" is specified after the address character "F".

The permissible range of F values is specified in the documentation of the machine manufacturer.

Possibly, the feed is limited by the servo system and the mechanical system in the upward direction. The maximum feed is set in the machine data and limited to the value defined there before an overshoot.

The path feed is generally composed of the individual speed components of all geometry axes participating in the movement and refers to the cutter center (see the two following figures).

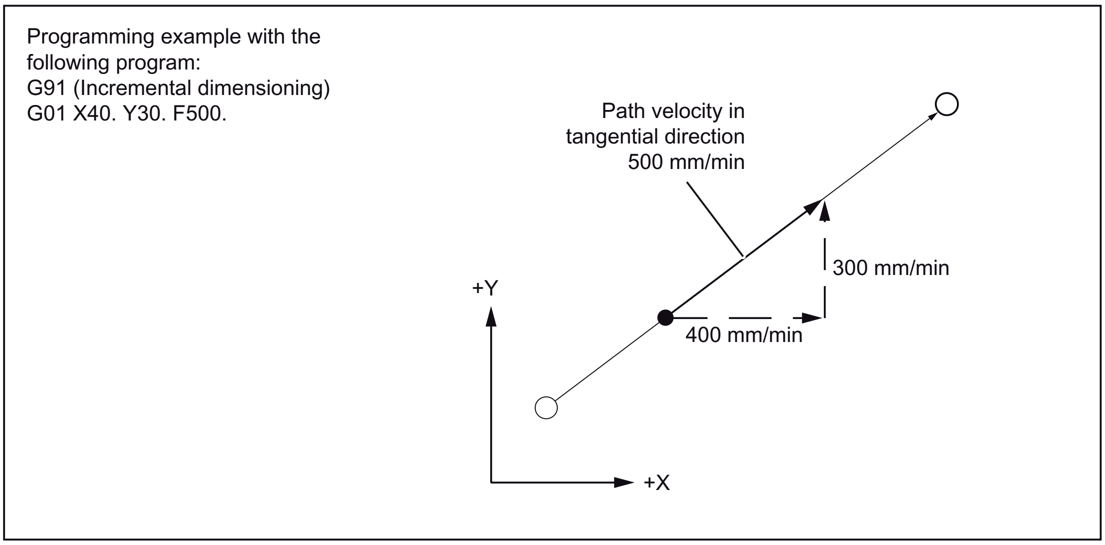


Figure 1-1 Linear interpolation with 2 axes

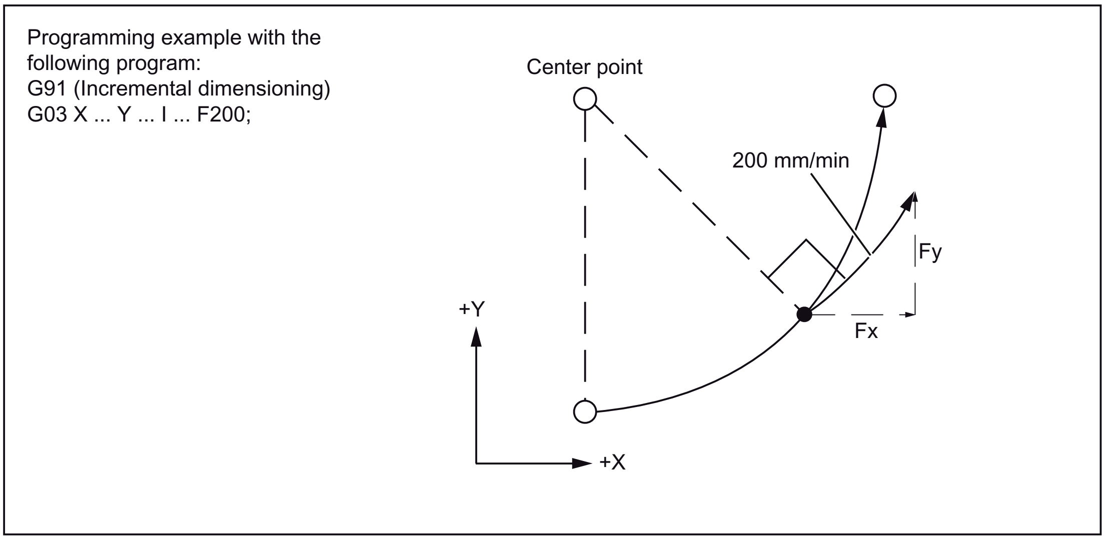

Programming example with the
following program:
G91 (Incremental dimensioning)
G03 X ... Y ... I ... F200;
Center point
200 mm/min
Fy
+Y
Fx
+X

Figure 1-2 Circular interpolation with 2 axes

In 3D interpolation, the feed of the resulting straight lines programmed with F are maintained in the space.

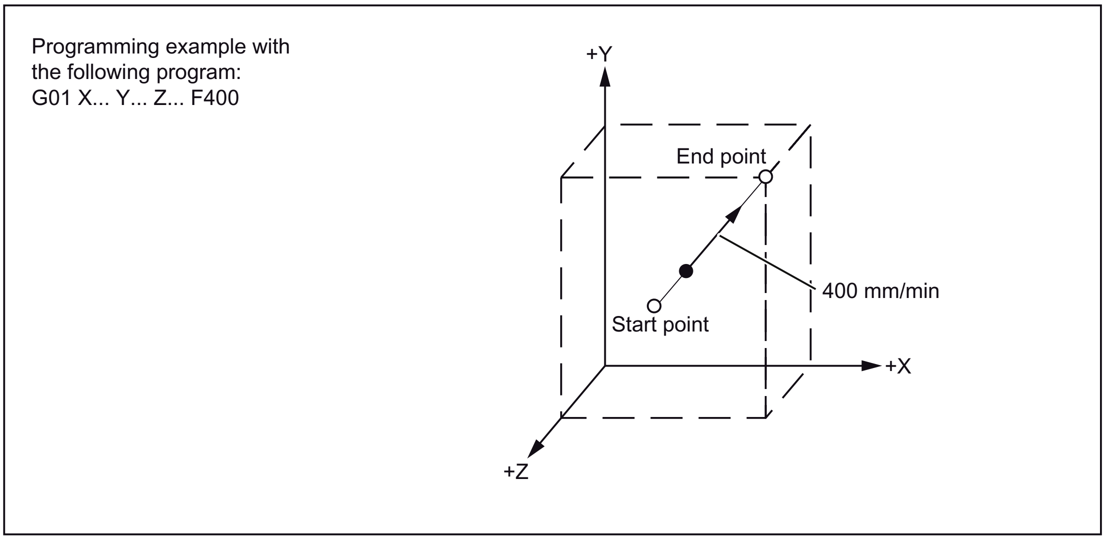


Figure 1-3 Feed in case of 3D interpolation

Note

If "F0" is programmed and the function "Fixed feedrate" is not active, then the Alarm 14800 "Programmed path velocity less than or equal to zero" is output.

1.2.3 Fixed feedrates F0 to F9

Activate feed values

Ten different feed values pre-set via setting data can be activated with F0 to F9. To activate the rapid traverse rate with F0, the corresponding speed must be entered in the setting data 42160 $SC_EXTERN_FIXED_FEEDRATE_F1_F9[0].

The feed values for F0 to F9 are entered in the setting data as real values. An evaluation of the input values is not undertaken.

The function is activated via the machine data 22920 $MC_EXTERN_FIXED_FEEDRATE_F1_ON. If the machine data is set to FALSE, F1 - F9 is interpreted as normal feed programming, e.g. F2 = 2 mm/min, F0=0 mm/min.

If the machine data = TRUE, the feed values for F0 - F9 are fetched from the setting data 42160 $SC_EXTERN_FIXED_FEEDRATE_F1_F9[ ]. If the value 0 exists in one of the setting data, then the corresponding address extension of feed 0 is activated during the programming.

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 at 5000 mm/min
N20 G01 X150 Y30 F1          ;Feed 1000 mm/min active
N30 Z0 F2                    ;Position approached at 500 mm/min
N40 Z10 F0                   ;Approach position at 5000 mm/min
```

Table 1- 2 Setting data for the default setting of feedrate F

F function	Setting Data
F0	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[0]
F1	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[1]
F2	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[2]
F3	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[3]
F4	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[4]
F5	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[5]
F6	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[6]
F7	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[7]
F8	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[8]
F9	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[9]
Note: Input format = REAL	

Note

If the function is activated with MD $MC_EXTERN_FIXED_FEEDRATE_F1_ON and if the feed value from the setting data is not to be active with F1 to F9, then the feed value is programmed as actual value. If, for instance, a feed value should be programmed at 1 mm/min, the feed must be programmed with F1.0 instead of F1.

If the "DRY RUN" (test run) switch is set to "ON", all the feed commands are traversed at the feed set for the test run.

The Feed Override function is effective even for the fixed feedrates F0 to F9.

The feed set in the setting data is stored even after the control system is switched off.

In a macro call with G65/G66, the value programmed with F is stored in the system variable $C_F, i.e. the numeric values 0 to 9 are stored.

If, in a cycle call, a fixed feed (F0 - F9) is programmed in a machining program, the feed value is read from the relevant setting data and stored in the variable $C_F.

Example

$SC_FIXED_FEEDRATE_F1_F9[0] = 1500.0

$SC_FIXED_FEEDRATE_F1_F9[1] = 550.0

```
N10 X10 Y10 Z10 F0 G94       ;Positioning with 1500
N20 G01 X150 Y30 F1          ;Feed 550 mm/min active
N40 Z10 F0                   ;Positioning with 1500
```

Note

While macroprogramming with G65/66, the programmed value for the address F is always stored in the cycle system variable. For F1 to F9, for example, the value 1 to 9 is entered in the cycle system variable $C_F. The address signifies a transfer variable here and has no direct reference to the feed.

The same is true of the thread lead programming in G33 - G34 with the address F. No feed is programmed with F here, instead the distance between two threads during a spindle revolution.

In cycle programming (e.g., G81 X.. Y.. Z.. R.. P.. Q.. F..), the feed is always programmed under the address F. In a part program block with a cycle call over a G function (G81 - G87 etc.), the corresponding feed value during the programming of F1 to F9 is written from the corresponding setting data in the variable $C_F.

Restriction

In the ISO dialect mode, the feed values are changed in the setting data with a handwheel. In the Siemens mode, the feeds can be influenced only like a directly programmed feed, e.g. through the override switch.

1.2.4 Linear feed (G94)

On specifying G94, the feed given after the address character F is executed in the mm/min, inch/min or degree/min unit.

1.2.5 Inverse-time feed (G93)

On specifying G93, the feed given after the address character F is executed in the 1/min unit. G93 is a modally effective G function.

Example

```
N10 G93 G1 X100 F2 ;
```

i.e., the programmed path is traversed within half a minute.

1.2.6 Revolutional feedrate (G95)

On entering G95, the feed is executed in the mm/revolution unit or inch/revolution related to the master spindle.

Note

All of the commands are modal. If the G feed command is switched among G93, G94 or G95, the path feed must be reprogrammed. The feed can also be specified in degree/revolution for the machining with rotary axes.

Drive commands 2

2.1 Interpolation commands

The positioning and interpolation commands, with which the tool path along the programmed contour, such as a straight line or a circular arc, is monitored, are described in the next Section.

2.1.1 Rapid traverse (G00)

You can use rapid traverse to position the tool rapidly, to traverse around the workpiece or to approach tool change points.

The following G functions can be used to call the positioning (refer to following table):

Table 2- 1 G function for positioning

G function	Function	G group
G00	Rapid traverse	01
G01	Linear movement	01
G02	Circle/helix in the clockwise direction	01
G02.2	Involute in the clockwise direction	01
G03	Circle/helix in the counterclockwise direction	01
G03.2	Involute in the counterclockwise direction	01

Positioning (G00)

Format

G00 X... Y... Z... ;

Explanation

The tool movement programmed with G00 is executed at the highest possible traversing speed (rapid traverse). The rapid traverse rate is defined separately for each axis in machine data. If the rapid traverse movement is executed simultaneously on several axes, the rapid traverse rate is determined by the axis which requires the most time for its section of the path.

Axes that are not programmed in a G00 block are not traversed. In positioning, the individual axes traverse independently of each other with the rapid traverse rate specified for each axis. The precise speeds of your machine can be consulted in the documentation of the manufacturer.

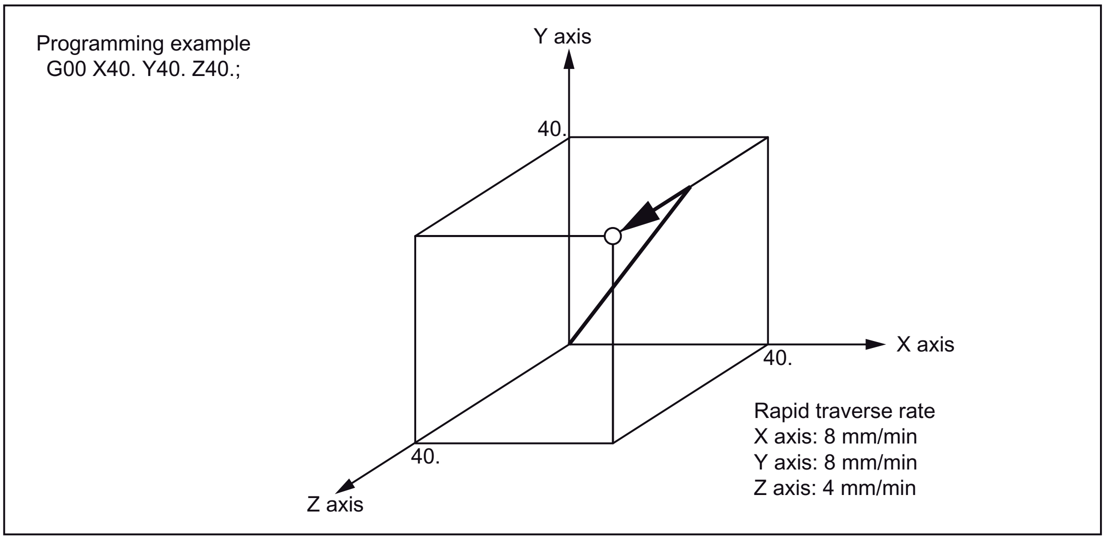


Figure 2-1 Positioning in the run state with 3 simultaneously controllable axes

Note

As in positioning with G00, the axes traverse independently of each other (not interpolated), each axis reaches its end point at a different time. Hence, one must be very careful in positioning with several axes, so that a tool does not collide with a workpiece of the tool during the positioning.

Linear interpolation (G00)

Linear interpolation with G00 is defined by setting the machine data 20732 $MC_EXTERN_GO_LINEAR_MODE. Here, all programmed axes traverse in rapid traverse with linear interpolation and reach their target positions simultaneously.

2.1.2 Linear interpolation (G01)

With G01 the tool travels on paraxial, inclined or straight lines arbitrarily positioned in space. Linear interpolation permits machining of 3D surfaces, grooves, etc.

Format

G01 X... Y... Z... F... ;

In G01, the linear interpolation is executed with the path feed. The axes that are not specified in the block with G01 are not traversed. The linear interpolation is programmed as in the example given above.

Feed F for path axes

The feedrate is specified under the address F. Depending on the default setting in the machine data, the units of measurement specified with the G commands (G93, G94, G95) are either in mm or inch.

One F value can be programmed per NC block. The unit of feedrate is defined over one of the mentioned G commands. The feed F acts only on path axes and remains active until a new feed value is programmed. Separators are permitted after address F.

Note

An alarm is triggered while executing a G01 block if no feed was programmed in a block with G01 or in the previous blocks.

The end point can be specified either as absolute or as incremental. Details are available in Chapter "Absolute/incremental dimensioning".

Programming example
G01 X40. Y40. Z40. F100;

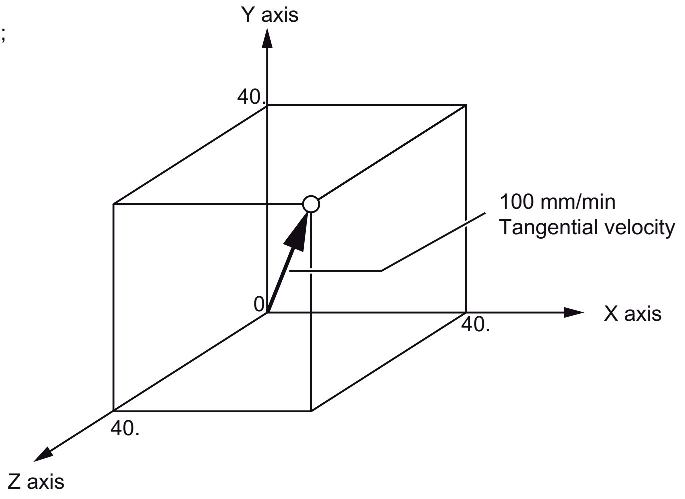


Figure 2-2 Linear interpolation

2.1.3 Circular interpolation (G02, G03)

Format

To start the circular interpolation, please execute the commands specified in the following table.

Table 2- 2 Commands to be executed for circular interpolation

Element	Command	Description
Designation of the plane	G17	Circular arc in Plane X-Y
	G18	Circular arc in Plane Z-X
	G19	Circular arc in Plane Y-Z
Direction of rotation	G02	clockwise
	G03	counterclockwise
End-point position	Two axes from X, Y or Z	End-point position in a workpiece coordinate system
	Two axes from X, Y or Z	Distance of start point - end point with sign
Distance between start point - center	Two axes from I, J or K	Distance start point - circle center with sign
Radius of circular arc	R	Radius of circular arc
Feed	F	Speed along the circular arc

Designation of the plane

With the commands specified below, a tool traverses along the specified circular arc in the plane X-Y, Z-X or Y-Z, so that the feed specified with "F" is maintained on the circular arc.

- in Plane X-Y:

  G17 G02 (or G03) X... Y... R... (or I... J...) F... ;
- in Plane Z-X:

  G18 G02 (or G03) Z... X... R... (or K... I...) F... ;
- in the Plane Y-Z:

  G19 G02 (or G03) Y... Z... R... (or J... K...) F... ;

Before the circle radius programming (with G02, G03), one must first select the desired interpolation plane with G17, G18 or G19. Circular interpolation is not allowed for the 4th and 5th axes, if these are linear axes.

Plane selection is also used to select the plane in which the tool radius compensation (G41/G42) is performed. The Plane X-Y (G17) is automatically set after activating the control system.

G17	X-Y plane
G18	Z-X plane
G19	Y-Z plane

The working planes should be specified, in general.

Circles can also be created outside the selected working plane. In this case, the axis addresses (specification of circle end positions) determine the circular plane.

Circular interpolation is possible in the Xβ, Zβ or Yβ plane while selecting an optional 5th linear axis, which also contains a 5th axis besides the X-Y, Y-Z and Z-X planes (β=U, V or W)

- Circular interpolation in the Xβ plane

  G17 G02 (or G03) X... β... R... (or I... J...) F... ;
- Circular interpolation in the Zβ plane

  G18 G02 (or G03) Z... β... R... (or K... I...) F... ;
- Circular interpolation in the Yβ plane

  G19 G02 (or G03) Y... β... R... (or J... K...) F... ;
- If the address characters for the 4th and 5th axes are omitted - such as in the commands "G17 G02 X... R... (or I... J...) F... ;", then the X-Y plane is selected automatically as the interpolation plane. Circular interpolation with the 4th and 5th axes is not possible if these additional axes are rotary axes.

Direction of rotation

The direction of rotation of the circular arc is to be specified as given in the following figure.

G02	clockwise
G03	counterclockwise

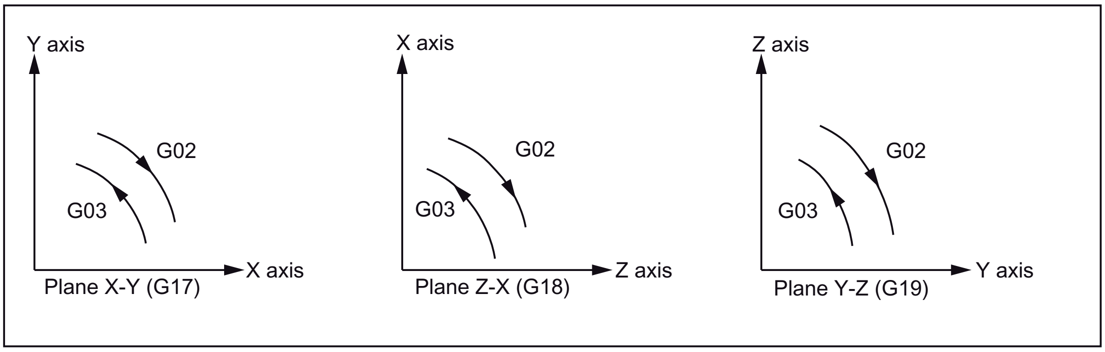


Figure 2-3 Direction of rotation of the circular arc

End point

The end point can be specified corresponding to the definition with G90 or G91 as absolute or incremental (not in G Code System A!).

If the specified end point does not lie on the circular arc, the system outputs Alarm 14040 "Error in end point of circle".

Possibilities of programming circular movements

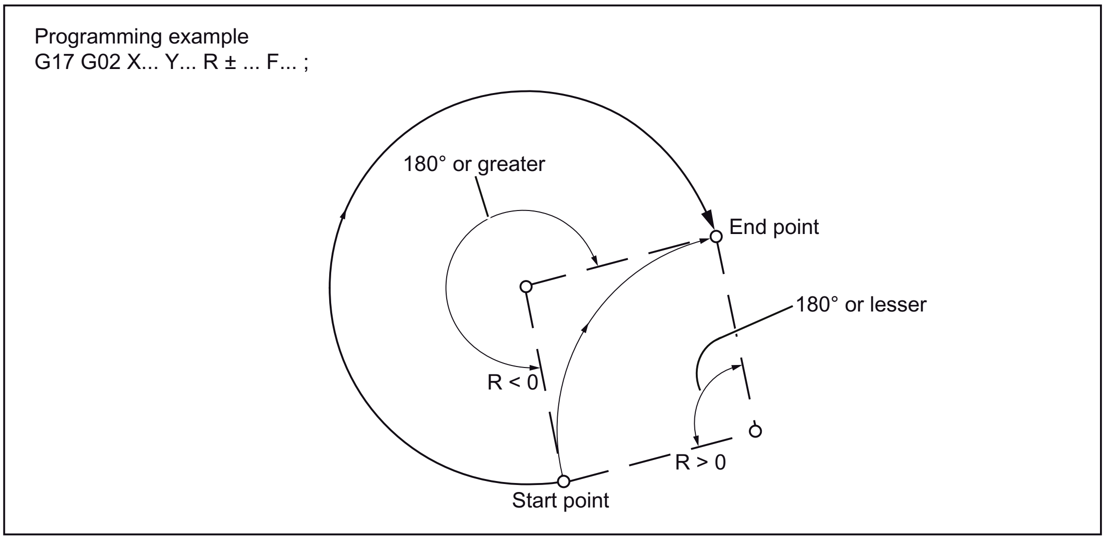
The control system offers two options of programming circular movements.

The circular motion is described by the:

- Center point and end point in the absolute or incremental dimension (default)
- Radius and end point in Cartesian coordinates

  For a circular interpolation with a central angle <= 180 degree, the programming should be "R > 0" (positive).

  For a circular interpolation with a central angle > 180 degree, the programming should be "R < 0" (negative).



Programming example

```
G17 G02 X... Y... R ± ... F... ;
```

Figure 2-4 Circular interpolation with specification of Radius R

Feed

During the circular interpolation, the feed can be specified exactly as during linear interpolation (see Chapter "Linear interpolation (G01)").

2.1.4 Contour definition programming and addition of chamfers or radiuses

Chamfers or radiuses can be added after each traversing block between linear and circular contours. For example, to grind sharp edges of workpieces.

The following combinations are possible during addition:

- between two straight lines
- between two circular arcs
- between a circular arc and a straight line
- between a straight line and a circular arc

Format

, C...; Chamfer

, R...; Rounding

Example

```
N10 G1 X10. Y100. F1000 G18
N20 A140 C7.5
N30 X80. Y70. A95.824, R10
```

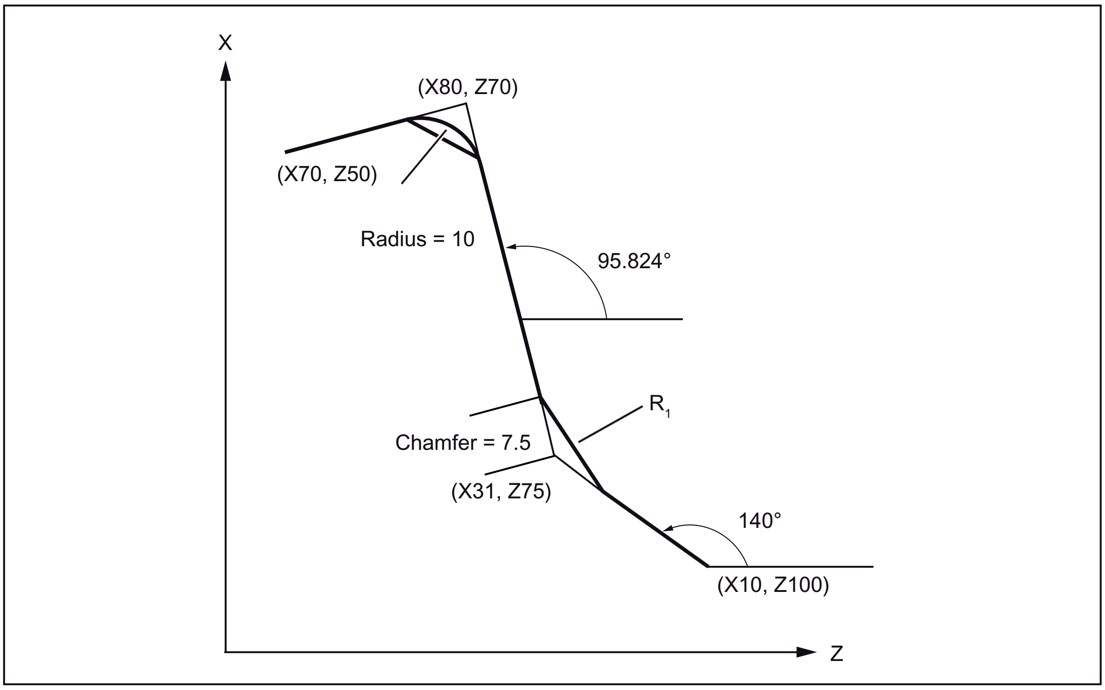


Figure 2-5 3 straight lines

ISO dialect mode

In the ISO dialect original, the C address can be used as axis name as well as for denoting a chamfer on the contour.

The Address R can either be a cycle parameter or an identifier of the radius of a contour.

To differentiate between these two possibilities, a comma "," must be used while programming the contour definition before the address "R" or "C".

Siemens mode

The identifiers of chamfer and radius are defined in the Siemens mode using the machine data. Name conflicts can be avoided this way. There should be no comma before the identifier of the radius or chamfer. The following machine data (MD) is used:

MD for the radius: $MN_RADIUS_NAME

MD for the chamfer: $MN_CHAMFER_NAME

Selection of plane

Chamfer or fillet is possible only in the plane specified through the plane selection (G17, G18 or G19). These functions cannot be used on parallel axes.

Note

No chamfer/rounding is inserted, if

- No straight- or circular contour is available in the plane,
- a movement takes place outside the plane,
- The plane is changed or a number of blocks specified in the machine data, that do not contain any information about traversing (e.g., only command outputs), is exceeded.

Coordinate system

After a block that changes the coordinate system (G92 or G52 to G59) or that contains a command of reference point approach (G28 to G30), should not contain any command for chamfering or rounding of corners.

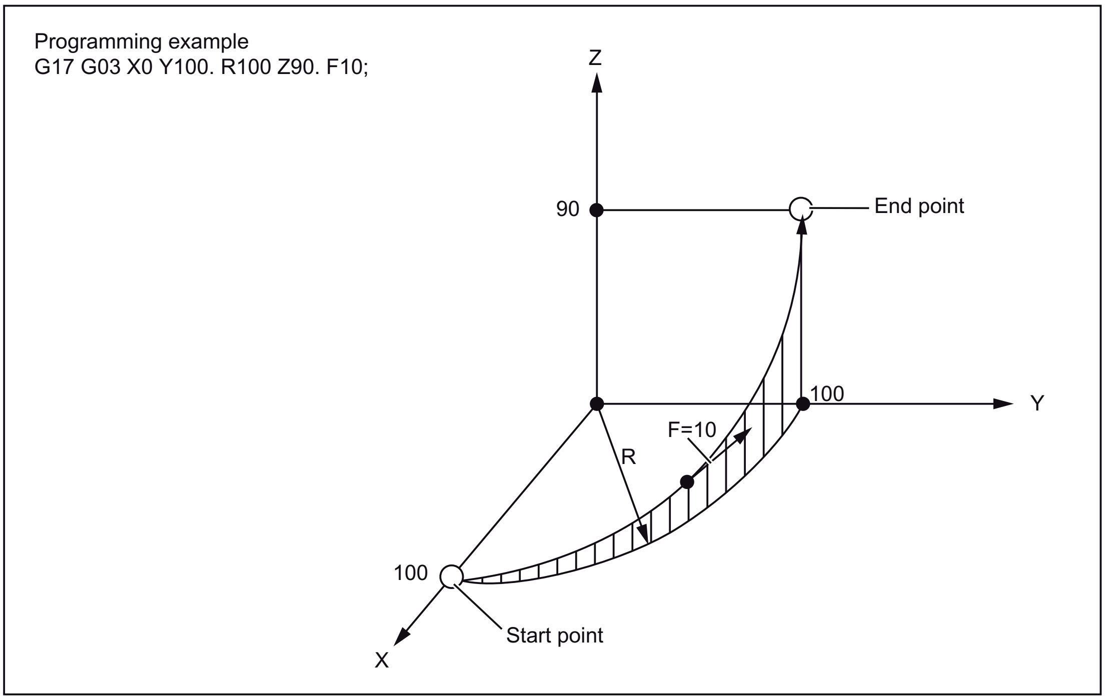
Thread cutting

The specification of fillet in thread cutting blocks is not permissible.

2.1.5 Helical interpolation (G02, G03)

With helical interpolation, two motions are superimposed and executed in parallel:

- A plane circular motion on which
- A vertical linear motion is superimposed.



Programming example

```
G17 G03 X0 Y100. R100 Z90. F10;
```

Figure 2-6 Helical interpolation

Note

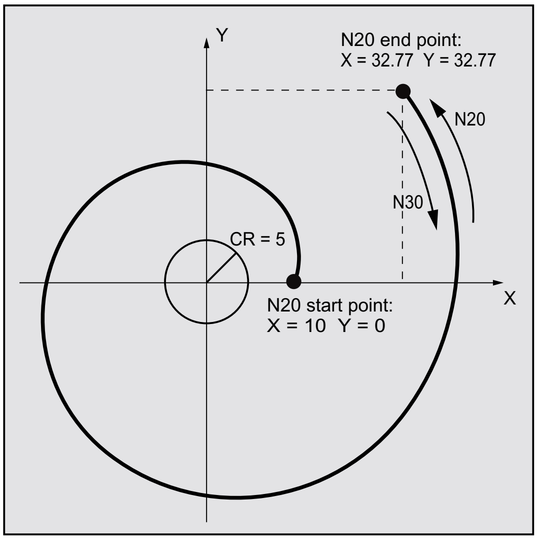
G02 and G03 are modal. The circular motion is performed in those axes that are defined by the specification of the working plane.

For detailed description of the interpolation parameters in case of helical interpolation, refer to "Programming Manual Fundamentals".

2.1.6 Involute interpolation (G02.2, G03.2)

Overview

The involute of a circle is a curve traced out from the end point on a "piece of string" unwinding from the curve. The involute interpolation allows trajectories along an involute. It is executed in the plane in which the base circle is defined. If the starting point and the end point are not in this plane, then, analogous to the helical interpolation for circles, there is a superimposition to a curve in space.



An involute can be traversed in space in case of additional specification of paths vertical to the active plane.

Format

G02.2 X... Y... Z... I... J... K... R

G03.2 X... Y... Z... I... J... K... R

G02.2:	Travel on an involute in clockwise direction
G03.2:	Travel on an involute in counterclockwise direction
X Y Z:	End point in Cartesian coordinates
I J K:	Center of the base circle in Cartesian coordinates
R:	Radius of the base circle

Supplementary conditions

Both the starting point and the end point must be outside the area of the base circle of the involute (circle with radius R around the center point specified by I, J, K). If this condition is not satisfied, an alarm is generated and the program execution is aborted.

Note

For more information about machine data and supplementary conditions that are relevant to involute interpolation, please see References: /FB1/, A2 Chapter "Settings for involute interpolation".

2.1.7 Cylindrical interpolation (G07.1)

Randomly running grooves can be cut on cylindrical workpieces with Function G07.1 (cylindrical interpolation). The path of the grooves is programmed with reference to the unwinded, plane surface of the cylinder.

The G functions specified below can be used to switch the operation of cylindrical interpolation on or off.

Table 2- 3 G functions for activating/deactivating the cylindrical interpolation

G function	Function	G group
G07.1	Operation with cylindrical interpolation	16

Format

```
G07.1 A (B, C) r              ;Activation of operation with cylindrical interpolation
G07.1 A (B, C) 0              ;Deselection of operation with cylindrical interpolation
```

A, B, C: Address of the rotary axis

r: Radius of the cylinder

No other commands should be present in the block containing G07.1.

The G07.1 command is modal. Once G07.1 is specified, the cylindrical interpolation remains active till G07.1 A (B, C) is deselected. The cylindrical interpolation is deactivated in closed position or after NC RESET.

Note

G07.1 is based on the Siemens option TRACYL. Appropriate machine data is to be set for this.

The corresponding data on this is available in the manual "Extended Functions", Section M1, TRACYL.

Programming example

At the cylindrical plane (it arises because the circumference of a cylindrical workpiece is rolled off), in which the Z-axis is accepted as the linear axis and the A-axis as the rotary axis, the following program is written:

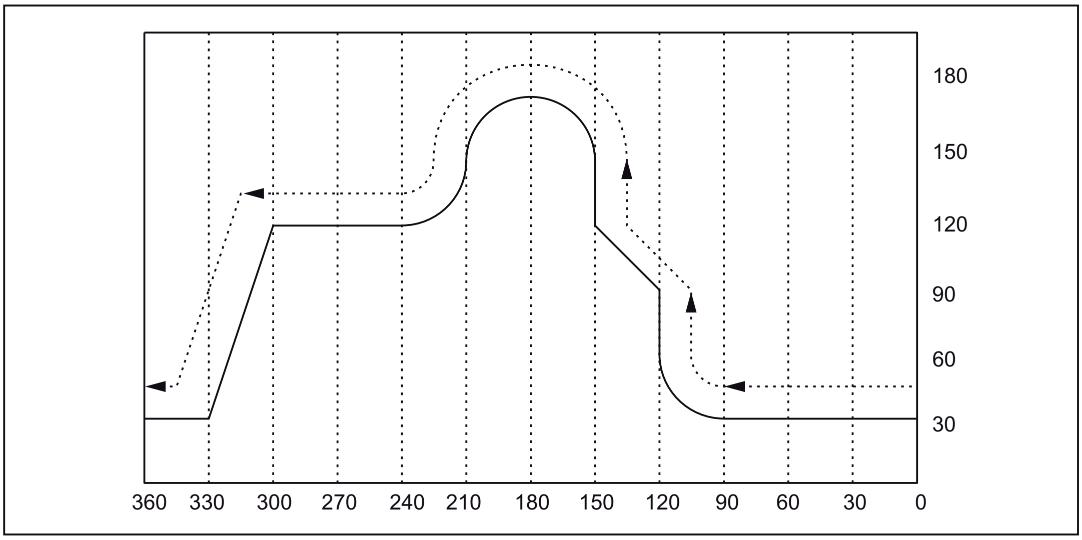


Figure 2-7 G07.1 - Programming example

Program

```
M19
G40
G00 Z30. A-10.
G07.1 A57.296                  ;Operation with cylindrical interpolation ON
                               ;(workpiece radius = 57.926)
G90
G42 G01 A0 F200
G00 X50.
G01 A90. F100
G02 A120. Z60. R30
G01 Z90.
Z120. A150.
Z150.
G03 Z150. A210. R30.
G02 Z120. A240. R30
G01 A300.
Z30. A330.
A360.
G00 X100.
G40 G01 A370.
G07.1 A0                       ;Operation with cylindrical interpolation OFF
G00 A0
```

Programming in operation with cylindrical interpolation

Only the following G functions can be used in cylindrical interpolation: G00, G01, G02, G03, G04, G40, G41, G42, G65, G66, G67, G90, G91 and G07.1. In operation with G00 only those axes can be used that are not involved at the cylindrical plane.

The following axes cannot be used as a positioning axis or a reciprocating axis:

1. The geometry axis in the peripheral direction of the surface of the cylinder (Y axis)
2. The additional linear axis for groove side offset (Z axis)

Relations between the cylindrical interpolation and operations with reference to the coordinate system

- The functions mentioned below should not be used in operation with cylindrical interpolation.
  - Mirroring
  - Scaling (G50, G51)
  - Rotation of the coordinate system (G68)
  - Setting the basic coordinate system
- The relevant overrides (rapid traverse, JOG, spindle speed) are effective.
- On deselecting this operation with cylindrical interpolation, the interpolation plane that was selected before the operation with the cylindrical interpolation was called becomes active again.
- To perform the tool length compensation, the command for the tool length compensation is to be written before specifying the G07.1 command.
- The work offset (G54 - G59) is also to be written before specifying the G07.1 command.

2.2 Reference point approach with G functions

2.2.1 Reference point approach with intermediate point (G28)

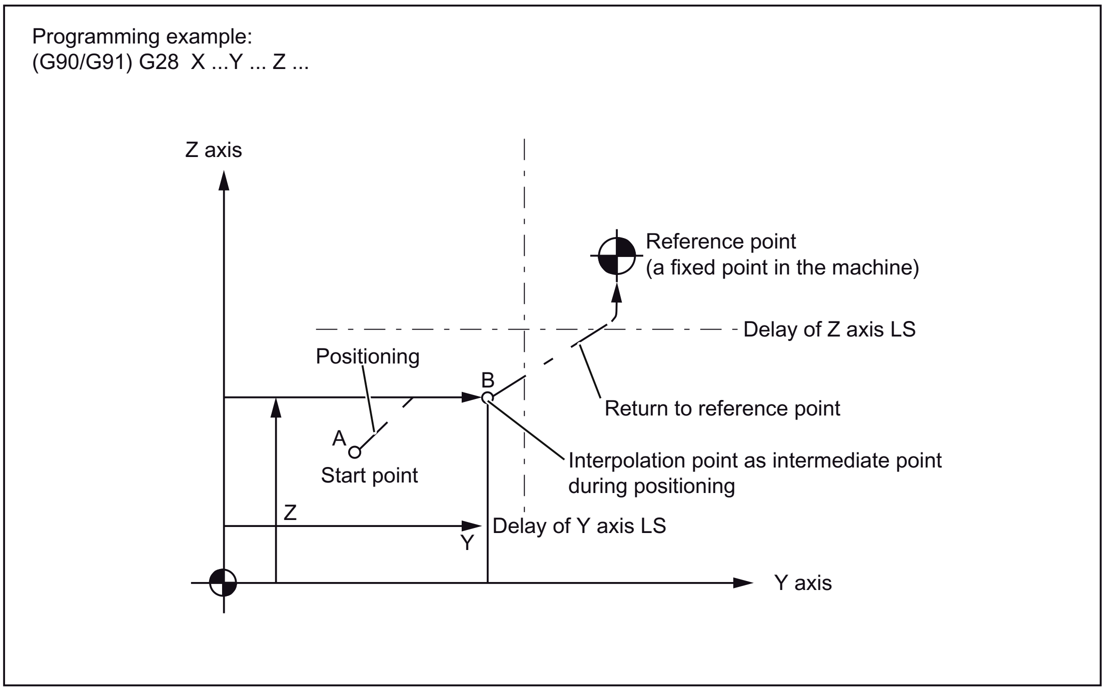
Format

G28 X... Y... Z... ;

The commands "G28 X... Y... Z... ;" can be used to traverse the programmed axes to their reference point. Here, the axes are first traversed to the specified position with rapid traverse, and from there to the reference point automatically. The axes not programmed in the block with G28 are not traversed to their reference point.

Reference position

When the machine has been powered up (where incremental position measuring systems are used), all the axes must approach their reference mark. Only then can traversing movements be programmed. The reference point can be approached in the NC program with G28. The reference point coordinates are defined with the machine data 34100 $_MA_REFP_SET_POS[0] up to [3]). A total of four reference positions can be defined.



Programming example:
(G90/G91) G28 X ...Y ... Z ...

Figure 2-8 Automatic reference point approach

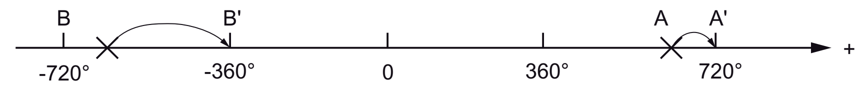
Return to reference point

Note

The G28 function is implemented with the shell cycle cycle328.spf. A transformation must not be programmed for an axis which is to approach the reference point with G28 which must approach the reference mark. The transformation is deactivated in cycle328.spf with command TRAFOOF.

Automatic reference point approach for rotary axes

Rotary axes can be used for automatic reference point approach exactly as linear axes. The approach direction of the reference traverse is defined with the machine data 34010 MD_$MA_REFP_CAM_DIR_IS_MINUS.



(Return to reference point: The positive traversing direction was selected as the direction of return to the reference point.)

Figure 2-9 Return to reference point - rotary axes

Additions to the commands for automatic reference point approach:

Tool radius compensation and defined cycles

G28 should not be used in operation with tool radius compensation (G41, G42) or in a defined cycle!

WARNING

Deactivation of the tool radius compensation

G28 is used to interrupt the tool radius compensation (G40) with eventual axis traverse movement to the reference point. Hence, tool radius compensation is to be deactivated before G28 is issued.

Tool offset in G28

In G28, the interpolation point is approached with the current tool offset. The tool offset is deselected when the reference point is finally approached.

2.2.2 Checking the reference position (G27)

Format

G27 X... Y... Z... ;

This function is used to check whether the axes are on their reference point.

Test procedure

If the check with G27 is successful, the processing is continued with the next part program block. If one of the axes programmed with G27 is not on the reference point, Alarm 61816 "Axes not on reference point" is triggered and the Automatic mode is interrupted.

Note

Function G27 is implemented with the cycle cycle 328.spf as with G28.

To avoid a positioning error, the function "mirroring" should be deselected before executing G27.

2.2.3 Reference point approach with reference point selection (G30)

Format

G30 Pn X... Y... Z... ;

For the commands "G30 Pn X... Y... Z;" the axes are positioned on the specified intermediate point in the continuous-path mode, and finally traversed to the reference point selected with P2 - P4. With "G30 P3 X30. Y50.;", the X- and Y-axes return to the third reference point. The second reference point is selected on omitting "P". Axes that are not programmed in a G30 block are also not traversed.

Reference point positions

The positions of all the reference points are always determined in relation to the first reference point. The distance of the first reference point from all subsequent reference points is set in the following machine data:

Table 2- 4 Reference points

Element	MD
2nd reference point	$_MA_REFP_SET_POS[1]
3rd reference point	$_MA_REFP_SET_POS[2]
4th reference point	$_MA_REFP_SET_POS[3]

Note

Further details of the points that were considered in the programming of G30 can be found in Section "Reference point approach with intermediate point (G28)". Function G30 is implemented with cycle 330.spf.

Motion commands 3

3.1 The coordinate system

The position of a tool is defined uniquely by its coordinates in the coordinate system. These coordinates are defined through axis positions. If, for instance, the three involved Axes are denoted by X, Y and Z, the coordinates are specified as follows:

X... Y... Z...

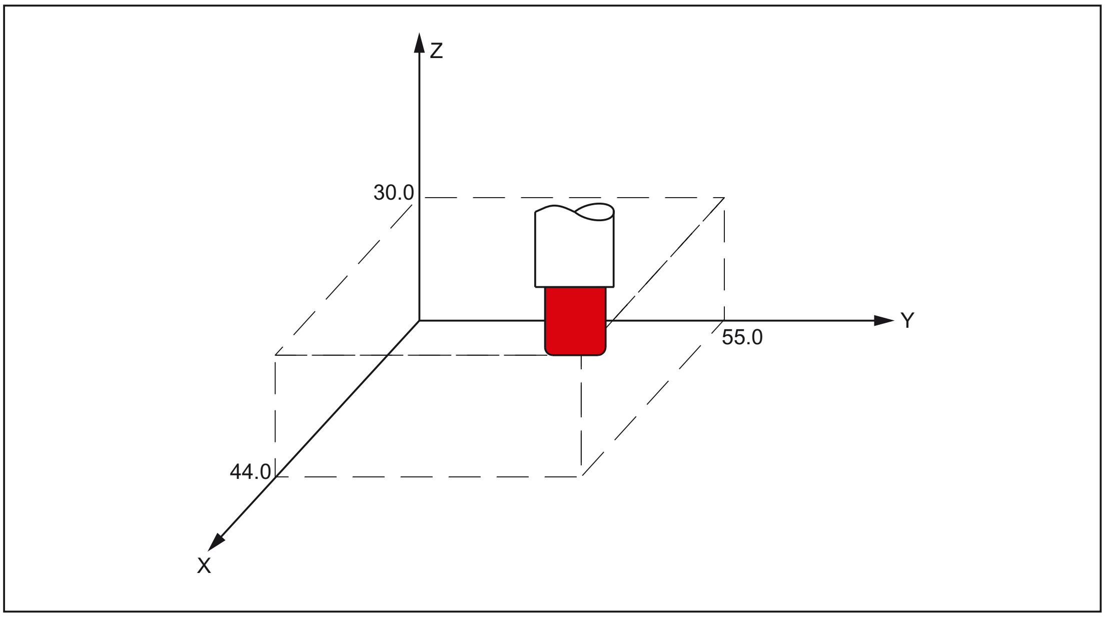


Figure 3-1 Tool positions specified with X... Y... Z...

The following coordinate systems are used to specify the coordinates:

1. Machine coordinate systems (G53)
2. Workpiece coordinate system (G92)
3. Local coordinate system (G52)

3.1.1 Machine coordinate systems (G53)

Defining machine coordinate system

The machine zero defines the machine coordinate system MCS. All other reference points refer to the machine zero.

The machine zero is a fixed point on the machine tool which can be referenced by all (derived) measuring systems.

This is not necessary if an absolute measuring system is used.

Format

(G90) G53 X... Y... Z... ;

X, Y, Z: absolute dimension word

Selection of machine coordinate system (G53)

G53 suppresses the programmable and adjustable work offset . Traversing in the machine coordinate system on the basis of G53 are always programmed if the tool is to traverse to a machine-specific position.

Compensation deselection

If MD10760 $MN_G53_TOOLCORR = 0, then the active tool length and tool radius compensation remains active in a block with G53

If MD10760 $MN_G53_TOOLCORR = 1, then the active tool length and tool radius compensations in a block are suppressed with G53.

Reference

MD24004 $MC_CHBFRAME_POWERON_MASK, Bit 0 is used to define whether channel-specific basic frames during Power On are to be reset.

Displacements and rotations are set to 0, scalings to 1.

Mirroring is switched off.

Value = 0: Basic frame is retained at Power ON

Value = 1: Basic frame is reset at Power On.

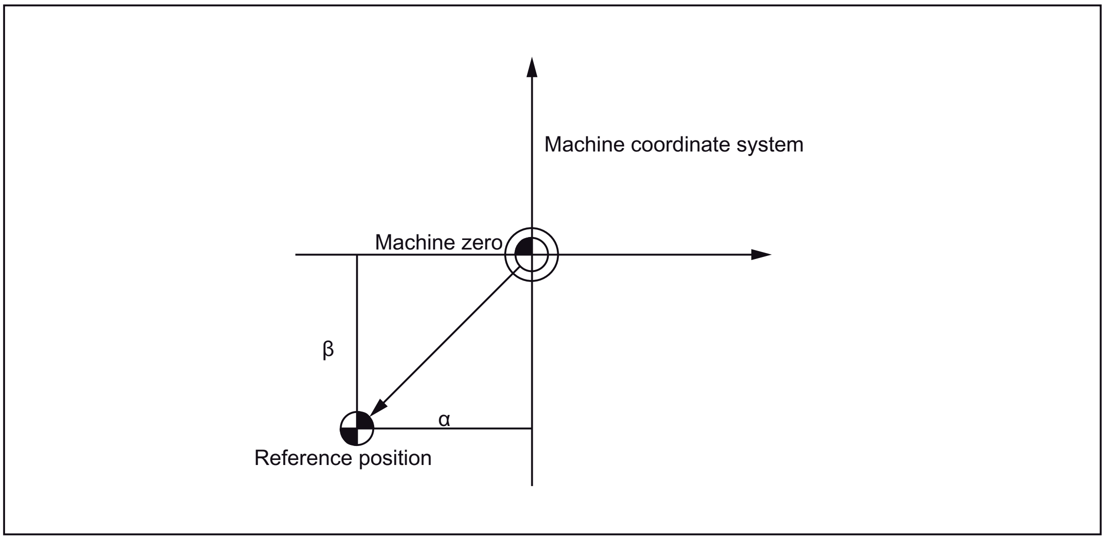


Figure 3-2 Reference

3.1.2 Workpiece coordinate system (G92)

Before machining, you must create a coordinate system for the workpiece, the so-called work piece coordinate system. This section describes different methods of setting, selecting and changing a workpiece coordinate system.

Setting a tool coordinate system

The following two methods can be used to set a tool coordinate system:

1. With G92 in the part program
2. manually through the HMI operator panel

Format

(G90) G92 X... Y... Z... ;

The base point traverses to the specified position on outputting an absolute command. The difference between tool tips and the base point is compensated through the tool length compensation; this way the tool tip can traverse to the target position in any case.

3.1.3 Resetting the tool coordinate system (G92.1)

With G92.1 X.. (G Code System A with G50.3 P0) one can reset a shifted coordinate system before the shift. The tool coordinate system is reset to the coordinate system that is defined by the active adjustable work offsets (G54-G59). The tool coordinate system is set to the reference position if no adjustable work offset is active. G92.1 resets shifts carried out through G92 or G52. However, only the axes that are 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
```

3.1.4 Selection of a workpiece coordinate system

As mentioned above, the user can select one of the already set workpiece coordinate systems.

1. G92

   Absolute commands function in connection with a workpiece coordinate system only if a workpiece coordinate system was selected earlier.

2. Selection of a workpiece coordinate system from a selection of specified workpiece coordinate systems via the HMI operator panel

   A workpiece coordinate system can be selected by specifying a G function in the area G54 to G59 and G54 P{1...100}.

   Workpiece coordinate systems are setup after the reference point approach after Power On. The closed position of the coordinate system is G54.

3.1.5 Writing work offset/tool offsets (G10)

The workpiece coordinate systems defined through G54 to G59 or G54 P{1 ... 93} can be changed with the following two processes.

1. Data inputting at HMI operator panel
2. with the program commands G10 or G92 (setting actual value, spindle speed limitation)

Format

Modified by G10:

G10 L2 Pp X... Y... Z... ;

p=0:	External workpiece work offset
p=1 to 6:	The value of the workpiece work offset corresponds to the workpiece coordinate system G54 to G59 (1 = G54 to 6 = G59)
X, Y, Z:	Workpiece work offset for each axis during an absolute command (G90). Value that must be added during an incremental command (G91) for each axis to the specified workpiece work offset.

G10 L20 Pp X... Y... Z... ;

p=1 to 93:	The value of the workpiece work offset corresponds to the workpiece coordinate system G54 P1 ... P93. The number of work offsets (1 to 93) can be set through MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES or MD28080 $MC_MM_NUM_USER_FRAMES.
X, Y, Z:	Workpiece work offset for each axis during an absolute command (G90). Value that must be added during an incremental command (G91) for each axis to the specified workpiece work offset.

Modified by G92:

G92 X... Y... Z... ;

Explanations

Modified by G10:

G10 can be used to change each workpiece coordinate system individually. If the work offset with G10 is to be written only when the G10 block is executed on the machine (main run block), then MD20734 $MC_EXTERN_FUNCTION_MASK, Bit 13 must be set. An internal STPPRE is executed in that case with G10. The machine data bits affect all G10 commands in the ISO Dialect T and ISO Dialect M.

Modified by G92:

By specifying G92 X... Y... Z..., a workpiece coordinate system that was selected earlier with a G command G54 to G59 or G54 P{1 ...93}, can be shifted and thus a new workpiece coordinate system can be set. If X, Y and Z are programmed incrementally, the workpiece coordinate system is defined in such a way that the current tool position matches the total of the specified incremental value and the coordinates of the previous tool position (shift of coordinate system). Finally, the value of the coordinate system shift is added to each individual value of the workpiece work offset. To put it another way: All workpiece coordinate systems are shifted systematically by the same value.

Example

The tool in operation with G54 is positioned at (190, 150), and the workpiece coordinate system 1 (X' - Y') is created each time in G92X90Y90 with a shift of Vector A.

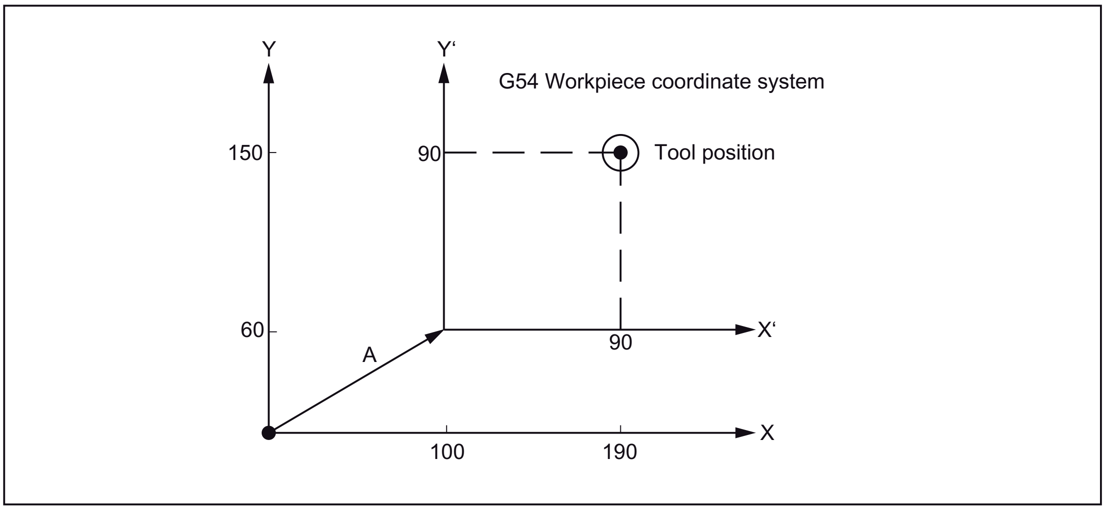


Figure 3-3 Example of setting coordinates

3.1.6 Local coordinate system (G52)

For programming simplification, a type of workpiece coordinate system can be setup to create a program in the workpiece coordinate system. This part coordination system is also called local coordinate system.

Format

G52 X... Y... Z... ; Setting the local coordinate system

G52 X0 Y0 Z0 ; Deselection of the local coordinate system

X, Y, Z: Origin of the local coordinate system

Explanations

G52 can be used to program work offsets for all path and positioning axes in the direction of the specified axis. This way one can work with changing zero points, e.g. during repetitive machining operations at different workpiece positions.

G52 X... Y... Z... is a work offset around the offset values programmed in the relevant specified axis directions. The last specified adjustable work offset (G54 to G59, G54 P1 - P93) serves as reference.

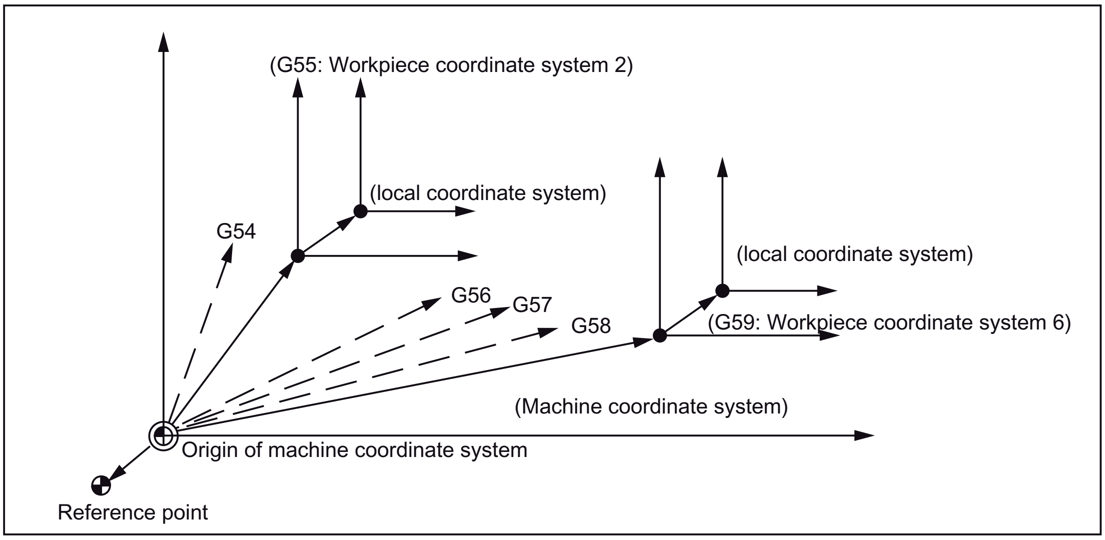

(G55: Workpiece coordinate system 2)
(local coordinate system)
G54
(local coordinate system)
G56
G57
G58
(G59: Workpiece coordinate system 6)
(Machine coordinate system)
Origin of machine coordinate system
Reference point

Figure 3-4 Setting the local coordinate system

3.1.7 Selection of the plane (G17, G18, G19)

The selection of the plane in which the circular interpolation, tool radius compensation and rotation of the coordinate system took place is done by specifying the following G functions.

Table 3- 1 G functions for selecting the plane

G function	Function	G group
G17	X-Y plane	02
G18	Z-X plane	02
G19	Y-Z plane	02

The plane is defined as described below (with the help of the example of Plane X-Y):

The horizontal axis in the first quadrant is the Axis +X, and the vertical axis in the same quadrant is Y-.

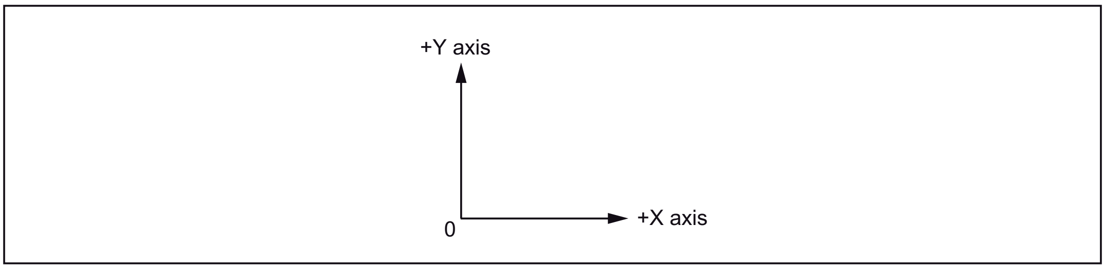


Figure 3-5 Selection of plane

- The Plane X-Y (G17) is selected automatically after activating the control system.
- The command for moving an individual axis can be specified independently of the plane selection by G17, G18 or G19. Thus for instance, the Z axis can be shifted by specifying "G17 Z ....;".
- The plane in which the tool radius compensation is executed with G41 or G42 is defined by specifying G17, G18 or G19.

3.1.8 Parallel axes (G17, G18, G19)

An axis located parallel to one of the three main axes of the coordinate system can be activated by using the function G17 (G18, G19) <Axis name>.

The three main axes are, e.g., X, Y and Z.

Example

G17 U0 Y0

The parallel axis U is activated when the X axis in the G17 plane is replaced.

Explanations

- An associated parallel axis can be defined for each geometry axis with machine data $MC_EXTERN_PARALLEL_GEOAX[ ].
- Only geometry axes from a plane defined with (G17, G18, G19) can be replaced.
- On replacing the axes, normally all shifts (frames) - with the exception of the handwheel and external shifts, the working area limitation and the protection areas - are deleted. The following machine data is to be set to prevent the values from being deleted:

  Shifts (frames)
  $MN_FRAME_GEOAX_CHANGE_MODE

  Protection areas
  $MC_PROTAREA_GEOAX_CHANGE_MODE

  Working area limitation
  $MN_WALIM_GEOAX_CHANGE_MODE
- Details are available in the machine data description.
- Alarm 12726 "Impermissible plane selection with parallel axes" is output if a main axis is programmed along with the associated parallel axis with a command for selecting the plane.

3.1.9 Rotation of the coordinate system (G68, G69)

Properties of G68 and G69

A coordinate system can be rotated with the following G functions.

Table 3- 2 G functions for rotating a coordinate system

G function	Function	G group
G68	Rotation of the coordinate system	16
G69	Deselection of Rotation of the coordinate system	16

G68 and G69 are modal G functions of the G group 16. G69 is set automatically on activating the control system and resetting the NC.

The blocks containing G68 and G69 should not contain any other G functions.

The rotation of the coordinate system is called with G68 and deselected with G69.

Format

G68 X_ Y_ R_ ;

X_, Y_ :

Absolute coordinate values of the rotation center. The actual position is accepted as the rotation center if these are omitted.

R_ :

Angle of rotation as a function of G90/G91 absolute or incremental. If R is not specified, the value of the channel-specific setting from the setting data 42150 $SC_DEFAULT_ROT_FACTOR_R is used as angle of rotation.

- By specifying G17 (or G18, G19) G68 X... Y... R... ; " the commands specified in the following blocks are rotated by the angle specified with R around the point (X, Y). The angle of rotation can be specified in units of 0.001 degree.

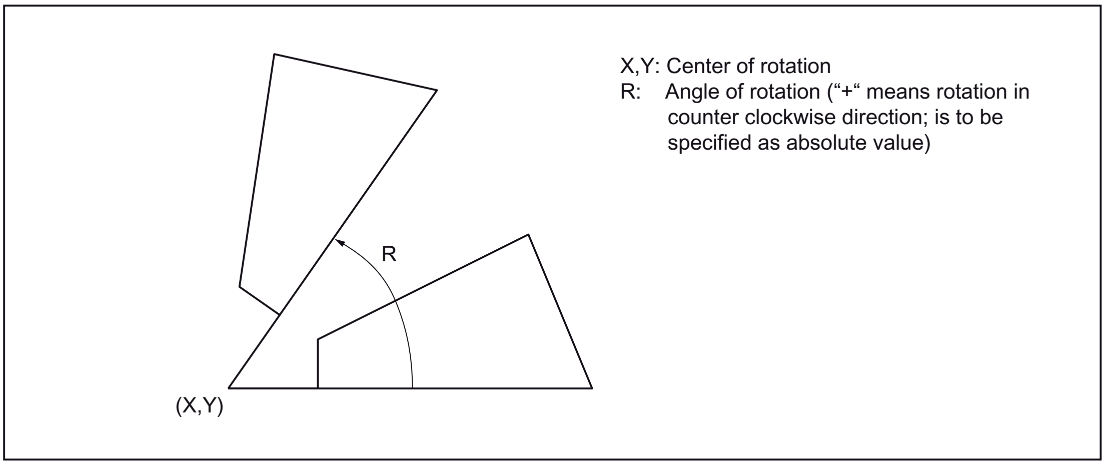

X,Y: Center of rotation
R: Angle of rotation (“+“ means rotation in counter clockwise direction; is to be specified as absolute value)
R
(X,Y)

Figure 3-6 Rotation of a coordinate system

- The deselection of the coordinate system rotation takes place through G69.
- G68 is executed in the plane that was selected through G68. The 4th and 5th axes must be linear axes.

  G17: X-Y plane

  G18: Z-X plane

  G19: Y-Z plane

Additions to the commands for rotating the coordinate systems

- To rotate a coordinate system, MD28081 $MC_MM_NUM_BASE_FRAMES must be set to a value of >= 3.
- If "X" and "Y" are omitted, the current position is used as the rotation center for the coordinate rotation.
- The positional data for the rotation of a coordinate system are specified in the rotated coordinate system.
- If you program a change of plane (G17 to G19) after a rotation, the angles of rotation programmed for the axes are retained and continue to apply in the new working plane. It is therefore advisable to deactivate the rotation before a change of plane.

3.1.10 3D rotation G68/G69

The G code G68 is extended for 3D rotation.

Format

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

X.. Y.. Z..: Coordinates of the pivot point related to the current workpiece zero. If no coordinate is programmed, the pivot point lies in the workpiece zero. The value is always interpreted as absolute. The coordinates of the pivot point act as a work offset. G90/G91 in the block does not affect the G68 command.

I.. J.. K..: Vector in pivot point. The coordinate system is rotated around this vector at angle R.

R..: Angle of rotation. The angle of rotation is always absolute. If no angle is programmed, the angle from the setting data 42150 $SA_DEFAULT_ROT_FACTOR_R is active. G68 must be alone in the block.

The 2D or 3D rotation differentiation takes place only through the programming of the vector I, J, K. If there is no vector in the block, G68 2DRot is selected. If there is a vector in the block, G68 3DRot is selected.

If a vector is programmed with the length 0 (I0, Y0, K0), the Alarm 12560 "Programmed value outside the permissible limits" is triggered.

Two rotations can be switched one after the other with G68. If so far no G68 is active in a block containing G68, then the rotation is written to the channel-specific basic frame 2. If G68 is already active, the rotation is written to the channel-specific basic frame 3. Thus, both rotations follow one another.

The 3D rotation is ended with G69. If two rotations are active, both are deselected with G69. G69 must not be alone in the block.

3.2 Defining the input modes of the coordinate values

3.2.1 Absolute/incremental dimensioning (G90, G91)

Whether the dimensions after an axis address should be absolute or relative (incremental) is specified with these G commands.

Properties of G90, G91

Table 3- 3 G commands for defining the absolute/incremental dimensioning

G command	Function	G group
G90	Absolute dimensioning	03
G91	Incremental dimensioning	03

- G90 and G91 are modal G functions of the G group 03. If G90 and G91 are programmed in the same block, the last G function in the block is effective.
- The closed position of G90 or G91 is set in machine data MD20154 $MC_EXTERN_GCODE_RESET_VALUES[2].

Format

- The programmed values are interpreted as absolute axis positions for all axis positions programmed according to G90, e.g. X, Y, Z.
- The programmed values are interpreted as incremental axis positions for all axis positions programmed according to G91, e.g. X, Y, Z.

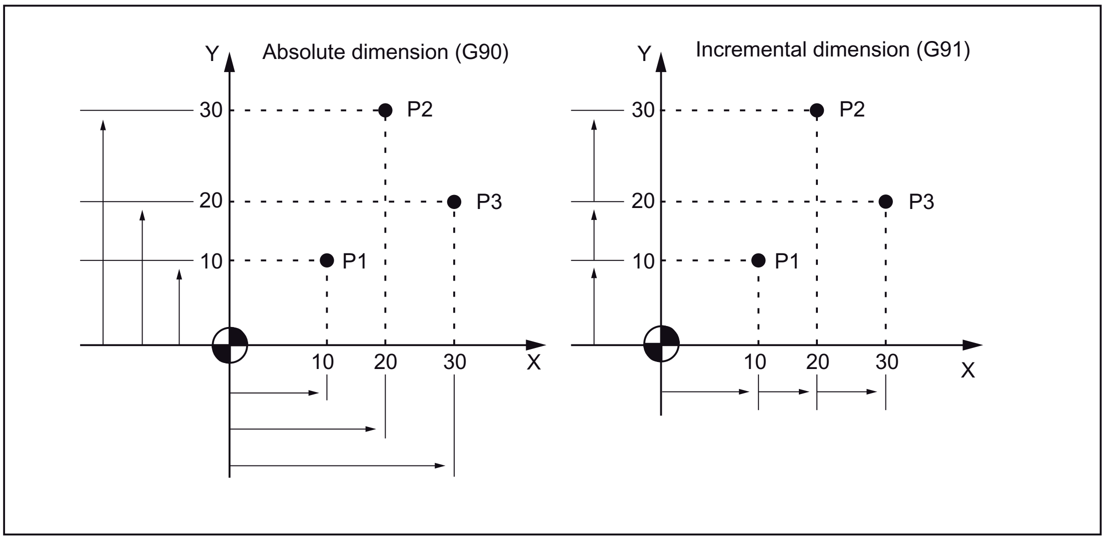


Figure 3-7 Absolute and incremental dimensioning (G90, G91)

3.2.2 Inch/metric input (G20, G21)

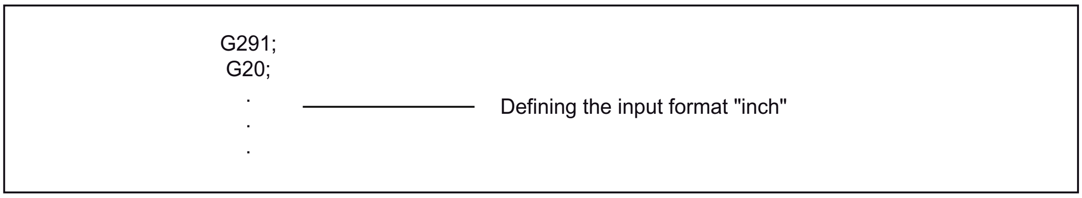
Workpiece-related axes can be programmed in metric or inch dimensions alternately, depending on the dimensioning in the production drawing. The input unit is selected with the following G functions.

Table 3- 4 G command for selecting the unit of measurement

G command	Function	G group
G20	Input in "inch"	06
G21	Input in "mm"	06

Format

G20 and G21 are always to be programmed at the start of the block and should not exist along with other commands in a block. The following values are processed in the selected unit of measurement while executing the G function for selecting the unit of measurement: All the following programs, offset values, certain parameters as well as certain manual operation and readout data.



```
G291;
G20;
  .
  .
  .
```

Figure 3-8 Programming example

Additions to the commands for defining the unit of measurement

- The closed position is defined via the machine data MD20154 $MC_EXTERN_GCODE_RESET_VALUES[5].
- During changeover, the values of the work offsets are changed completely.
- If the unit of measurement is changed over during program execution, the following must be executed in advance:

  While using a workpiece coordinate system (G54 to G59), this is to be traced back to the basic coordinate system.

  All tool offsets are to be deactivated (G41 to G48).
- The following is to be done after switching the measuring system from G20 to G21:

  G92 must be executed before specifying the traversing commands for the axes (to setup the coordinate system).
- G20 and G21 are not used to switch the hand wheel- and incremental weighting. This takes place through the PLC program. The machine data responsible for this is called $MA_JOG_INCR_WEIGHT.

3.2.3 Scaling (G50, G51)

Properties of G50, G51

The form defined by a part program can be enlarged or reduced according to the required scale. The desired scaling can be selected and deselected via the following functions.

Table 3- 5 G functions for selecting the scale

G command	Function	G group
G50	Scaling OFF	11
G51	Scaling ON	11

The selection for scaling and mirroring takes place with G51. A distinction is made between two options in scaling:

- Axial scaling with the parameters I, J, K

  If I, J, K is not programmed in the G51 block, the relevant default value from the setting data 43120 $A_DEFAULT_SCALE_FACTOR_AXIS is effective.

  Negative axial scaling factors lead additionally to mirroring.

- Scaling in all axes with the scaling factor P

  If P is not written in the block G51, the default value from the setting data is effective.

  Negative P values are not possible.

Format

There are two different types of scaling.

Scaling along all axes with the same scaling factor

G51 X... Y... Z... P... ; Start scaling

G50; Deselection of scaling

X, Y, Z: Center coordinate value for the scaling (absolute command)

P: Scaling factor

Scaling along each individual axis with different scaling factors

G51 X... Y... Z... I... J... K... ; Start scaling
G50; Deselection of scaling

X, Y, Z: Reference point of scaling (absolute command)
I, J, K: Scaling factor for the X-, Y- and Z-axis

The type of the scaling factor depends on MD22914 $MC_AXES_SCALE_ENABLE.

$MC_AXES_SCALE_ENABLE = 0:
The scaling factor is specified with "P". If "I,J,K" is programmed in this setting, the setting data 42140 $SC_DEFAULT_SCALE_FACTOR_P is used for the scaling factor.

$MC_AXES_SCALE_ENABLE = 1:
The scaling factor is specified with "I,J,K". If only "P" is programmed in this MD setting, the setting data 43120 $SA_DEFAULT_SCALE_FACTOR_AXIS is used for the scaling factors.

Weighting of scaling factors

The scaling factors are multiplied either with 0.001 or 0.00001. The factors are selected with MD22910 $MC_WEIGHTING_FACTOR_FOR_SCALE=0, scaling factor 0.001, $MC_WEIGHTING_FACTOR_FOR_SCALE=1, scaling factor 0.00001.

The workpiece zero is always the reference point for the scaling. A reference point cannot be programmed.

Programmable mirroring (negative scaling)

A mirror image can be created with a negative value of the axial scaling factor.

To do this, MD22914 $MC_AXES_SCALE_ENABLE = 1 must be active. If I, J or. K is omitted from the blocks with G51, the values preset in the setting data 43120 $SA_DEFAULT_SCALE_FACTOR_AXIS are activated.

Example

```
_N_0512_MPF                       ;(Part program)
N01 G291
N10 G17 G90 G00 X0 Y0             ;Start position for the approach motion
N30 G90 G01 G94 F6000
N32 M98 P0513                     ;1) Contour programmed as in the subroutine
N34 G51 X0. Y0. I-1000 J1000      ;2) Contour, mirrored on X
N36 M98 P0513
N38 G51 X0. Y0. I-1000 J-1000     ;3) Contour, mirrored on X and Y
N40 M98 P0513
N42 G51 X0. Y0. I1000 J-1000      ;4) Contour, mirrored on Y
N44 M98 P0513
N46 G50                           ;Deselection of scaling and mirroring
N50 G00 X0 Y0
N60 M30

_N_0513_MPF                       ;(Subroutine of 00512)
N01 G291
N10 G90 X10. Y10.
N20 X50
N30 Y50
N40 X10. Y10.
N50 M99
```

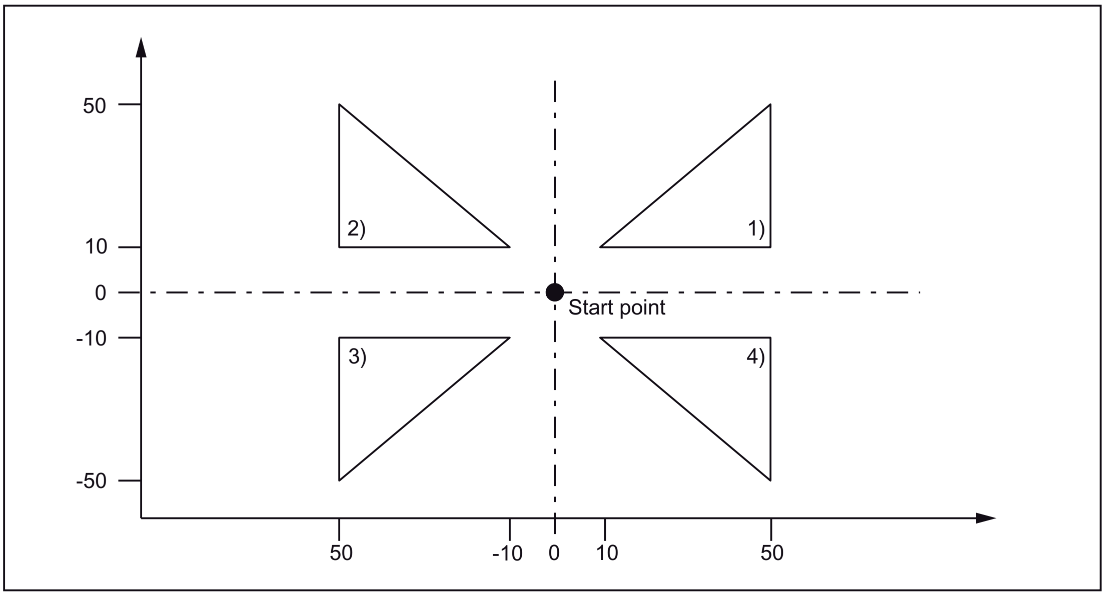


Figure 3-9 Scaling for each axis and programmable mirroring

Tool offset

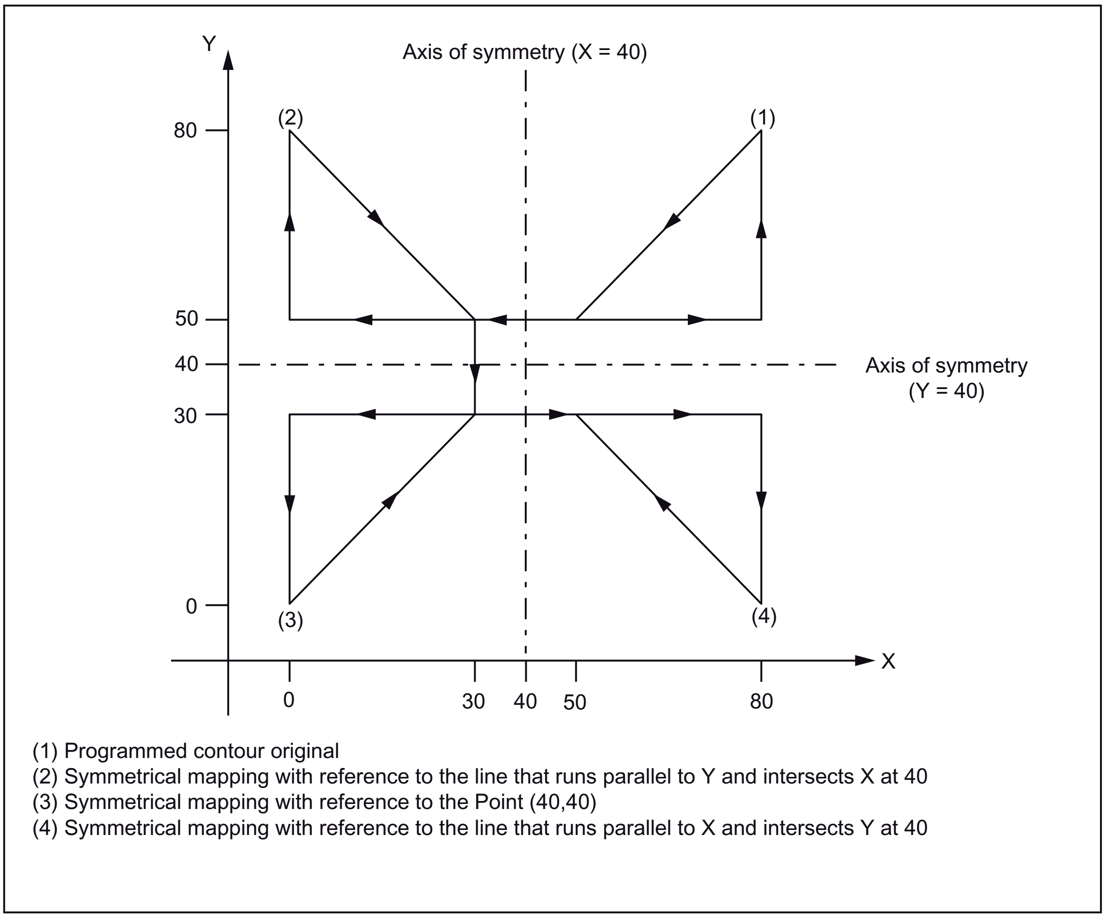
This scaling is not valid for cutter radius compensations, tool length compensations and tool offset values.

Commands for reference point approach and for changing the coordinate system

The G27, G28 and G30 functions as well as commands related to the coordinate system (G52 to G59, G92), should not be used when scaling is active.

3.2.4 Programmable mirroring (G50.1, G51.1)

G51.1 can be used to mirror workpiece shapes on coordinate axes. All programmed traversing movements are then executed as mirrored.



(1) Programmed contour original
(2) Symmetrical mapping with reference to the line that runs parallel to Y and intersects X at 40
(3) Symmetrical mapping with reference to the Point (40,40)
(4) Symmetrical mapping with reference to the line that runs parallel to X and intersects Y at 40

Figure 3-10 Programmable Mirroring

Format

X, Y, Z: Positions and mirroring axis

G51.1: Command for activating the mirroring

Mirroring takes place on a mirroring axis which is parallel to X, Y or Z and whose position is programmed with X, Y or Z. G51.1 X0 is used to mirror on the X axis, G51.1 X10 is used to mirror on a mirroring axis that runs 10 mm parallel to the X axis.

Example

```
N1000 G51.1 X... Y... Z...    ; Activate mirroring
...                           ; All the axis positions mirrored in the following
                              ; blocks are mirrored at the mirroring axis programmed
                              ; in N1000
...                           ;
...                           ;
...                           ;
G50.1 X... Y... Z..           ; Deselection of programmable mirroring
```

Required machine data

G51.1 uses the channel-specific basic frame [1]. Hence, MD28081 $MC_MM_NUM_BASE_FRAMES > = 2 is to be set.

Mirroring with reference to a single axis in a specified plane

The following commands can change if the mirroring is used on one of the axes in the specified plane as described below:

Table 3- 6 Individual axes in specified plane

Command	Explanation
Circular interpolation	G02 and G03 are exchanged mutually
Cutter radius compensation	G41 and G42 are exchanged with each other
Coordinate rotation	The "clockwise" (CW) and "counter-clockwise" (CCW) directions of rotation are exchanged mutually.

Commands for reference point approach and for changing the coordinate system

The G27, G28 and G30 functions as well as commands related to the coordinate system (G52 to G59, G92, etc), should not be used when mirroring is active.

3.3 Time-controlled commands

3.3.1 Dwell time (G04)

One can use G04 to interrupt workpiece machining between two NC blocks for a programmed time/number of spindle revolutions, e.g. for backing off.

One can set with MD20734 $MC_EXTERN_FUNCTION_MASK, whether the dwell time for Bit 2 is to be interpreted as time (s or ms) or alternatively as spindle revolutions. If $MC_EXTERN_FUNCTION_MASK, Bit 2=1 is set, the dwell time is interpreted in seconds if G94 is active; it is specified in spindle revolutions (R) if G95 is selected.

Format

G04 X_; or G04 P_;

X_: Time display (commas possible)

P_: Time display (commas not possible)

- The dwell time (G04 ..) must be programmed alone in a block.

If the values of X and U are programmed in the standard notation (without decimal point), they are converted to internal units, depending on IS B, IS C (for input resolution, see Chapter "Decimal point programming"). P is always interpreted in internal units.

```
N5 G95 G04 X1000
```

Standard notation: 1000*0.001 = 1 Spindle revolution

Calculator notation: 1000 spindle revolutions

3.4 Tool offset functions

3.4.1 Tool offset data memory

The Siemens tool data memory must be used, as programs in the Siemens Mode and in the ISO Direct Mode must run alternately on the control system. Hence, length, geometry and wear exist in each tool offset data memory. In the Siemens mode, the offset data memory is addressed with "T" (Tool No.) and "D" (cutting edge no.). abbreviated as T/D No.

In programs that are written in ISO dialect, the tool offset no. is addressed with "D" (radius) or H (length), denoted hereafter as D/H No.

For unique assignment between D and H numbers or the T/D number, one must add the $TC_DPH[t,d] element to the tool data offset memory. The D/H number is input in ISO dialect in this element.

Table 3- 7 Example: Set tool offset data

T	D/cutting edge	ISO_H $TC_DPH	Radius	Length
1	1	10		
1	2	11		
1	3	12		
2	1	13		
2	2	14		
2	3	15		

For an assignment of tool length compensations of the geometry axes that is independent of the plane selection, the setting data $SC_TOOL_LENGTH_CONST must contain the value "17". Length 1 is always assigned to the Z axis in this case.

3.4.2 Tool length compensation (G43, G44, G49)

In tool length compensation, the amount of the specified values in the program stored in the tool offset data memory is added to the Z axis or subtracted from it to undertake a offset of the programmed paths according to the length of the cutting tool.

Commands

While executing the tool length compensation, the addition or subtraction of the tool offset data is determined through the used G function and the direction of offset is determined with the H functions.

G functions used for the tool length compensation

The tool length compensation is called with the following G functions.

Table 3- 8 G functions used for the tool length compensation

G function	Function	G group
G43	Addition	08
G44	Subtraction	08
G49	Deselection	08

- G43 and G44 are modal and remain active till they are deselected through G49. The tool length compensation is deselected with G49. H00 can also be used to deselect the tool length compensation.
- By specifying "G43 (or G44) Z... H... ; " the tool offset amount specified with the H function is added to or subtracted from the specified position of the Z axis, and the Z axis then traverses to the corrected target position, i.e., the target position of the Z axis specified in the program is shifted by the magnitude of the tool offset.
- By specifying "(G01) Z... ; G43 (or G44) H... ; " the Z axis traverses the path that corresponds to the tool offset amount specified via the H function.
- By specifying "G43 (or G44) Z...H...H... ; " the Z axis traverses the path that corresponds to the difference between the previous tool offset amount and the new tool offset amount.

H function for specification of the tool offset direction

The direction of tool offset is determined by the sign of the tool length compensation that is activated by the H function, and the programmed G function.

Table 3- 9 Signs are present before the amount of tool offset and direction of tool offset

	Signs of tool offset amount (H function)	
	positive	negative
G43	Tool offset in positive direction	Tool offset in negative direction
G44	Tool offset in negative direction	Tool offset in positive direction

```
Programming example
H10 ....... Offset -3.0
H11 ....... Offset 4.0
            Position data indicator
            including offset
            (Z axis only)

N101  G92 Z0; 0.000
N102  G90 G00 X1.0 Y2.0; 0.000
N103  G43 Z-20. H10; -23.000
N104  G0I Z-30. FI000; -33.000
N105  G00  Z0H00; 0.000
    .    .
    .    .
    .    .
N201 G00 X-2.0 Y-2.0
N202 G44 Z-30 HI1; -34.000
N203 G01 Z-40 F1000; -44.000
N204 G00 Z0 H00; 0.000
```

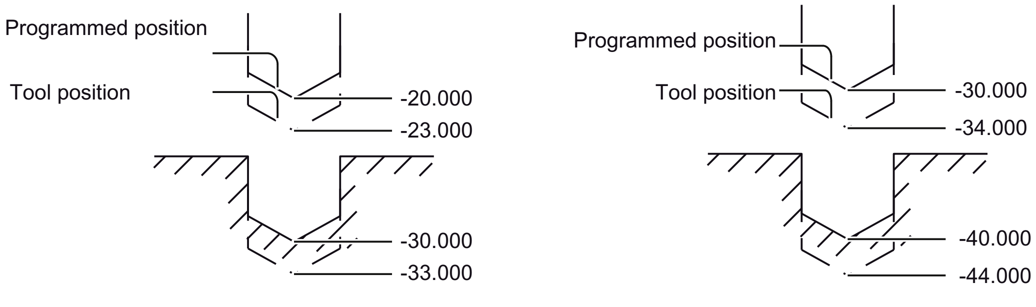


Figure 3-11 Tool position offset

Settings

- The machine data $MC_TOOL_CORR_MOVE_MODE determines whether the tool length compensation is to be undertaken with the selection of the tool offset or only during the programming of an axis motion.

  $MC_CUTTING_EDGE_DEFAULT = 0 defines that initially no tool length compensation is active during a tool change.

  $MC_AUXFU_T_SYNC_TYPE defines whether the output of the T function to the PLC takes place during or after the traversing movement.

  $MC_RESET_MODE_MASK, Bit 6, can be used to define that the currently active tool length compensation will remain active even after a RESET.
- The cutter radius compensation can also be called for an operation with tool length compensation.

Tool length compensation in several axes

Tool length compensation can also be activated for several axes. A display of the resulting tool length compensation is not possible any more in that case.

3.4.3 Cutter radius compensation (G40, G41, G42)

In cutter radius compensation, the programmed tool paths are shifted automatically by the radius of the cutting tool used. The path to be corrected (radius of the cutting tool) can be stored in the tool offset data memory using the NC operator panel. The tool offsets can also be overwritten with the G10 command in the part program; G10 cannot be used to create new tools.

The tool offset data in the program is called by specifying the number of the tool offset data memory with a D function.

Commands

The cutter radius compensation is called with the following G functions.

Table 3- 10 G functions for calling the cutter radius compensation

G function	Function	G group
G40	Deselection of the tool radius compensation	07
G41	Tool radius compensation (tool works in machining direction to the left of the contour)	07
G42	Tool radius compensation (tool works in machining direction to the right of the contour)	07

The tool radius compensation is called by executing G41 or G42 and deselected through G40. The offset direction is determined through the specified G function (G41, G42) and the offset amount is determined through the D function.

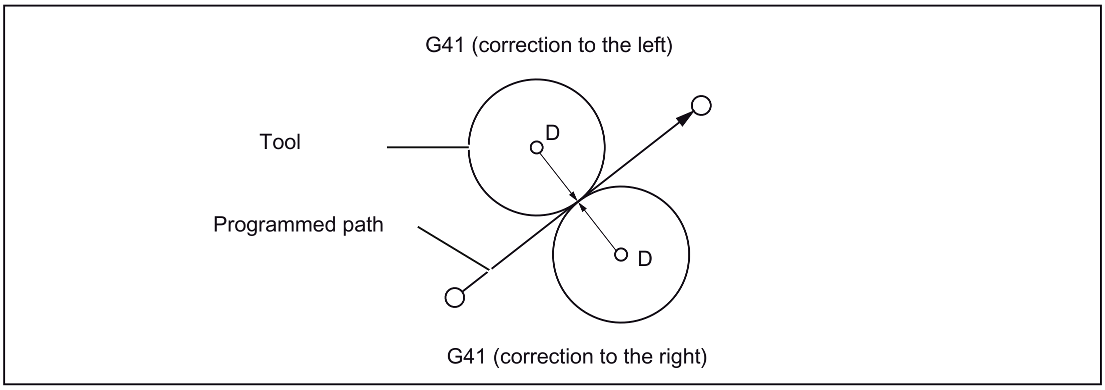


Figure 3-12 Cutter radius compensation

- A negative offset value of the tool radius is equivalent to a change of compensation side (G41, G42). The D function must either be programmed in the same block as G41 or G42 or in a previous block. D00 means tool radius = "0".
- The selection of the plane in which the tool radius is active is done with G17, G18 or G19. The G function used to select the plane is to be programmed in the same block as G41 or G42 or in the block before G41 or G42.

Table 3- 11 G functions for selecting the plane

G function	Function	G group
G17	Selection of plane X-Y	02
G18	Selection of plane Z-X	02
G19	Selection of plane Y-Z	02

- The selected plane should not be changed if the tool offset is selected, otherwise there is an error message.

Activation/deactivation of tool radius compensation

A drive command must be programmed with G0 or G1 if an NC block contains G40, G41 or G42. At least one axis of the selected working plane must be specified in this drive command.

Note

Compensation mode

Compensation mode may only be interrupted by a certain number of consecutive blocks or M functions which do not contain drive commands or positional data in the compensation plane: Standard 3.

Note

Machine manufacturer

The number of successive interruptions blocks or M functions can be set via the machine data 20250 CUTCOM_MAXNUM_DUMMY_BLOCKS (refer to machine manufacturer).

Note

A block with path zero is also taken as interruption!

Changeover between G41 and G42 in operation with cutter radius compensation

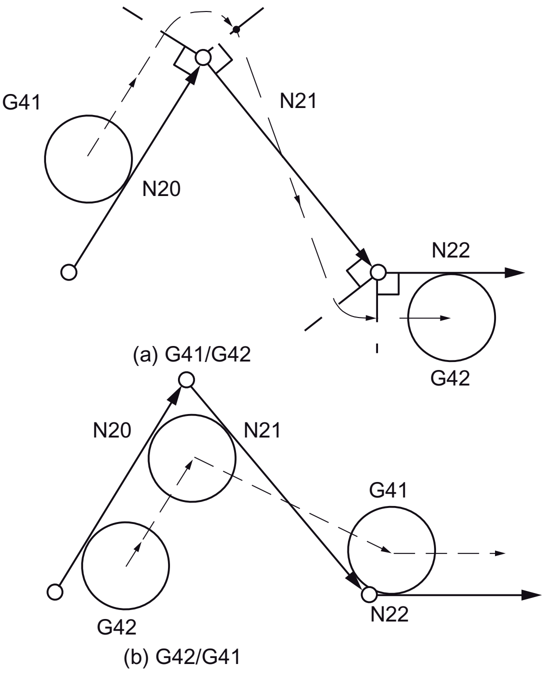
The offset direction (left or right) can be changed over directly without having to leave the compensation mode.

The new offset direction is approached with the next block, through an axis motion.

Programming example

```
N10 G17 G01 F... ;
N11 G41 (G42) D... ;
      .
      .
      .
N20 G01 X... Y... F... ;
N21 G42 (G41) X... Y... ;
N22 X... ;
```

Set for switching the offset direction



Note: If the content of Block N21 is reproduced in two blocks as specified below,
G42 (or G41);
X   Y;
the offset direction is also switched over in the same way.

Figure 3-13 Changeover of the tool offset direction at block start and end of block

Deselection of the tool offset

There are two methods of deselecting the tool offset, which can be set through the setting data 42494 $SC_CUTCOM_ACT_DEACT_CTRL.

1. Method A:

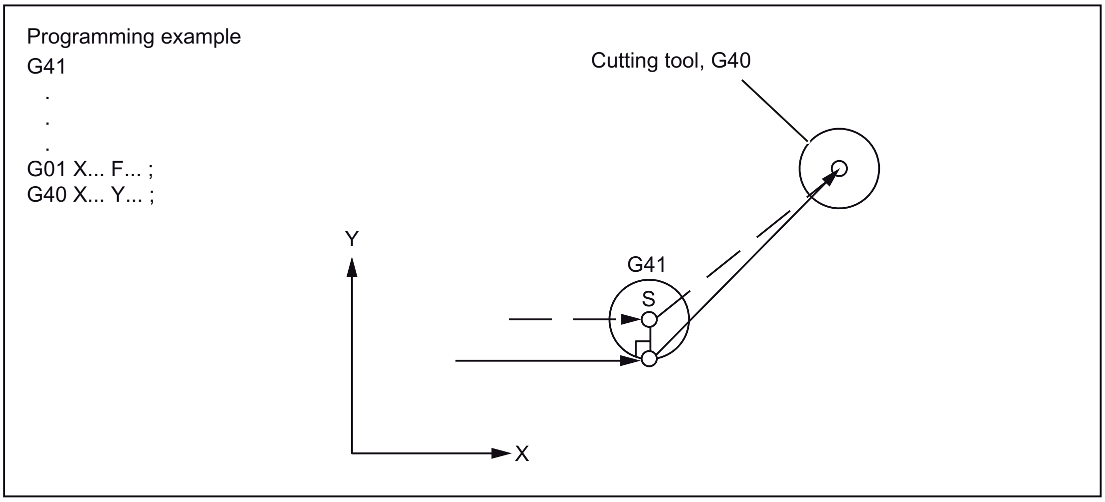
   If G40 is programmed in a block without axis motion, the tool radius compensation is deselected only with the next block through an axis motion.

2. Method B:

   If G40 is programmed in a block without axis motion, the tool radius compensation is deselected immediately. In other words, that linear interpolation (G00 or G01) must be active in the block, because the tool radius compensation can be deselected only with a linear movement. An alarm is triggered if no linear interpolation is active during the selection of the tool radius compensation.

Deselection of the compensation mode at an internal angle (smaller than 180°):

Straight line - straight line

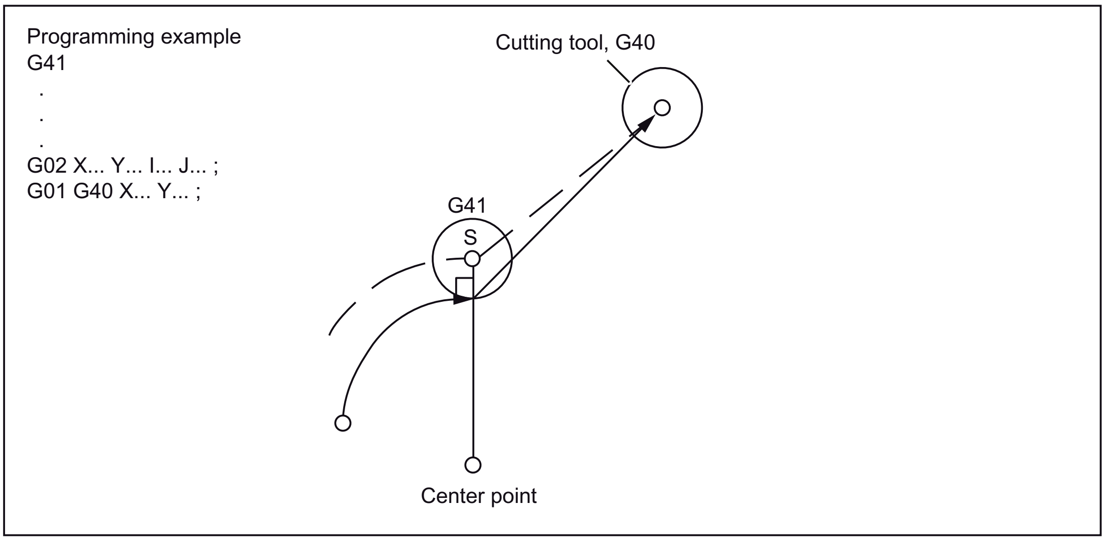


Programming example

```
G41
  .
  .
  .
G01 X... F... ;
G40 X... Y... ;
```

Figure 3-14 Deselection of the compensation mode at an internal angle (straight line - straight line)

Circular arc - straight line



Programming example

```
G41
  .
  .
  .
G02 X... Y... I... J... ;
G01 G40 X... Y... ;
```

Figure 3-15 Deselection of the compensation mode at an internal angle (circular arc - straight line)

3.4.4 Collision detection

Activation via the NC program

Although the "Collision detection" function is available only in the Siemens mode, it can also be used in the ISO dialect mode. Activation and deactivation must be undertaken only in the Siemens mode.

```
G290                    ;Activation of the Siemens mode
CDON                    ;Activation of the detection of bottlenecks
G291                    ;Activation of the ISO dialect mode
...
...
G290                    ;Activation of the Siemens mode
CDOF                    ;Deactivation of the detection of bottlenecks
G291                    ;Activation of the ISO dialect mode
```

Activation by setting machine data

MD20150 $MC_GCODE_RESET_VALUES[22] = 2: CDON (effective modal)

MD20150 $MC_GCODE_RESET_VALUES[22] = 1: CDOF (not effective modal)

Function

With active CDON (Collision Detection ON) and active tool radius compensation, the control system monitors tool paths through look-ahead contour calculation. This Look Ahead function allows possible collisions to be detected in advance and permits the control to actively avoid them.

With deactivated bottleneck detection (CDOF), a search is made in the previous traversing block (at inside corners) for a common point of intersection for the current block; if necessary the search is extended to even earlier blocks. An error message is triggered if no point of intersection is found with this method.

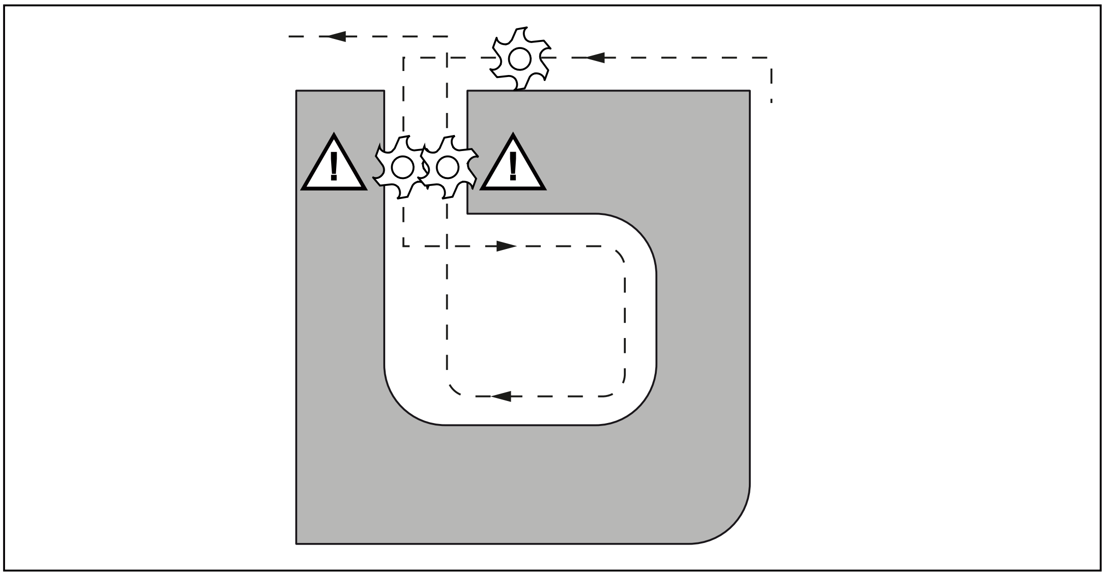


Figure 3-16 Collision detection

CDOF can be used to avoid the faulty detection of bottlenecks, resulting, for example, from missing information that is not available in the NC program.

Note

Machine manufacturer

The number of NC blocks that are included in the monitoring can be set via machine data (see machine manufacturer).

Examples

In the following pages you will find a few examples of critical machining situations that can be detected by the control system and corrected through changes in the tool paths.

To avoid program interruptions, during program validation only the ones that have the biggest radius from among all tools should be selected.

In each of the following examples, a tool with a too large radius was selected for machining the contour.

Detection of bottlenecks

As the selected tool radius for machining this inside contour is too big, the bottlenecks are bypassed. An alarm is output.

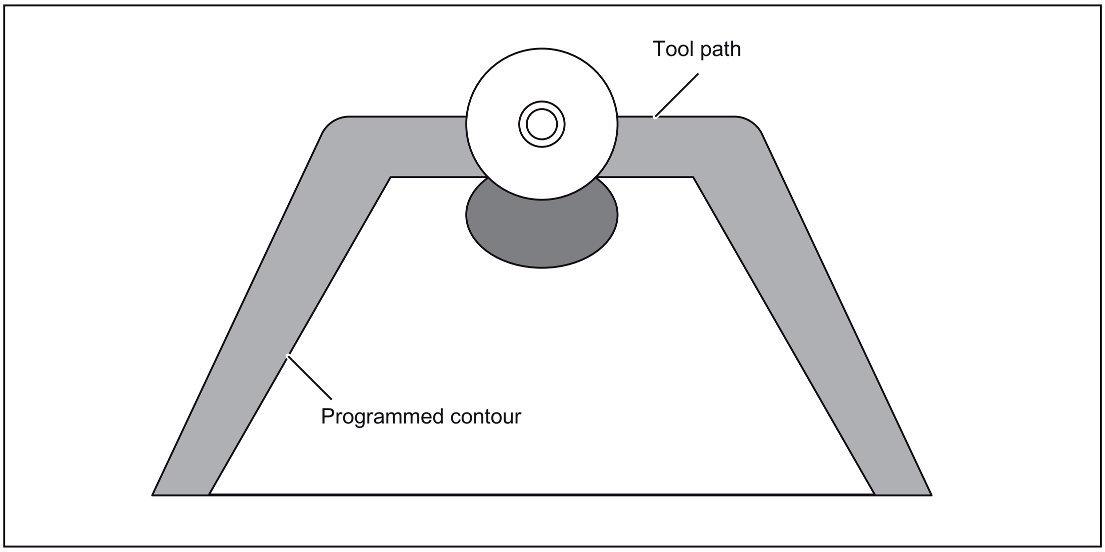


Figure 3-17 Detection of bottlenecks

Contour definition shorter than tool radius

The tool traverses the tool angle on a transition circle and then follows exactly the programmed contour.

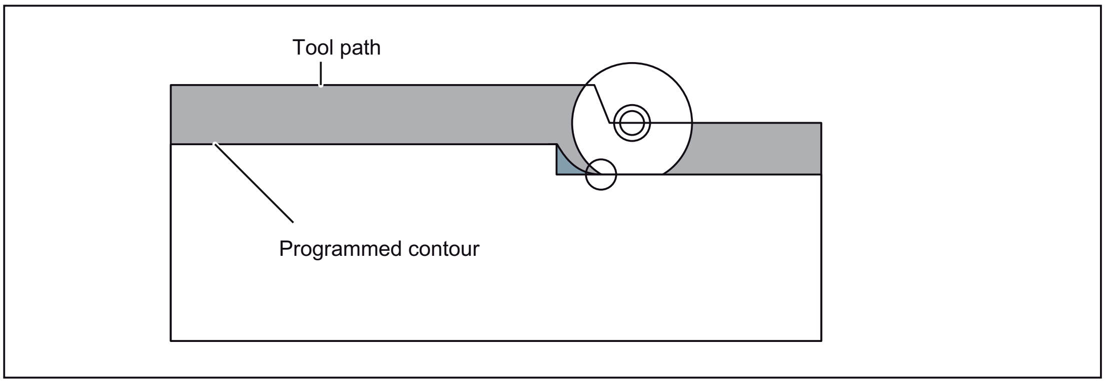


Figure 3-18 Contour definition shorter than tool radius

Tool radius too large for internal machining

In such cases, a machining of the contour takes place only to the extent possible without damaging the contour.

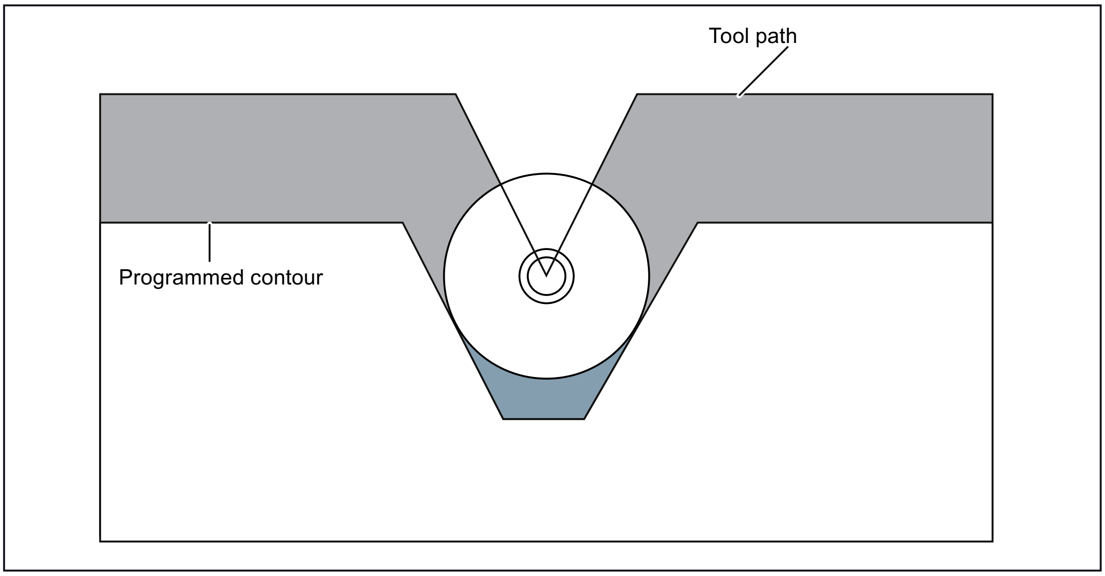


Figure 3-19 Tool radius too large for internal machining

3.5 S-, T-, M- and B functions

3.5.1 Spindle function (S function)

The spindle speed is specified in rpm in Address S. The direction of spindle rotation is selected with M3 and M4. M3 = right direction of spindle rotation, M4 = left direction of spindle rotation. The spindle stops with M5. Details are available in the documentation of your machine manufacturer.

- S commands are modal, i.e., they remain active up to the next S command once they are programmed. The S command is maintained if the spindle is stopped with M05. If M03 or M04 is programmed thereafter without specifying an S command, then the spindle starts at the originally programmed speed.
- If the spindle speed is changed, please pay attention to which gear stage is currently set for the spindle. Details are available in the documentation of your machine manufacturer.
- The lower limit for the S command (S0 or an S command near S0) depends on the drive motor and the drive system of the spindle and is different from machine to machine. Negative values are not permitted for S! Details are available in the documentation of your machine manufacturer.

3.5.2 Tool function

There are different options of command output for the tool function. Details are available in the documentation of your machine manufacturer.

3.5.3 Additional function (M function)

The M functions initiate switching operations, such as "Coolant ON/OFF" and other functions on the machine. Various M functions have already been assigned a fixed functionality by the CNC manufacturer (see the following section).

Programming

M... Possible values: 0 to 9999 9999 (max. INT value), integer

All free M function numbers can be assigned by the machine manufacturer, e.g. for switching functions to control the clamping devices or for switching on/off of further machine functions. See data of the machine manufacturer.

The NC-specific M functions are described below.

M functions to stop operations (M00, M01, M02, M30)

A program stop is triggered with this M function and the machining is interrupted or ended. Whether the spindle is also stopped depends on the specification of the machine manufacturer. Details are available in the documentation of your machine manufacturer.

M00 (program stop)

The machining is stopped in the NC block with M00. One can now, e.g., remove chips, re-measure, etc. A signal is output to the PLC. The program can be continued with NC start.

M01 (optional stop)

M01 can be set via

- HMI/dialog box "Program control" or the
- VDI interface

The program processing of the NC is maintained with M01 only if the corresponding signal of the VDI interface is set or "Program control" was selected in the HMI/dialog box.

M30 or M02 (end of program)

A program is ended with M30 or M02.

Note

A signal is output to the PLC with M00, M01, M02 or M30.

Note

Data on whether spindle is stopped with the commands M00, M01, M02 or M30 or the coolant supply is interrupted is available in the documentation of your machine manufacturer.

3.5.4 M functions of spindle control

Table 3- 12 M functions of spindle control

M function	Function
M19	Positioning the spindle
M29	Changeover of spindle to the axis/open-loop control mode

The spindle is traversed to the spindle position defined in the setting data 43240 $SA_M19_SPOS[spindle number] with M19. The positioning mode is stored in $SA_M19_SPOS.

The M function number for the changeover of the spindle mode (M29) can also be set over a machine data variable. MD20095 $MC_EXTERN_RIGID_TAPPING_N_NR is used to pre-set the M function number. Only the M function numbers that are not used as standard M functions can be assigned. For example, M0, M5, M30, M98, M99 etc are not allowed.

3.5.5 M functions for subroutine calls

Table 3- 13 M functions for subroutine calls

M function	Function
M98	Subprogram call
M99	Subprogram end

In the ISO mode, the spindle is switched to the axis mode with M29.

3.5.6 Macro call via M function

Via M numbers, one can call a subroutine (macro) similar to G65.

The configuration of a maximum of 10 M functions replacements is undertaken via machine data 10814 $MN_EXTERN_M_NO_MAC_CYCLE and machine data 10815 $MN_EXTERN_M_NO_MAC_CYCLE_NAME.

Programming takes place identical to G65. Repetitions can be programmed with the L address.

Restrictions

Only one M function replacement (or only one subroutine call) can be executed per part program line. Conflicts with other subroutine calls are signaled by alarm 12722. There is no further M function replacement in the replaced subroutine.

Otherwise, the same restrictions are valid as in G65.

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

Configuration example

Call of subroutine M101_MAKRO via the M101 M function:

$MN_EXTERN_M_NO_MAC_CYCLE[0] = 101

$MN_EXTERN_M_NO_MAC_CYCLE_NAME[0] = "M101_MAKRO"

Call of subroutine M6_MAKRO via the M6 M function:

$MN_EXTERN_M_NO_MAC_CYCLE[1] = 6

$MN_EXTERN_M_NO_MAC_CYCLE_NAME[1] = "M6_MAKRO"

Programming example for tool change with M function:

```
PROC MAIN
...
N10                M6 X10 V20                      ;Call of M6_MAKRO program
...
N90                M30
PROC M6_MAKRO
...
N0010              R10 = R10 + 11.11
N0020              IF $C_X_PROG == 1 GOTOF N40     ;($C_X_PROG)
N0030              SETAL(61000)                    ;programmed variable not
                                                   ;transferred correctly
N0040              IF $C_V == 20 GTOF N60          ;($C_V)
N0050              SETAL(61001)
N0060              M17
```

3.5.7 M functions

General M functions

The non-specific M functions are defined by the machine manufacturer. A representative example of the use of of general M functions is available under. Details are available in the documentation of your machine manufacturer. If an M command is programmed with an axis motion in the same block, whether the M function is to be executed at the start or end of the block on reaching the axis position depends on the machine data setting of the machine manufacturer. Details are available in the documentation of your machine manufacturer.

Table 3- 14 Other general M functions

M function	Function	Remarks
M08	Coolant ON	These M functions are defined by the machine manufacturer.
M09	Coolant OFF	

Specification of several M functions in one block

A maximum of five M functions can be programmed in on block. Possible combinations of M functions and possible restrictions are specified in the documentation of your machine manufacturer.

Additional auxiliary functions (B functions)

If B is not used as axis identifier, B can be used as extended auxiliary function. B functions are output to the PLC as auxiliary functions (H functions with the address extension H1=).

Example: B1234 is output as H1=1234.

3.6 Controlling the feedrate

3.6.1 Automatic corner override G62

An inside corner with active tool radius compensation is often meaningful to reduce the feedrate.

G62 operates only on internal corners with active tool radius compensation and active continuous-path mode. Only the corners whose internal angle is smaller than MD42526 $SC_CORNER_SLOWDOWN_CRIT. The internal angle is defined from the bend in the contour.

The feedrate is lowered by the factor from the setting data 42524 $SC_CORNER_SLOWDOWN_OVR:

traversed feedrate = F * $SC_CORNER_SLOWDOWN_OVR * feedrate override.

The feedrate override comprises of the feedrate override set in the machine control panel multiplied with the override from synchronized actions.

The feedrate slowdown is started at the distance before the corner registered in the setting data 42520 $SC_CORNER_SLOWDOWN_START. It ends with the distance after the corner (see following Figure) registered in the setting data 42522 $SC_CORNER_SLOWDOWN_END. An appropriate path is used for curved contours.

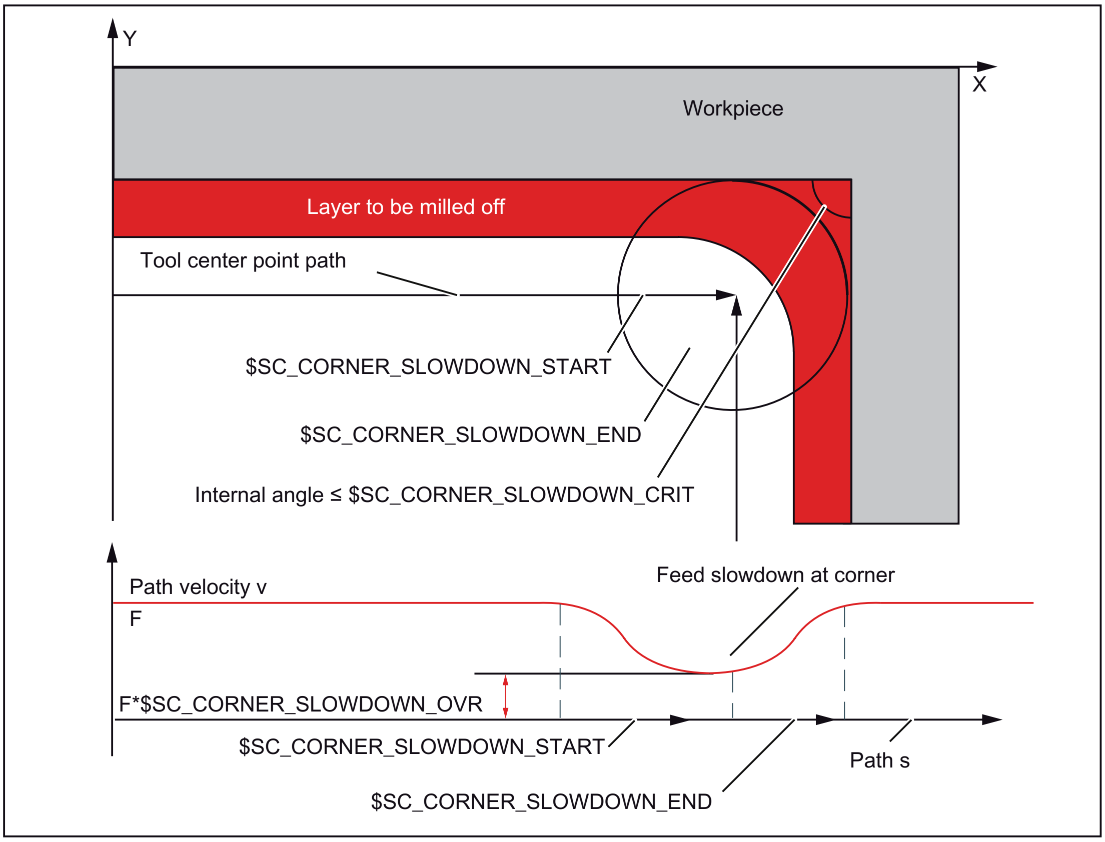


Figure 3-20 Parameter assignment of feedrate reduction G62 with the example of a 90°corner

Configuration

The override value is set over 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 is preassigned with the value 0.

- If $SC_CORNER_SLOWDOWN_CRIT = 0, the corner deceleration is effective only at reversal points.
- If $SC_CORNER_SLOWDOWN_START and $SC_CORNER_SLOWDOWN_END are equal to 0, then the feedrate reduction is approached with the permissible dynamic response.
- If $SC_CORNER_SLOWDOWN_OVR = 0, then a transient stop is inserted.
- $SC_CORNER_SLOWDOWN_CRIT is related to the geometry axes in G62. It defines the maximum internal angle in the current machining plane up to which the corner deceleration is used. G62 is not effective in rapid traverse.

Activating

The function is activated via G62. The G code is active either via the corresponding part program command or as default through MD20150 $MC_GCODE_RESET_VALUES[56].

Example

Tool data in the Siemens mode

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

Setting the setting data in the Siemens mode

```
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.
```

Program in the ISO mode

```
N2010 X00 Y30 G90 T1 D1 G64
N2020 X40 Y0 G62 G41            ; Inner corner of N2030,
                                ; but WRK still selected
N2030 X80 Y30                   ; Inner corner of N2040 127 degree
N2040 Y70                       ; Inner corner of N2050 53 degree
N2050 X40 Y40                   ; Outer corner of N2060
```

```
N2060 X20 Y70                       ; Inner corner of N2070 97 degree
N2070 Y60                           ; Inner corner of N2080 90 degree
N2080 X20 Y20                       ; Outer corner of N2090,
                                    ; irrelevant due to WRK deselection
N2090 X00 Y00 G40 G64               ; Deselect G62 and on continuous-path mode

M30
```

3.6.2 Compressor in the ISO dialect mode

The commands COMPON, COMPCURV, COMPCAD are commands of the Siemens language and they activate a compressor function that combines several linear blocks into one machining section. If this function is activated in the Siemens mode, even linear blocks in the ISO mode can be compressed with this function.

The blocks can at the most consist of the following commands:

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

If a block contains other commands (e.g., auxiliary functions, other G codes, etc.), then compression does not take place.

Value assignments with $x for G, axes and feedrate are possible, just as 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                  ; Auxiliary function in block
N35   X100 S200                 ; Spindle speed in block
```

3.6.3 Exact stop (G09, G61), Continuous-path mode (G64), tapping (G63)

The path feedrate is controlled as specified in the table below.

Table 3- 15 Control of the path feedrate

Identifier	G function	Efficacy of the G functions	Description
Exact stop	G09	effective only in the block in which the relevant G function is programmed	Braking and stop at end of block and position control before transition to the next block
Exact stop	G61	Modal G function; remains effective till it is deselected via G62, G63 or G64.	Braking and stop at end of block and position control before transition to the next block
Continuous-path mode	G64	Modal G function; remains effective till it is deselected via G61, G62 or G63.	No braking at end of block before transition to the next block
Tapping	G63	Modal G function; remains effective till it is deselected via G61, G62 or G64.	No braking at end of block before transition to the next block; feedrate override is not effective

Format

```
G09 X... Y... Z...          ; Exact stop, non-modal
G61                         ; Exact stop, modal
G64                         ; Continuous-path mode
G63                         ; Tapping
```

4 Additional functions

4.1 Program supporting functions

4.1.1 Fixed drilling cycles

The fixed drilling cycles simplify the creation of new programs for the programmer. Frequently occurring machining steps can be executed with a G function; without fixed cycles, several NC blocks must be programmed. Thus fixed drilling cycles shorten the machining program and save memory space.

In the ISO dialect mode, a shell cycle is called which uses the functionality of the Siemens standard cycles. This way, the addresses programmed in the NC block are transferred to the shell cycle via system variables. The shell cycle adjusts this data and calls a Siemens standard cycle.

The fixed drilling cycles are called with the following G functions.

Table 4- 1 Overview of drilling cycles

G function	Drilling (-Z direction)	Machining at drilling base	Retraction (+Z direction)	Applications
G73	Interrupted working feedrate (delay possible at each in-feed)	—	Rapid traverse	High-speed deep-hole drilling
G74	Cutting feedrate	Spindle stop → Spindle revolution after dwelling in the opposite direction	Cutting feedrate → dwell time → Spindle turns in the opposite direction	Left-hand thread boring (in the opposite direction)
G76	Cutting feedrate	Spindle positioning → Withdraw lift-off path	Rapid traverse → Return lift-off path, spindle start	Precision drilling Boring
G80	—	—	—	Deselection
G81	Cutting feedrate	—	Rapid traverse	Drilling, Preboring
G82	Cutting feedrate	Dwell	Rapid traverse	Drilling, countersinking
G83	Interrupted working feedrate	—	Rapid traverse	Deep-hole drilling
G84	Cutting feedrate	Spindle stop→ Spindle start after dwelling in the opposite direction	Cutting feedrate → dwell time → Spindle turns in the opposite direction	Tapping
G85	Cutting feedrate	—	Cutting feedrate	Boring

G function	Drilling (-Z direction)	Machining at drilling base	Retraction (+Z direction)	Applications
G86	Cutting feedrate	Spindle stop	Rapid traverse → spindle start	Boring
G87	Spindle positioning → Withdraw lift-off path → Rapid traverse → Return lift-off path → Spindle run right → Cutting feedrate	Spindle positioning after dwelling → Withdraw lift-off path	Rapid traverse → Return lift-off path → Spindle start	Boring
G89	Cutting feedrate	Dwell	Cutting feedrate	Boring

Explanations

On using fixed cycles, the sequence of operation in general is always as described below:

- 1. Working cycle

  Positioning in X-Y plane with cutting feedrate or rapid traverse rate
- 2. Working cycle

  Rapid traverse movement to plane R
- 3. Working cycle

  Machining up to drilling depth Z
- 4. Working cycle

  Machining at drilling base
- 5. Working cycle

  Return up to plane R with cutting feedrate or rapid traverse rate
- 6. Working cycle

  Rapid retraction with rapid traverse rate to positioning plane X-Y

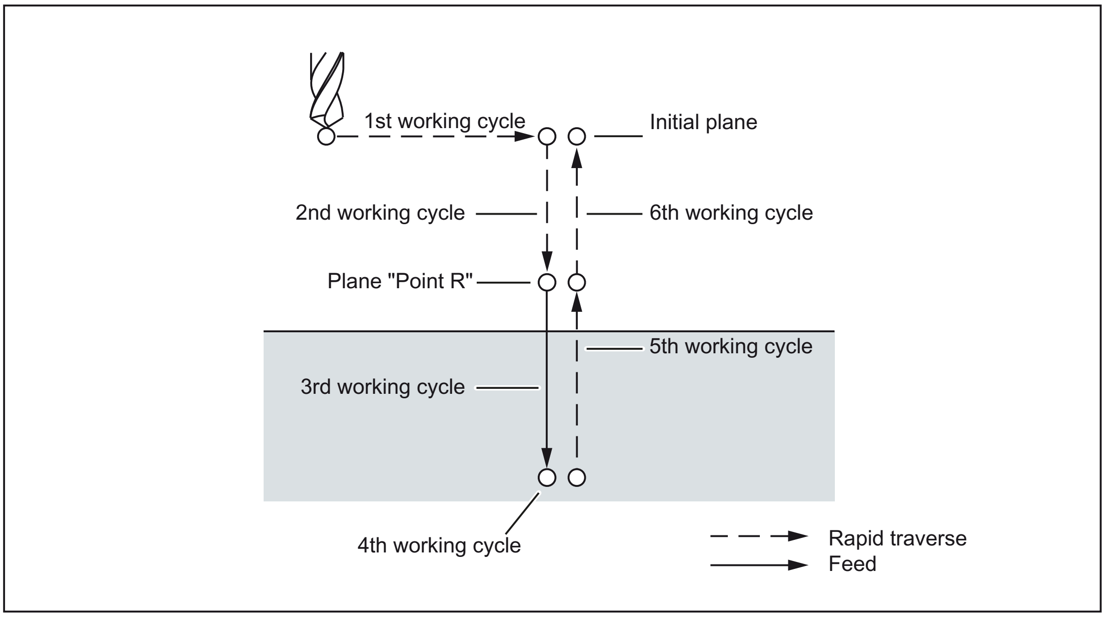


Figure 4-1 Sequence of operations in the drilling cycle

If the term "drill" is used in this Chapter, it refers only to the working cycle that are executed with the help of fixed cycles, even though naturally there are fixed cycles for tapping, boring or drilling cycles too.

Definition of the current plane

In drilling cycles, one generally assumes that the current coordinate system in which the machining operation is to be executed, is defined through the selection of plane G17, G18 or G19 and activation of a programmable work offset. Drilling axis is then always the application of this coordinate system.

Before calling the cycle, one must always select a tool length compensation. It is always effective perpendicular to the selected plane and remains active even beyond the end of the cycle.

Table 4- 2 Positioning plane and drilling axis

G function	Positioning plane	Drilling axis
G17	Xp-Yp plane	Zp
G18	Zp-Xp plane	Yp
G19	Yp-Zp plane	Xp

Xp: X axis or an axis parallel to the X axis

Yp: Y axis or an axis parallel to the Y axis

Zp: Z axis or an axis parallel to the Z axis

Note

Whether the Z axis should always be used as the drilling axis can be defined with MD55800 $SCS_ISO_M_DRILLING_AXES_IS_Z. The Z axis is then always the drilling axis, if $SCS_ISO_M_DRILLING_AXES_IS_Z is equal to "1".

Execution of a fixed cycle

The following is necessary to execute a fixed cycle:

1. Cycle call

   G73, 74, 76, 81 to 89

   as a function of the desired machining

2. Data format G90/91

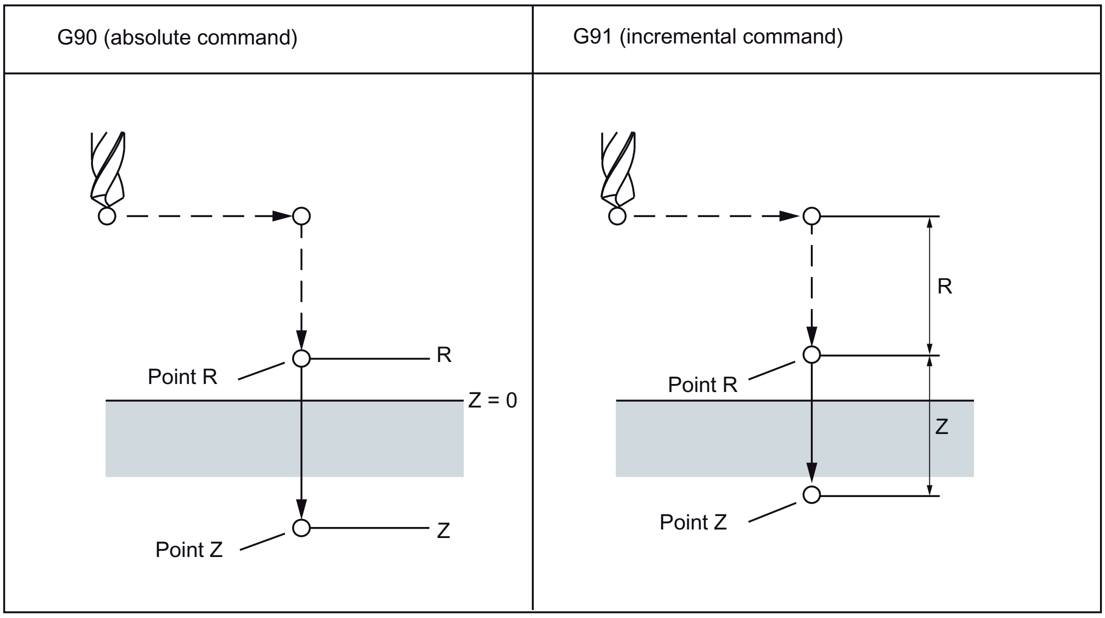


Figure 4-2 Absolute/incremental command G90/G91

3. Drilling mode

   G73, G74, G76 and G81 to G89 are modal G functions and they remain active till they are deselected. The selected drilling cycle is called in each block. The complete parameter assignment of the drilling cycles must be programmed only during the selection (e.g., G81). Only the parameters that are supposed to change are to be programmed in the following blocks.

4. Positioning/reference plane (G98/G99)

   While using the fixed cycles, the retraction plane for the Z-axis is defined with G98/99. G98/G99 are modal G functions. The closed position is normally G98.

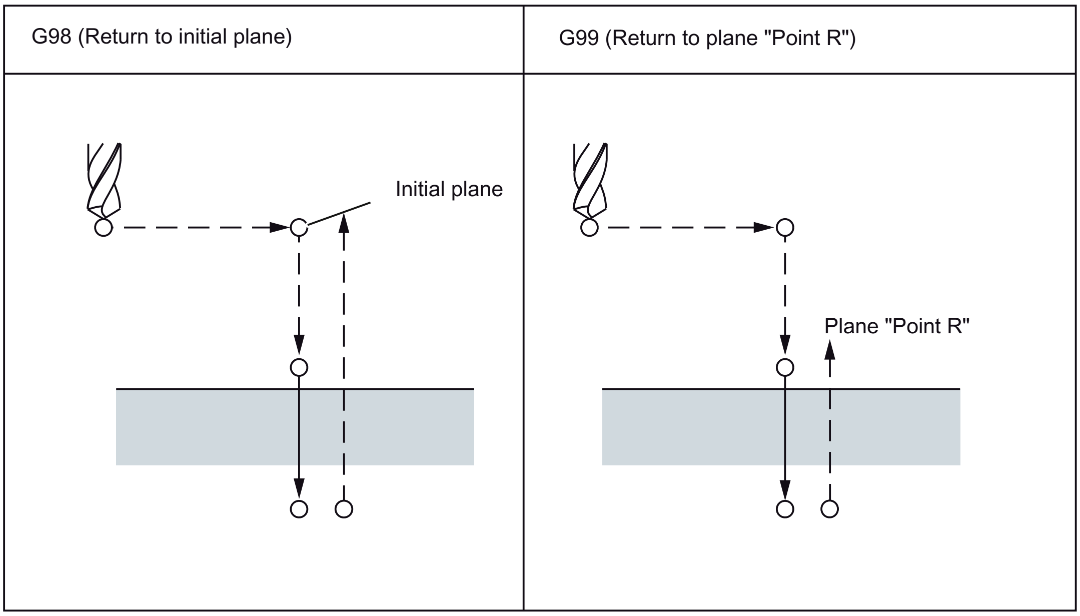

G98 (Return to initial plane)
G99 (Return to plane "Point R")
Initial plane
Plane "Point R"

Figure 4-3 Plane for the return point (G98/G99)

Repeat

If several holes are drilled at uniform spacing, the number of repetitions is specified with "K". "K" is effective only in the block in which it is programmed. If the drilled hole position is programmed as absolute (G90), drilling is done at the same position again; hence the drilled hole position is to be specified as incremental (G91).

Comments

A cycle call remains active till it is deselected again with the G functions G80, G00, G01, G02 or G03 or another cycle call.

Symbols and numbers

The individual fixed cycles are explained in the following sections. The following symbols are used in the numbers occurring in these explanations:

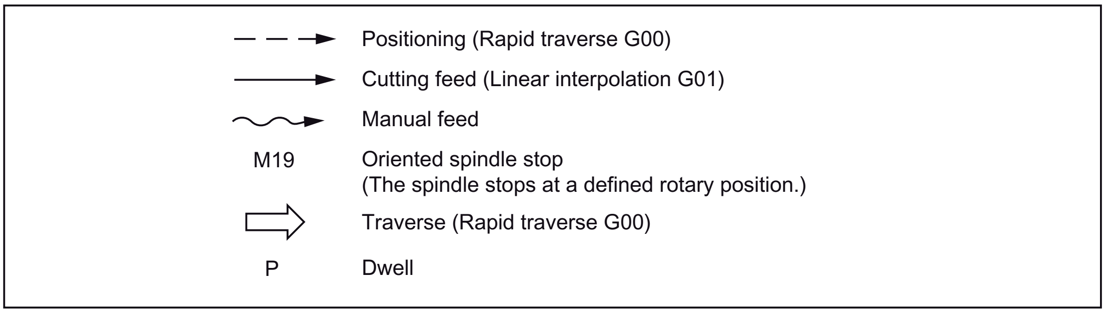

Positioning (Rapid traverse G00)
Cutting feed (Linear interpolation G01)
Manual feed
M19
Oriented spindle stop
(The spindle stops at a defined rotary position.)
Traverse (Rapid traverse G00)
P
Dwell

Figure 4-4 Icons in the numbers

4.1.2 Deep hole drilling cycle with chip breakage (G73)

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. Deep hole drilling is performed with a depth infeed of a maximum defined depth executed several times, increasing gradually until the final drilling depth is reached. Optionally, the twist drill can be retracted after each infeed depth either to the reference plane + safety clearance for chip removal or by the length of the programmed retraction path for chip breakage.

Format

G73 X.. Y... R... Q... F... K... ;

X,Y: Drilled hole position

Z: Distance from Point R to the base of the drilled hole

R: Distance from the initial plane to plane R

Q: Single drilling depth

F: Feedrate

K: Number of repetitions

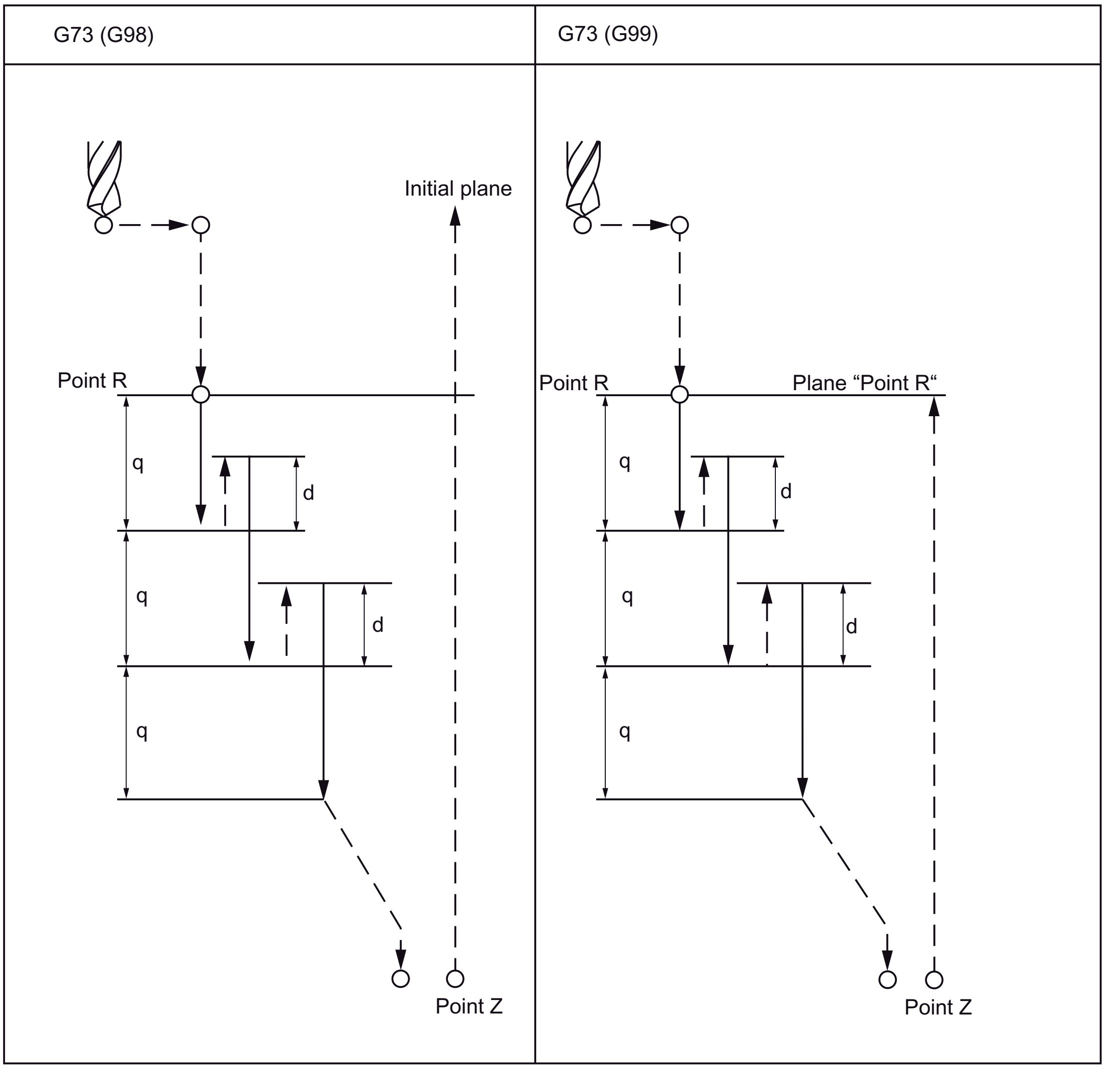


Figure 4-5 Deep hole drilling cycle with chip breakage (G73)

Explanations

On using the G73 cycle, the retraction motion takes place after the drilling with rapid traverse. The safety clearance can be specified with GUD _ZSFR[0]. The retraction amount from chip breaking (d) is defined with GUD _ZSFR[1]:

_ZSFR[1] > 0 Retraction amount as input

_ZSFR[1] v 0 Retraction amount in chip breaking is always 1 mm

The in-feed takes place by using the cutting depth for each cutting Q which is incremented with the retraction amount d as second in-feed.

A rapid drilling infeed results with this drilling cycle. Chip removal takes place through the retraction motion.

Restrictions

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Deep-hole drilling

The drilling cycle is executed only if an axis motion, e.g., is programmed with X, Y, Z or R.

Q/R

Always program Q and R in one block with an axis motion, otherwise the programmed values will not be stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G73 should not be used together in one block, as otherwise G73 is deselected.

Example

```
M3 S1500                         ;Rotary motion of stem
G90 G0 Z100
G90 G99 G73 X200. Y-150. Z-100.  ;Positioning, drilled hole 1,
R50. Q10. F150.                  ;then return to Point R
Y-500.                           ;Positioning, drilled hole 2,
                                 ;then return to Point R
Y-700.                           ;Positioning, drilled hole 3,
                                 ;then return to Point R
X950.                            ;Positioning, drilled hole 4,
                                 ;then return to Point R
Y-500.                           ;Positioning, drilled hole 5,
                                 ;then return to Point R
G98 Y-700.                       ;Positioning, drilled hole 6,
                                 ;then return to initial plane
G80                              ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                 ;Return to reference position
M5                               ;Spindle stop
```

4.1.3 Fine drilling cycle (G76)

Precision drilling takes place with the fine drilling cycle.

Format

G76 X... Y... R... Q... P... F... K... ;

X,Y: Drilled hole position

Z_: Distance from point R to the bottom of the hole

R_: Distance from the initial plane to plane "Point R"

Q_: Amount of offset at the bottom of a hole

P_: Dwell time at the bottom of a hole

F_: Feedrate

K_: Number of repetitions

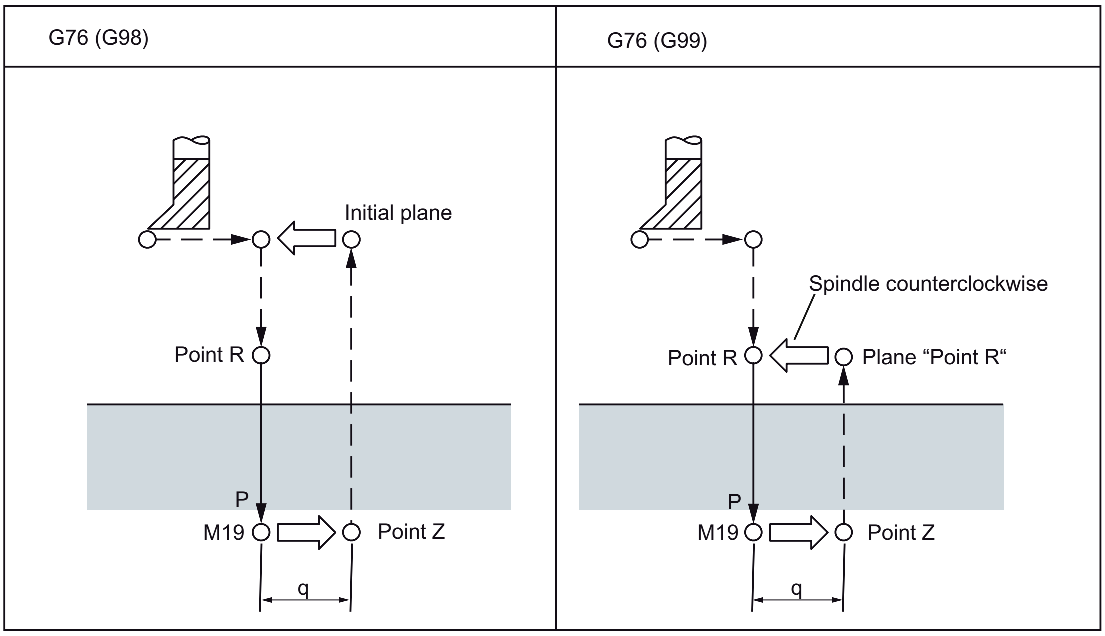
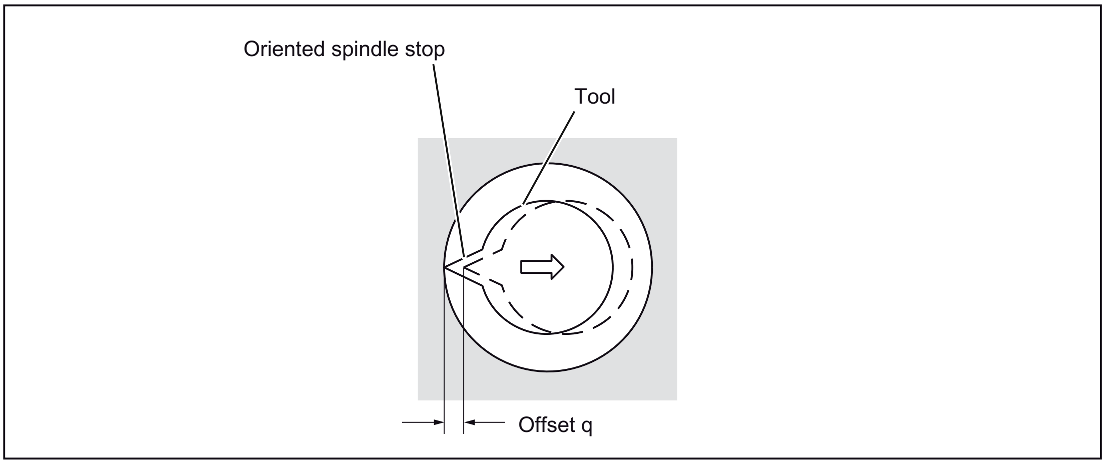


Figure 4-6 Fine drilling cycle (G76)



⚠ WARNING

Address Q

Address Q is a modal value that is stored in fixed cycles. Please ensure that this address is also used as interface for the cycles G73 and G83!

Explanations

The spindles stops at a fixed spindle position after the bottom of a hole is reached. The tool is returned opposite the tool tip.

The safety clearance can be specified with GUD _ZSFR[0]. The lift-off path can be specified with _ZSFI[5].

	G17	G18	G19
_ZSFI[5] = 1	+X	+Z	+Y
_ZSFI[5] = 0 or 2	-X	-Z	-Y
_ZSFI[5] = 3	+Y	+X	+Z
_ZSFI[5] = 4	-Y	-X	-Z

The angle must therefore be specified in GUD7 _ZSFR[2] in such a way that the tool tip points at the opposite direction after the spindle stop, for the lift-off path.

Restrictions

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Boring

The drilling cycle is executed only if an axis motion is programmed, e.g. with X, Y, Z or R.

Q/R

Always program Q and R only in one block with a retracting movement, otherwise the programmed values are not stored modally.

Only one positive value is to be specified in each case for the value of Address Q. If a negative value is specified for Q, the sign is ignored. Q is set as equal to "0" if no lift-off path is programmed. In this case, the cycle is executed without lifting.

Deselection

The G functions of Group 01 (G00 to G03) and G76 should not be used together in one block, as otherwise G76 is deselected.

Example

```
M3 S300                         ;Rotary motion of stem
G90 G0 Z100
G90 G99 G76 X200. Y-150. Z-100. ;Positioning, drilling of drilled hole 1,
R50. Q10. P1000 F150.           ;then return to point R and
                                ;for 1 s stop at the bottom of a hole
Y-500.                          ;Positioning, drilled hole 2,
                                ;then return to point R
Y-700.                          ;Positioning, drilled hole 3,
                                ;then return to point R
X950.                           ;Positioning, drilled hole 4,
                                ;then return to point R
Y-500.                          ;Positioning, drilled hole 5,
                                ;then return to point R
G98 Y-700.                      ;Positioning, drilled hole 6,
                                ;then return to initial plane
G80                             ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                ;Return to reference position
M5                              ;Spindle stop
```

4.1.4 Drilling cycle, preboring (G81)

This cycle can be used for centering and preboring. The retraction motion starts immediately with rapid traverse rate on reaching the drilling depth Z.

Format

G81 X... Y... Z... R... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to plane R

F: Cutting feedrate

K: Number of repetitions

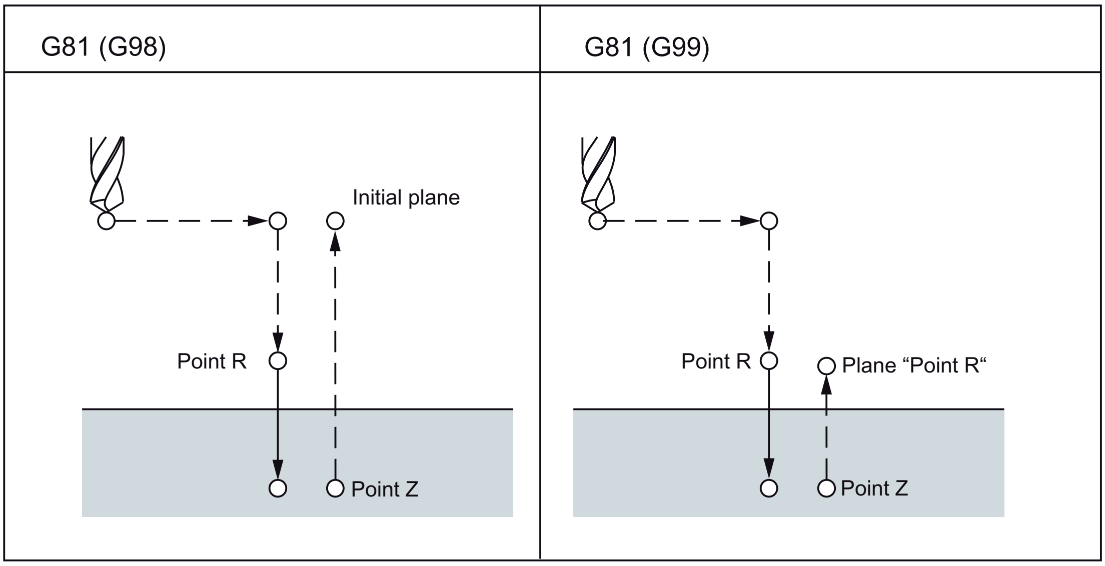


Figure 4-7 Drilling cycle, preboring (G81)

Restrictions

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion is programmed, e.g. with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G76 should not be used together in one block, as otherwise G76 is deselected.

Example

```
M3 S1500                          ;Spindle rotation
G90 G0 Z100
G90 G99 G81 X200. Y-150. Z-100.   ;Positioning, drilled hole 1,
R50. F150.                        ;then return to point R and
                                  ;for 1 s stop at the bottom of the hole
Y-500.                            ;Positioning, drilled hole 2,
                                  ;then return to point R
Y-700.                            ;Positioning, drilled hole 3,
                                  ;then return to point R
X950.                             ;Positioning, drilled hole 4,
                                  ;then return to point R
Y-500.                            ;Positioning, drilled hole 5,
                                  ;then return to point R
G98 Y-700.                        ;Positioning, drilled hole 6,
                                  ;then return to initial plane
G80                               ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                  ;Return to reference position
M5                                ;Spindle stop
```

4.1.5 Drilling cycle, preboring (G82)

This cycle can be used for normal drilling. A programmed dwell time can be active on reaching the drilling depth Z; the retraction motion is then executed in rapid traverse.

Format

G82 X... Y... R... P... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to plane R

P: Dwell time at the bottom of a hole

F: Feedrate

K: Number of repetitions

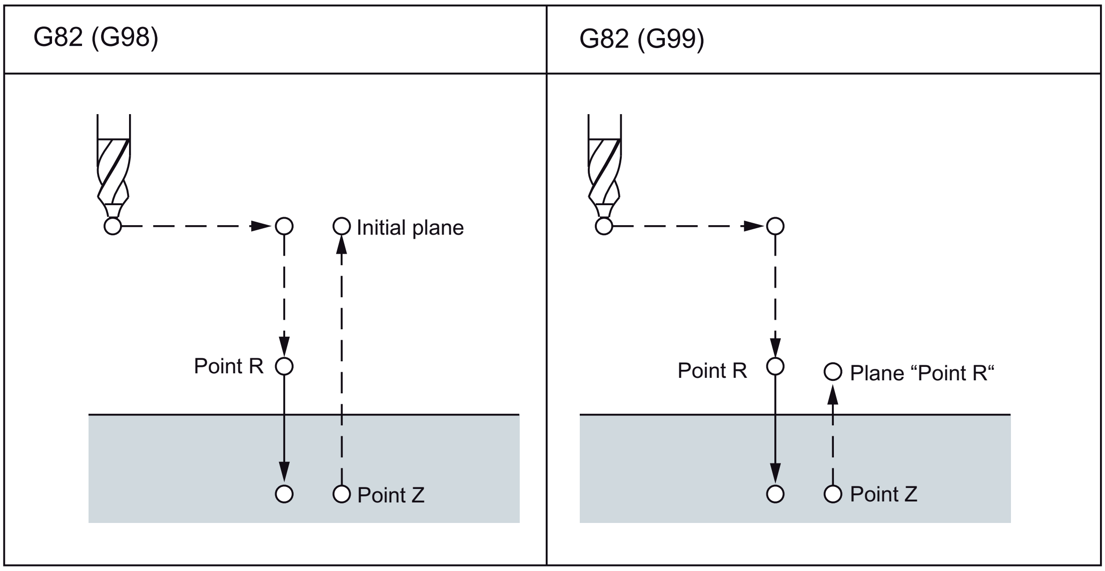


Figure 4-8 Drilling cycle, countersink cycle (G82)

Restrictions

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G82 should not be used together in one block, as otherwise G82 is deselected.

Example

```
M3 S2000                          ;Rotary motion of stem
G90 G0 Z100
G90 G99 G82 X200. Y-150. Z-100.   ;Positioning, drilled hole 1,
R50. P1000 F150.                  ;stop on the bottom of a hole for 1 s
                                  ;then return to point R
Y-500.                            ;Positioning, drilled hole 2,
                                  ;then return to point R
Y-700.                            ;Positioning, drilled hole 3,
                                  ;then return to point R
X950.                             ;Positioning, drilled hole 4,
                                  ;then return to point R
Y-500.                            ;Positioning, drilled hole 5,
                                  ;then return to point R
G98 Y-700.                        ;Positioning, drilled hole 6,
                                  ;then return to initial plane
G80                               ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                  ;Return to reference position
M5                                ;Spindle stop
```

4.1.6 Deep hole drilling cycle with chip removal (G83)

The cycle "Deep hole drilling with chips removal" can, e.g., be used for deep hole drilling with recutting.

Format

G83 X... Y... R... Q... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to plane R

Q: Cutting depth for each cutting feedrate

F: Feedrate

K: Number of repetitions

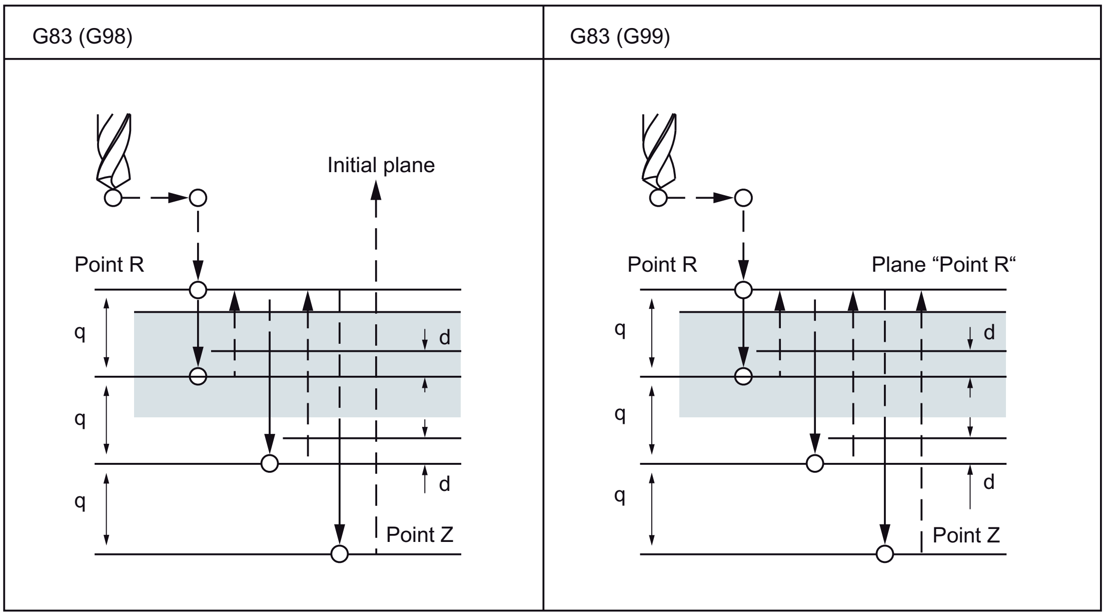


Figure 4-9 Deep hole drilling cycle with chip removal (G83)

Restrictions

Explanations

After the programmed cutting depth is reached for each cutting feedrate Q, the return to the reference plane R takes place in rapid traverse. The approach motion for a renewed step is also executed in rapid traverse, around the path (d) that can be set in GUD7 _ZSFR[10]. The path and the cutting depth for each cutting feedrate Q are traversed with cutting feedrate. Q is incremental without having to specify signs.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. X, Y, Z or R is programmed.

Q/R

Always program Q and R in one block with an axis motion, otherwise the programmed values will not be stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G83 should not be used together in one block, as otherwise G83 is deselected.

Example

```
M3 S2000                         ;Rotary motion of stem
G90 G0 Z100
G90 G99 G83 X200. Y-150. Z-100.  ;Positioning, drilled hole 1,
R50. Q10. F150.                  ;then return to point R
Y-500.                           ;Positioning, drilled hole 2,
                                 ;then return to point R
Y-700.                           ;Positioning, drilled hole 3,
                                 ;then return to point R
X950.                            ;Positioning, drilled hole 4,
                                 ;then return to point R
Y-500.                           ;Positioning, drilled hole 5,
                                 ;then return to point R
G98 Y-700.                       ;Positioning, drilled hole 6,
                                 ;then return to initial plane
G80                              ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                 ;Return to reference position
M5                               ;Spindle stop
```

Note

If _ZSFR[10]

- > 0 = Value is used for the derivative path "d" (minimum distance 0.001)
- = 0 The derivative path is 30 mm and the value of the derivative path is always 0.6 mm. The drilling depth/50 formula is always used for larger drilling depths (maximum value 7 mm).

4.1.7 Drilling cycle (G85)

Format

G85 X... Y... R... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to plane R

F: Feedrate

K: Number of repetitions

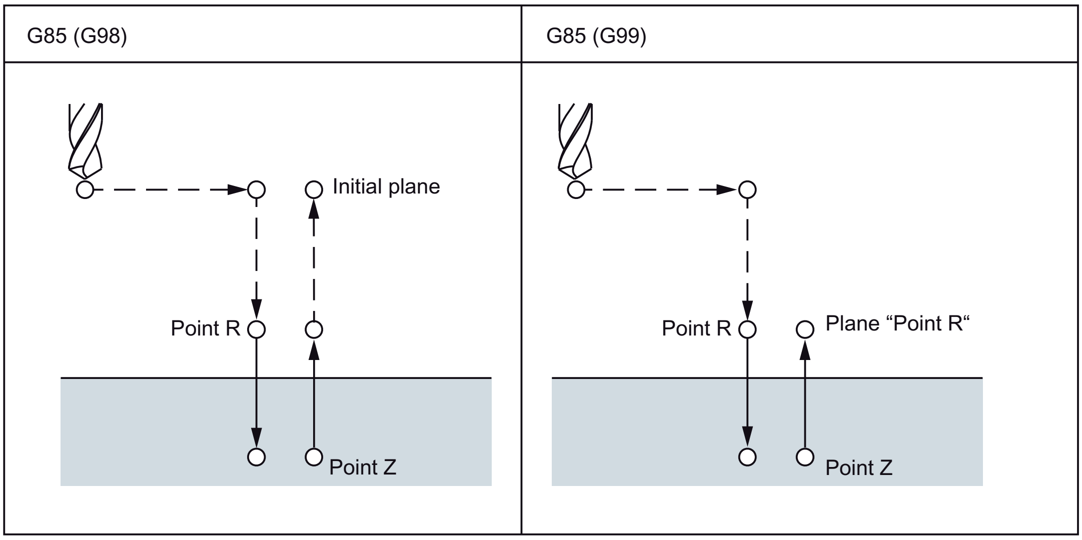


Figure 4-10 Drilling cycle (G85)

Explanations

A traversing movement takes place to point R in rapid traverse after the positioning along the X and Y axis. Drilling takes place from point R to point Z. On reaching point Z, a retraction motion to point R takes place with cutting feedrate.

Restrictions

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G85 should not be used together in one block, as otherwise G85 is deselected.

Example

```
M3 S150                          ;Rotary motion of stem
G90 G0 Z100
G90 G99 G85 X200. Y-150. Z-100.  ;Positioning, drilled hole 1,
R50. F150.                       ;then return to point R
Y-500.                           ;Positioning, drilled hole 2,
                                 ;then return to point R
Y-700.                           ;Positioning, drilled hole 3,
                                 ;then return to point R
X950.                            ;Positioning, drilled hole 4,
                                 ;then return to point R
Y-500.                           ;Positioning, drilled hole 5,
                                 ;then return to point R
G98 Y-700.                       ;Positioning, drilled hole 6,
                                 ;then return to initial plane
G80                              ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                 ;Return to reference position
M5                               ;Spindle stop
```

4.1.8 Boring cycle (G86)

Format

G86 X... Y... R... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to point R

F: Feedrate

K: Number of repetitions

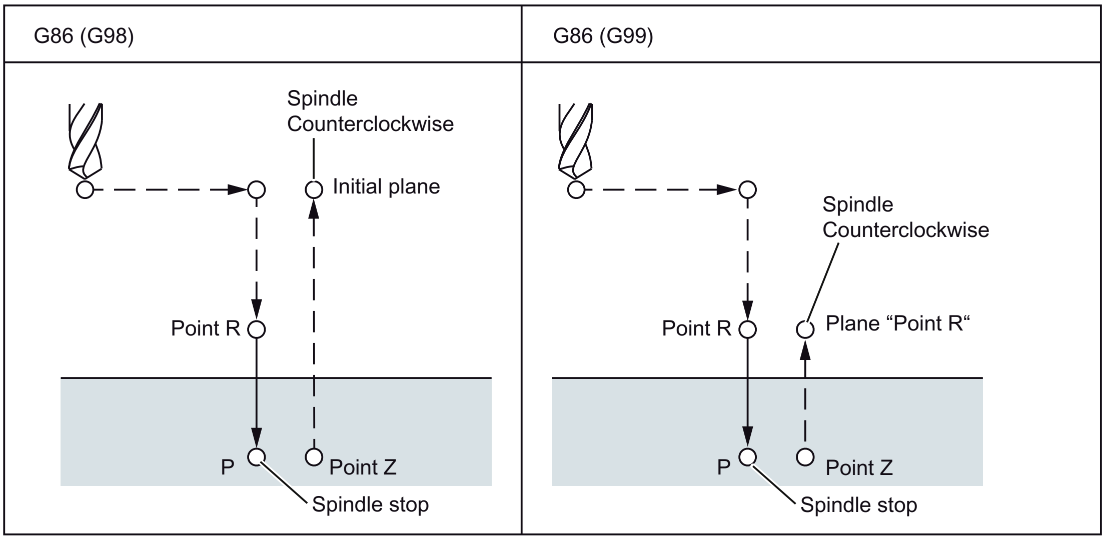


Figure 4-11 Boring cycle (G86)

Explanations

Point R is approached in rapid traverse after positioning the X and Y axes. Drilling takes place from point R to point Z. The tool returns in rapid traverse mode after the spindle is stopped at the bottom of a hole.

Restrictions

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G86 should not be used together in one block, as otherwise G86 is deselected.

Example

```
M3 S150                          ;Rotary motion of stem
G90 G0 Z100
G90 G99 G86 X200. Y-150. Z-100.  ;Positioning, drilled hole 1,
R50. F150.                       ;then return to point R
Y-500.                           ;Positioning, drilled hole 2,
                                 ;then return to point R
Y-700.                           ;Positioning, drilled hole 3,
                                 ;then return to point R
X950.                            ;Positioning, drilled hole 4,
                                 ;then return to point R
Y-500.                           ;Positioning, drilled hole 5,
                                 ;then return to point R
G98 Y-700.                       ;Positioning, drilled hole 6,
                                 ;then return to initial plane
G80                              ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                 ;Return to reference position
M5                               ;Spindle stop
```

4.1.9 Boring cycle, reverse countersinking (G87)

This cycle can be used for precision drilling.

Format

G87 X... Y... R... Q... P... F... K... ;

X,Y: Drilled hole position

Z: Distance from bottom of the hole to point Z

R: Distance from the initial plane to plane R (bottom of a hole)

Q: Tool offset amount

P: Dwell time

F: Feedrate

K: Number of repetitions

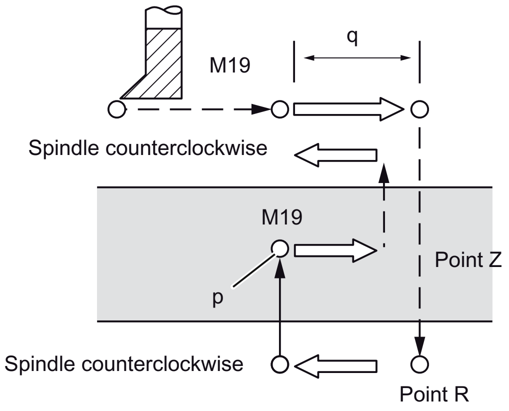
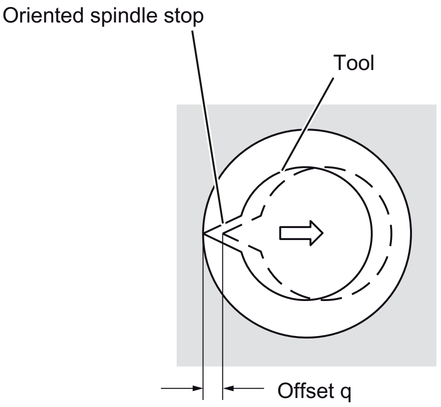
G87 (G98)	G87 (G99)
 q M19 Spindle counterclockwise M19 Point Z p Spindle counterclockwise Point R	Is not used

Figure 4-12 Boring cycle, reverse countersinking (G87)



WARNING

Address Q

Address Q (gear change at the base of a drilled hole) is a modal value that is stored in fixed cycles. Please ensure that this address is also used as interface for the cycles G73 and G83!

Explanations

The spindle stops at a fixed rotary position after positioning along the X and Y axis. The tool travels in the direction opposite to that of the tool tip. It is positioned on the bottom of a hole (Point R) at rapid traverse.

Finally, the tool is shifted in the direction of the tool tip and the spindle is moved with clockwise rotation. Boring takes place along the Z axis in the positive direction up to point Z.

The spindles stops at a fixed spindle position after the bottom of a hole is reached. The tool is returned opposite the tool tip.

The safety clearance can be specified with GUD _ZSFR[0].

The lift-off path can be specified with _ZSFI[5].

	G17	G18	G19
_ZSFR[5] = 1	+X	+Z	+Y
_ZSFI[5] = 0 or 2	-X	-Z	-Y
_ZSFI[5] = 3	+Y	+X	+Z
_ZSFI[5] = 4	-Y	-X	-Z

The angle must therefore be specified in GUD7 _ZSFR[2] in such a way that the tool tip points at the opposite direction after the spindle stop for the lift-off path.

Example:

If plane G17 is activated, the tool tip must point in direction +X.

Restrictions

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Boring

The drilling cycle is executed only if an axis motion is programmed, e.g. with X, Y, Z or R.

Q/R

Always program Q and R in one block with an axis motion, otherwise the programmed values will not be stored modally.

Only one positive value is to be specified in each case for the value of Address Q. If a negative value is specified for "Q", the sign is ignored. "Q" is set as equal to "0" if no lift-off path is programmed. In this case, the cycle is executed without lifting.

Deselection

The G functions of Group 01 (G00 to G03) and G87 should not be used together in one block, as otherwise G87 is deselected.

Example

```
M3 S400                          ;Rotary motion of stem
G90 G0 Z100
G90 G87 X200. Y-150. Z-100.      ;Positioning, drilled hole 1,
R50. Q3. P1000 F150.             ;orientation towards initial plane,
                                 ;then travel 3 mm,
                                 ;halt for 1 s at point Z
Y-500.                           ;Positioning, drilled hole 2
Y-700.                           ;Positioning, drilled hole 3
X950.                            ;Positioning, drilled hole 4
Y-500.                           ;Positioning, drilled hole 5
G98 Y-700.                       ;Positioning, drilled hole 6
G80                              ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                 ;Return to reference position
M5                               ;Spindle stop
```

4.1.10 Drilling cycle (G89), return with G01

Format

G89 X... Y... R... P... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to point R

P: Dwell time at the bottom of a hole

F: Feedrate

K: Number of repetitions

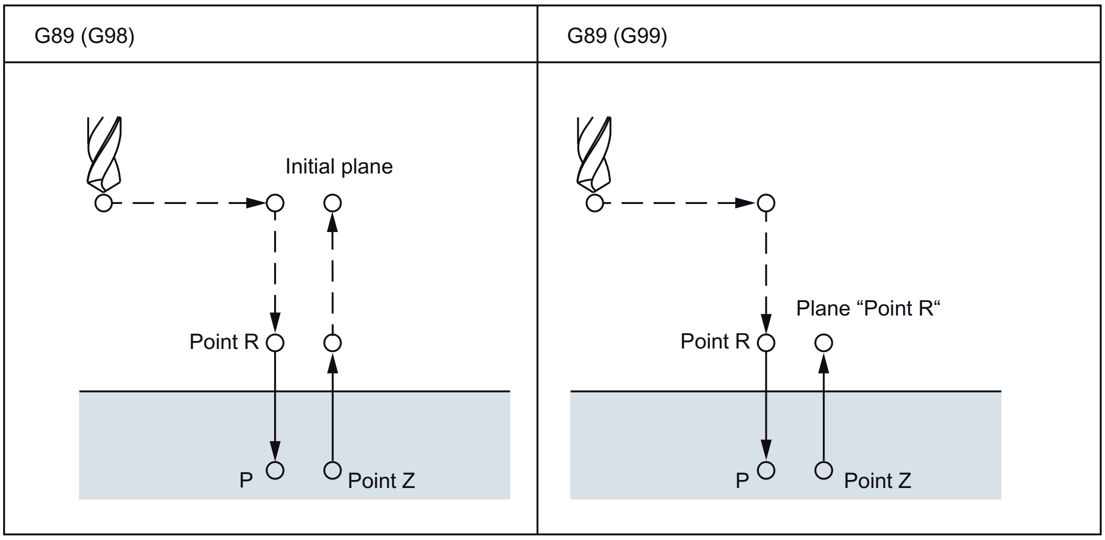


Figure 4-13 Boring cycle (G89)

Explanations

This cycle is similar to G86, with the only exception that here, a dwell time at the bottom of the hole is still available.

Before programming G89, the spindle must be started with an M function.

Restrictions

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G89 should not be used together in one block, as otherwise G89 is deselected.

Example

```
M3 S150                           ;Rotary motion of stem
G90 G0 Z100
G90 G99 G89 X200. Y-150. Z-100.   ;Positioning, drilled hole 1,
R50. P1000 F150.                  ;then 1 s stop at the bottom of a hole
Y-500.                            ;Positioning, drilled hole 2,
                                  ;then return to point R
Y-700.                            ;Positioning, drilled hole 3,
                                  ;then return to point R
X950.                             ;Positioning, drilled hole 4,
                                  ;then return to point R
Y-500.                            ;Positioning, drilled hole 5,
                                  ;then return to point R
G98 Y-700.                        ;Positioning, drilled hole 6,
                                  ;then return to initial plane
G80                               ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                  ;Return to reference position
M5                                ;Spindle stop
```

4.1.11 Cycle "Tapping without compensating chuck" (G84)

The tool drills at the programmed spindle speed and feedrate to the entered final thread depth. With G84 you can produce rigid tapping.

Note

G84 can be used if the spindle to be used for the drilling operation is technically able to be operated in the position-controlled spindle mode.

Format

G84 X... Y... Z... R... P... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to plane R

P: Dwell time at the bottom of the hole and at point R during return

F: Cutting feedrate

K: Number of repetitions (if necessary)

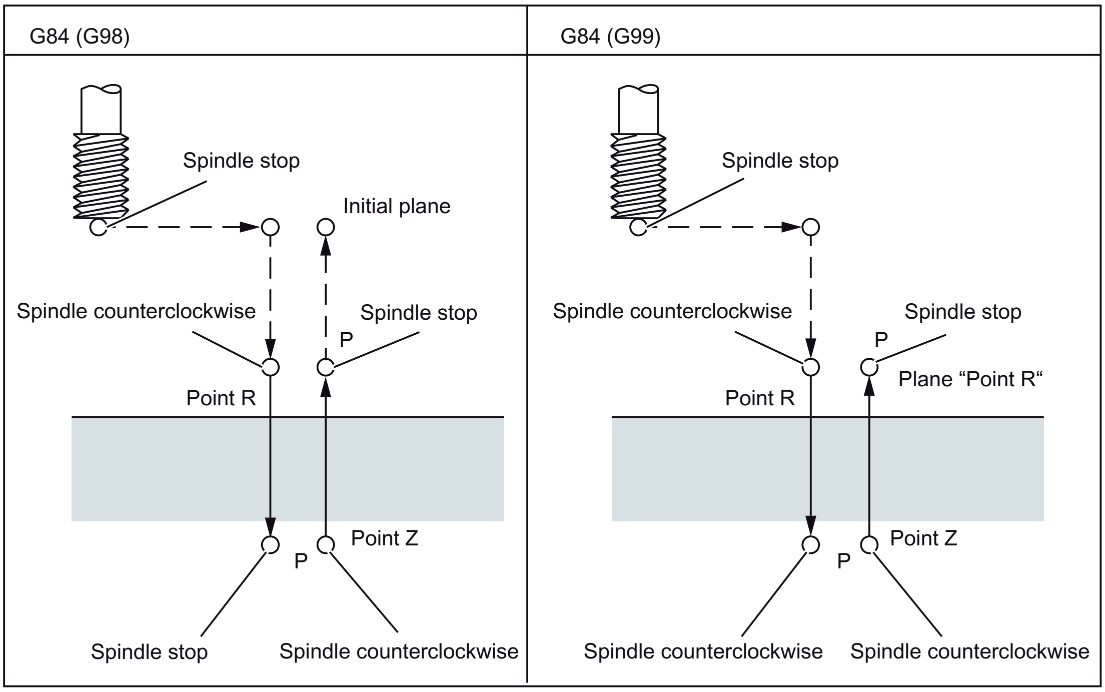


Figure 4-14 Tapping without compensating chuck (G84)

Explanations

The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0.
- Oriented spindle stop and transfer of spindle in the Axis mode.
- Tapping to the final drilling depth.
- Execution of dwell time at thread depth.
- Retraction to the reference plane and reversion of direction of rotation brought forward by the safety clearance.
- Retraction to the retraction plane with G0.

During tapping, rapid traverse override and spindle override are accepted at 100%.

The speed of rotation can be affected during the retraction with GUD _ZSFI[2]. Example: _ZSFI[2]=120; the retraction takes place at 120% of the speed during tapping.

Restrictions

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle. An alarm is output if the drilling axis in the "Drilling without compensating chuck" mode is changed over.

Tapping

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G84 should not be used together in one block, as otherwise G84 is deselected.

S command

An error message is displayed if the specified gear stage is one step higher than the maximum permissible value.

F function

An error message is displayed if the value specified for the cutting feedrate exceeds the maximum permissible value.

Unit of the F command

	Metric input	Input in inch	Remarks
G94	1 mm/min	0.01 inch/min	Decimal point programming is permitted
G95	0.01 mm/rev	0.0001 inch/rev	Decimal point programming is permitted

Example

Feedrate for the Z axis 1.000 mm/min

Spindle speed 1,000 rev/min

Thread lead 1.0 mm

```
<Programming as feedrate per minute>
S100 M3
G94                              ;Feedrate per minute
G00 X100.0 Y100.0                ;Positioning
G84 Z-50.0 R-10.0 F1000          ;Tapping without compensating chuck
<Programming as revolutional feedrate>
G95                              ;Rev. feedrate
G98 Y-700.                       ;Positioning, drilled hole 6,
                                 ;then return to initial plane
G00 X100.0 Y100.0                ;Positioning
G84 Z-50.0 R-10.0 F1.0           ;Tapping without compensating chuck
```

4.1.12 "Drilling a left-hand thread without compensating chuck" cycle (G74)

The tool drills at the programmed spindle speed and feedrate to the entered final thread depth. With G74 you can produce left rigid tapping.

Note

G74 can be used if the spindle to be used for the drilling operation is technically able to be operated in the position-controlled spindle mode.

Format

G74 X... Y... Z... R... P... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to point R

P: Dwell time at the bottom of the hole and at point R during return

F: Cutting feedrate

K: Number of repetitions (if necessary)

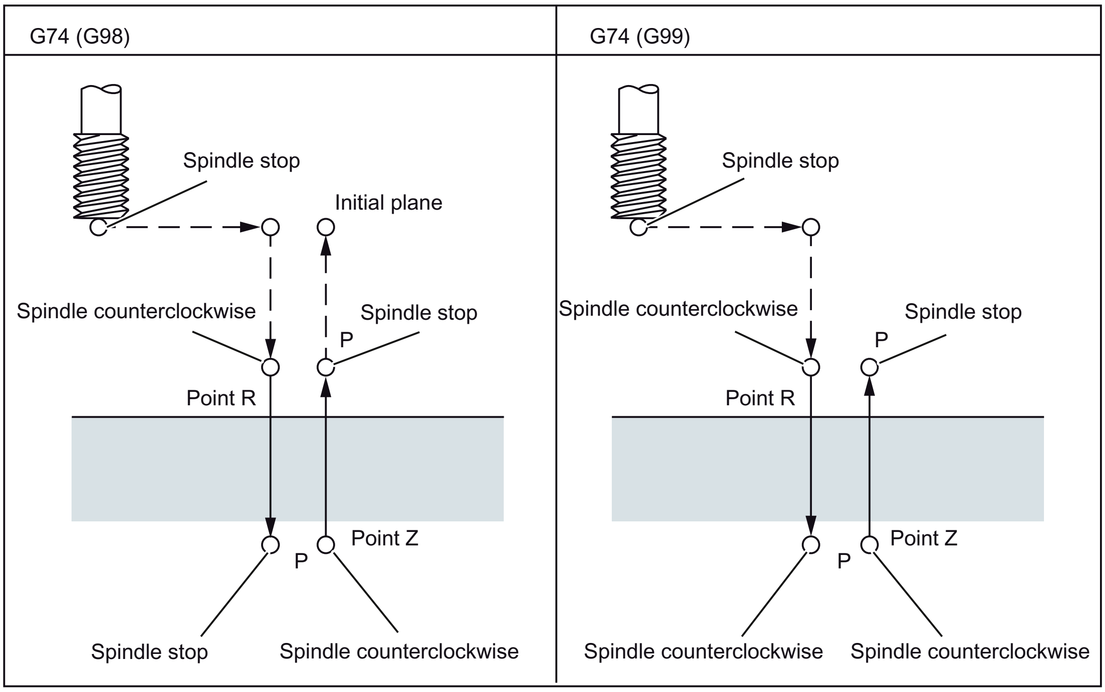


Figure 4-15 "Drilling a left-hand thread without compensating chuck" cycle (G74)

Explanations

The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0.
- Oriented spindle stop and transfer of spindle in the Axis mode.
- Tapping to the final drilling depth.
- Execution of dwell time at thread depth.
- Retraction to the reference plane and reversion of direction of rotation brought forward by the safety clearance.
- Retraction to the retraction plane with G0.

During tapping, rapid traverse override and spindle override are accepted at 100%.

The speed of rotation can be affected during the retraction with GUD _ZSFI[2]. Example: _ZSFI[2]=120; the retraction takes place at 120% of the speed during tapping.

Restrictions

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle. An alarm is output if the drilling axis in the "Drilling without compensating chuck" mode is changed over.

Tapping

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G84 should not be used together in one block, as otherwise G84 is deselected.

S command

An error message is displayed if the specified gear stage is one step higher than the maximum permissible value.

F function

An error message is displayed if the value specified for the cutting feedrate exceeds the maximum permissible value.

Unit of the F command

	Metric input	Input in inch	Remarks
G94	1 mm/min	0.01 inch/min	Decimal point programming is permitted
G95	0.01 mm/rev	0.0001 inch/rev	Decimal point programming is permitted

Example

Feedrate for the Z axis 1.000 mm/min

Spindle speed 1,000 rev/min

Thread lead 1.0 mm

```
<Programming as feedrate per minute>
S100 M3
G94                              ;Feedrate per minute
G00 X100.0 Y100.0                ;Positioning
G84 Z-50.0 R-10.0 F1000          ;Tapping without compensating chuck
<Programming as revolutional feedrate>
G95                              ;Rev. feedrate
G98 Y-700.                       ;Positioning, drilled hole 6,
                                 ;then return to initial plane
G00 X100.0 Y100.0                ;Positioning
G84 Z-50.0 R-10.0 F1.0           ;Tapping without compensating chuck
```

4.1.13 Left or right tapping cycle (G84 or G74)

Due to the chips adhering to the tool and an increased resistance associated with this, it may be difficult to perform the deep-hole tapping without compensating chuck. In such cases the tapping cycle with chip breakage or chip removal is helpful.

The cutting movement is executed in this cycle until the root is reached. There are a total of two tapping cycles for this: Deep-hole tapping with chip breakage and deep-hole tapping with chip removal.

The G84 and G74 cycles can be selected with MD55800 $SCS_ISO_M_DRILLING_AXIS_IS_Z as follows:

2: Deep-hole tapping with chip breakage

3: Deep-hole tapping with chip removal

Format

G84 (or G74) X... Y... Z... R... P... Q... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to "Point R"

P: Dwell time at the bottom of the hole and at point R during return

Q: Cutting depth for each cutting feedrate

F: Feedrate

K: Number of repetitions

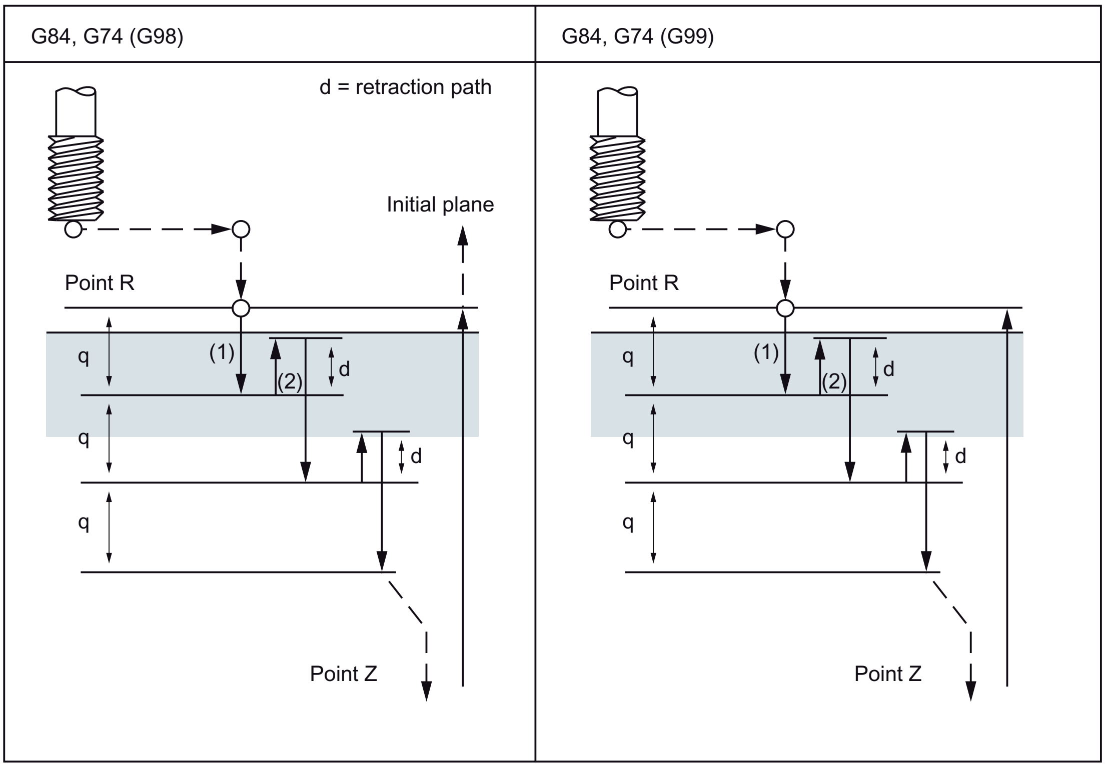


Figure 4-16 Deep-hole tapping with chip breakage (2)

1. The tool is traversed with the programmed feedrate.
2. The retraction velocity can be influenced by means of MD55804 $SCS_ISO_M_RETRACTION_FACTOR.

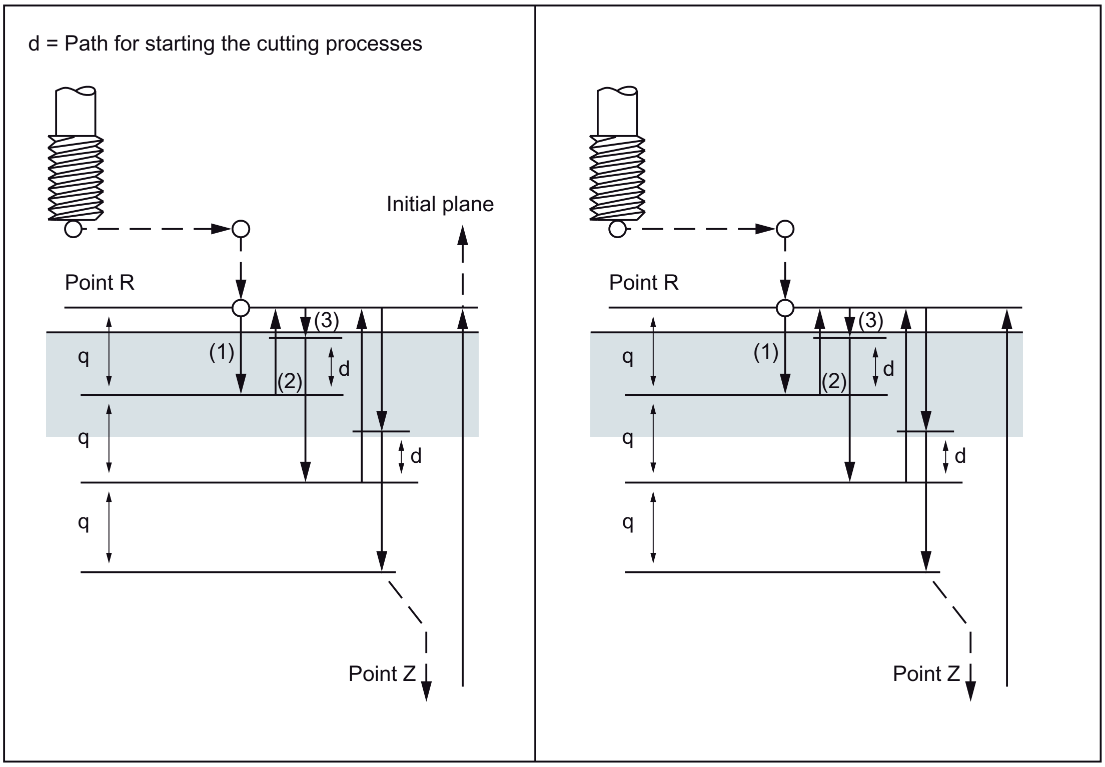


Figure 4-17 Deep-hole drilling with chip removal (3)

Deep-hole tapping with chip breakage/removal

After positioning along the X and Y axes, there is a traversing movement at rapid traverse to point R. The machining is done from point R onwards with a cutting depth Q (cutting depth per cutting feedrate). Finally, the tool is retracted by the distance d. If a value not equal to 100 % is specified in $SCS_ISO_M_RETRACTION_FACTOR, it can be specified whether the retraction is overlaid or not. The spindle stops as soon as point Z is reached; the direction of rotation is finally reversed and a retraction is executed. The retraction distance d is set in MD55802 $SCS_ISO_M_DRILLING_TYPE.

Note

If "0" is specified in $SCS_ISO_M_DRILLING_TYPE, the default setting for the retraction distance of 1 mm or 1 inch is effective.

If 0 mm or 0 inch is to be specified, a value less than the travel triggering should be specified.

4.1.14 Deselection of a fixed cycle (G80)

Fixed cycles can be deselected with G80.

Format

G80;

Explanations

All modal cycles are deselected in the ISO mode with G80 or with a G function of the 1st group (G00, G03, G33, G34, ...).

4.1.15 Program example with a tool length compensation and fixed cycles

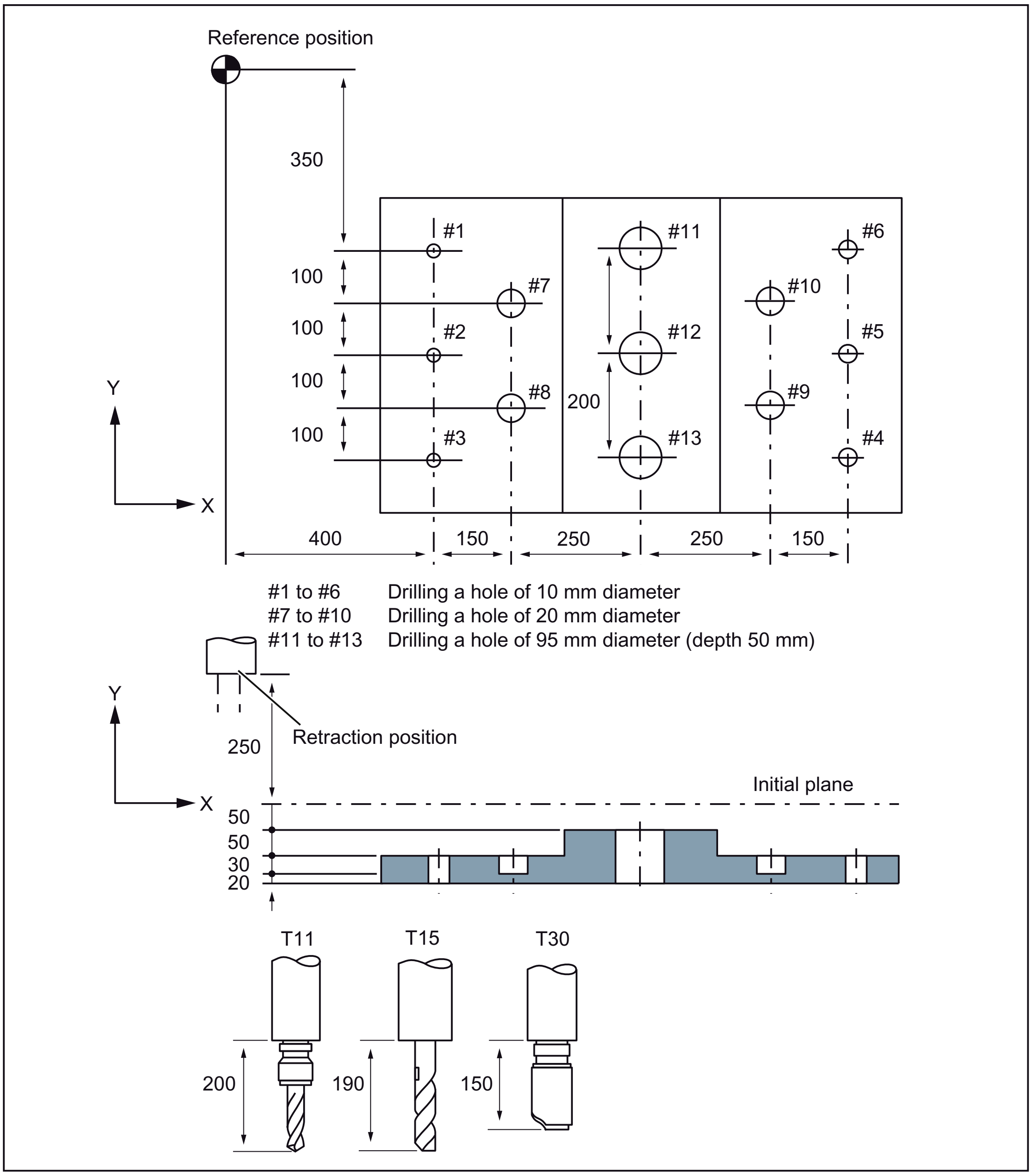


Figure 4-18 Program example (drilling cycle)

Offset value +200.0 is set in TO No. 11, +190.0 is set in TO No. 15 and +150.0 is set in tool offset No. 30.

Sample program

```
;
N001 G49                            ; Deselect the tool length compensation
N002 G10 L10 P11 R200.              ; Setting the tool offset 11 to +200.
N003 G10 L10 P15 R190.              ; Setting the tool offset 15 to +190.
N004 G10 L10 P30 R150.              ; Setting the tool offset 30 to +150.
N005 G92 X0 Y0 Z0                   ; Setting the coordinates at the reference
                                    ; position
N006 G90 G00 Z250.0 T11 M6          ; Tool change
N007 G43 Z0 H11                     ; Initial plane, tool length compensation
N008 S30 M3                         ; Spindle start
N009 g99 G81 X400.0 Y-350.0 Z-153.0 ; Positioning, then drill #1
R-97.0 F1200
N010 Y-550.0                        ; Positioning, then drilling #2 and return to
                                    ; plane point R
N011 G98 Y-750.0                    ; Positioning, then drilling #3 and return to
                                    ; initial plane
N012 G99 X1200.0                    ; Positioning, then drilling #4 and return to
                                    ; plane point R
N013 Y-550.0                        ; Positioning, then drilling #5 and return to
                                    ; plane point R
N014 G98 Y-350.0                    ; Positioning, then drilling #6 and return to
                                    ; initial plane
N015 G00 X0 Y0 M5                   ; Return to reference position,
                                    ; Spindle stop
N016 G49 Z250.0 T15 M6              ; Deselection of tool length compensation, tool
                                    ; change
N017 G43 Z0 H15                     ; Initial plane, tool length compensation
N018 S20 M3                         ; Spindle start
N019 G99 G82 X550.0 Y-450.0 Z-130.0 ; Positioning, then drilling #7 and return to
R-97.0 P300 F700                    ; plane point R
N020 G98 Y-650.0                    ; Positioning, then drilling #8 and return to
                                    ; initial plane
N021 G99 X1050.0                    ; Positioning, then drilling #9 and return to
                                    ; plane point R
N022 G98 Y-450.0                    ; Positioning, then drilling #10 and return to
                                    ; initial plane
N023 G00 X0 Y0 M5                   ; Return to reference position,
                                    ; Spindle stop
N024 G49 Z250.0 T30 M6              ; Deselection of tool length compensation, tool
                                    ; change
 N025 G43 Z0 H30                    ; Initial plane, tool length compensation
N026 S10 M3                         ; Spindle start
N027 G85 G99 X800.0 Y-350.0 Z-153.0 ; Positioning, then drilling #11 and return to
R47.0 F500                          ; plane point R
N028 G91 Y-200.0 K2                 ; Positioning, then drilling #12 and 13, and
                                    ; return to plane point R
```

```
N029 G28 X0 Y0 M5                     ; Return to reference position,
                                      ; Spindle stop
N030 G49 Z0                           ; Deselect the tool length compensation
N031 M30                              ; End of the program
```

4.1.16 Multiple-start threads with G33

Multiple-start threads are programmed with the G code G33 in the ISO dialect mode.

Format

G33 X.. Z.. F.. Q..

X.. Z.. = End point of the tread

F.. = Thread lead

Q.. = Starting angle

Threads with offset slides are programmed by specifying the mutually offset start points in the G33 block. The starting point offset is specified as the absolute angle position under the address "Q". The associated setting data ($SD_THREAD_START_ANGLE) is changed appropriately.

Example:

Q45000 means: Start offset 45.000 degree

Range of values: 0.0000 to 359.999 degrees

The start angle must always be programmed as an integer. The input resolution of the angular data is 0.001 degree.

Example:

```
N200 X50 Z80 G01 F.8 G95 S500 M3
```

```
N300 G33 Z40 F2 Q180000
```

A thread with a lead of 2 mm and a starting point offset of 180 degree is produced.

4.2 Programmable data input (G10)

4.2.1 Changing the tool offset value

Existing tool offsets can be overwritten via G10. It is not possible to create new tool offsets.

Format

G10 L10 P... R... ; Tool length compensation, geometry

G10 L11 P... R... ; Tool length compensation, wear and tear

G10 L12 P... R... ; Tool radius compensation, geometry

G10 L13 P... R... ; Tool radius compensation, wear and tear

P: Number of tool offset memory

R: Value statement

L1 can also be programmed instead of L11.

4.2.2 Working area limitation (G22, G23)

G22/G23

G22/G23 limits the working area (working area, working space) in which the tool can traverse, limited to the channel axis. The areas outside the working area limitations defined with G22/G23 are inhibited for any tool movement.

While using the commands G22 and G23, a protection zone as per the setting in the machine data must be available and it must be active.

18190 $MN_NUM_PROTECT_AREA_NCK = 1

28210 $MC_NUM_PROTECT_AREA_ACTIVE = 1

Further, the following machine data must be set:

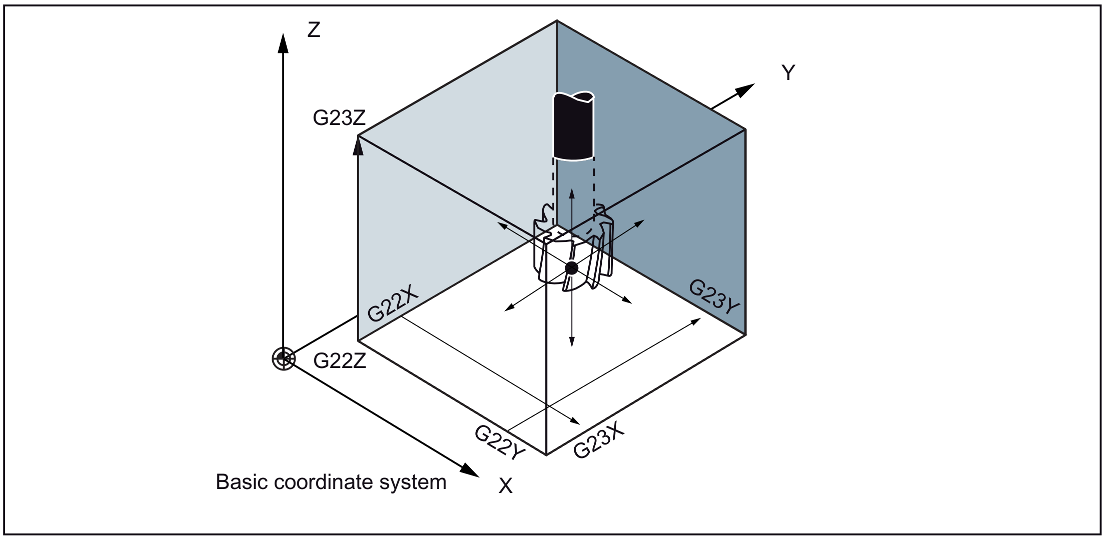
18190 $MN_NUM_PROTECT_AREA_NCK = 2 (minimum)

28210 $MC_NUM_PROTECT_AREA_ACTIVE = 2 (minimum)

One upper limit (G23) and a lower limit (G22) is defined for each axis for the working area. These values are effective right away and they remain active even after RESET or POWER ON.

Consideration of the tool radius must be activated separately. This takes place over MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS.

If the tool reference point lies outside the working area limitation defined by the working space or if this area is left, the program sequence is stopped.



Status at Power ON

Whether the working area limitation is activated or deactivated is defined in the following machine data:

$MC_EXTERN_GCODE_RESET_VALUES[3]

Normally, this MD is set to Value 2 (G23).

4.2.3 M function for calling subroutines (M98, M99)

This function can be used if subprograms are stored in the part program memory. Subprograms that are registered in the memory and whose program numbers are assigned can be called and executed any number of times.

Commands

The following M functions are used to call the subprograms.

Table 4- 3 M functions for calling subprograms

M function	Function
M98	Subprogram call
M99	End of subprogram

Subprogram call (M98)

- M98 P nnn mmmm

  m: Program no. (max. 4 digits)
  n: No. of repetitions (max. 4 digits)

- If for example, M98 P21 is programmed, the part program memory is browsed by program name 21.mpf and the subprogram is executed once. To call the subprogram three times, one must program M98 P30021. An alarm is output if the specified program number is not found.
- A nesting of subprograms is possible, up to 16 subprograms are allowed. An alarm is output if more subprogram levels are assigned than is allowed.

End of subprogram (M99)

A subprogram is ended with the command M99 Pxxxx and program processing is continued in Block No. Nxxxx. The control system first searches forward for the block number (from the subprogram call up to the end of the program). If no matching block number is found, the part program is eventually searched in the reverse direction (in the direction of the start of the part program).

If M99 is without a block number (Pxxxx) in a main program, the control goes to the start of the main program and the main program is processed afresh. In case of M99 with navigation to the block number in the main program (M99xxxx), the block number is always searched from the start of program.

M99 does not reset the program runtime. An activated workpiece counter is not incremented.

4.3 Eight-digit program number

An eight-digit program number selection is activated with the machine data 20734 $MC_EXTERN_FUNCTION_MASK, Bit 6=1. This function affects M98, G65/66 and M96.

y: Number of program runs

x: Program number

Subprogram call

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 0

M98 Pyyyyxxxx or

M98 Pxxxx Lyyyy

Max. four-digit program number

Addition of program number always to 4 digits with 0

Example:

M98 P20012: calls 0012.mpf 2 flows

M98 P123 L2: calls 0123.mpf 2 flows

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 1

M98 Pxxxxxxxx Lyyyy

There is no extension with 0, even if the program number has less than 4 digits.

The programming of number of passes and program number in P(Pyyyyxxxxx) is not possible, the number of passes must always be programmed with L!

Example:

M98 P123: calls 123.mpf 1 Pass

M98 P20012: calls 20012.mpf 1 Pass

Caution: This is no longer compatible with ISO dialect original

M98 P12345 L2: calls 12345.mpf 2 passes

Modal and blockwise Macro G65/G66

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 0

G65 Pxxxx Lyyyy

Addition of program number to 4 digits with 0. Program number with more than 4 digits leads to an alarm.

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 1

G65 Pxxxx Lyyyy

There is no extension with 0, even if the program number has less than 4 digits. A program number with more than 8 digits leads to an alarm.

Interrupt M96

Does not function in SINUMERIK 802D sl.

$MC_EXTERN_FUNCTION_MASK, Bit6 = 0

M96 Pxxxx

Addition of program number always to 4 digits with 0

$MC_EXTERN_FUNCTION_MASK, Bit6 = 1

M96 Pxxxx

There is no extension with 0, even if the program number has less than 4 digits. A program number with more than 8 digits leads to an alarm.

4.4 Polar coordinates (G15, G16)

While programming in polar coordinates, the positions in the coordinate system are defined with a radius and/or angle. Polar coordinate programming is selected with G16. It is deselected again with G15. The first axis of the plane is interpreted as polar radius, the second axis as polar angle.

Format

```
G17 (G18, G19) G90 (G91) G16          ;Polar coordinates command ON
G90 (G91) X... Y... Z...              ;Polar coordinates command
...
...
G15                                   ;Polar coordinates command OFF
```

G16: Polar coordinates command

G15: Deselection of polar coordinates command

G17, G18, G19: Selection of plane

G90: The pole is located on the workpiece zero.

G91: The pole is located on the current position.

X, Y, Z: First axis: Radius of polar coordinate, second axis: Angle of polar coordinate

Note

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,
                           ;the pole is at the workpiece zero,
                           ;Position X 70.711 Y 70.711
                           ;in the Cartesian coordinate system
N15 G91 X100 Y0            ;the pole is the current Position,
                           ;i.e. the Position X 170.711 Y 70.711
N20 G90 Y90.               ;No X in the block
                           ;The pole is on the workpiece zero,
                           ;Radius = SORT(X*X +Y*Y) = 184.776
G15
```

The polar radius is always taken as absolute value, while the polar angle can be interpreted as absolute value as well as incremental value.

4.5 Polar coordinates interpolation (G12.1, G13.1)

An interpolation between a rotary axis and a linear axis in the machining plane is switched on or off via G12.1 and G13.1. Another possible linear axis lies vertical to this plane.

This function corresponds to the TRANSMIT function in the Siemens mode.

Note

A detailed description of the TRANSMIT function can be found in the NC Function Description "SINUMERIK 840D sl, Extended Functions", Section "Kinematic Transformation (M1)" and in the work planning Programming Manual (PGA) "SINUMERIK 840D sl" in Section "Transformation".

G12.1 is based on the Siemens function TRANSMIT. Appropriate machine data is to be set for this.

Format

```
G12.1        ;Selection of polar coordinates interpolation
...
...
G13.1        ;Deselection of polar coordinates interpolation
```

CAUTION

Plane selection

On specifying G12.1, the relevant previously used plane (G17, G18, G19) is deselected.

Operation with polar coordinates interpolation is deselected via NC RESET, where the previously active plane is activated again.

Possible G functions in operation with polar coordinates interpolation

G01: Linear interpolation

G02, G03: Circular interpolation

G04: Dwelling, exact stop

G40, G41, G42: Cutter radius compensation

G65, G66, G67: Customer macro command

G90, G91: Absolute command, incremental command

G94, G95: Feedrate per minute, revolutional feedrate

Example

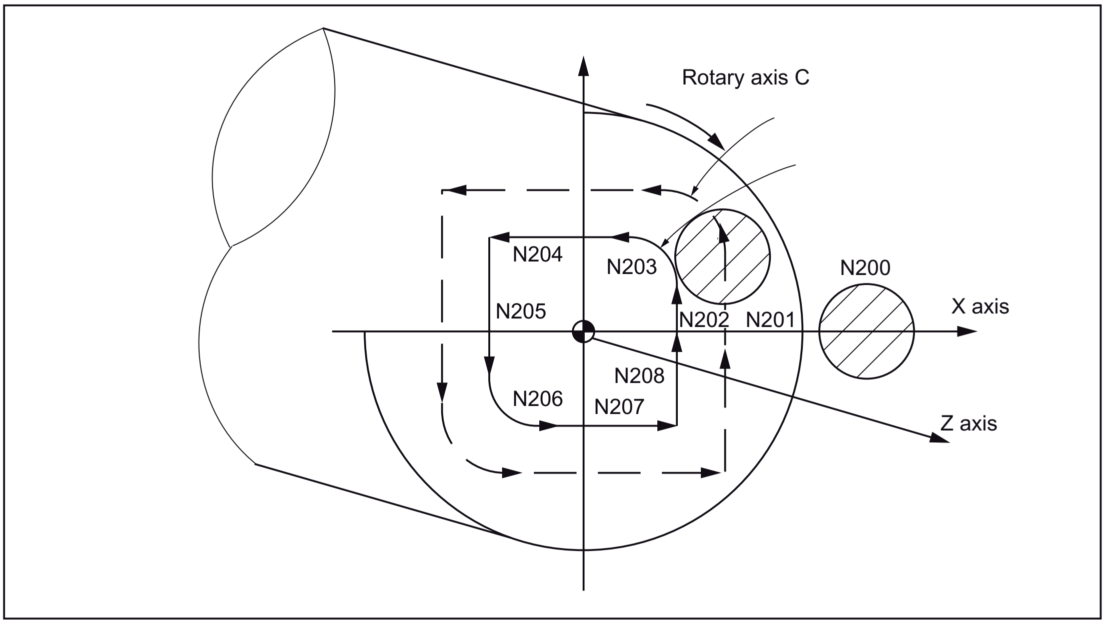


Figure 4-19 Example of polar coordinates interpolation

```
00001
N010 T0101
N0100 G90 G00 X60.0 C0 Z..                ;TRANSMIT selection
N0200 G12.1
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

No geo axis interchange (parallel axes with G17 (G18. G19) should be active.

4.6 Measuring functions

4.6.1 Rapid lift with G10.6

A retraction position for the rapid lifting of a tool can be activated with G10.6 <Axis position> (e.g., in case of tool breakage). The retraction motion itself is started with a digital signal. The 2nd rapid input of NC is used as the start signal.
Another rapid input (1-8) can also be selected with machine data 10820 $MN_EXTERN_INTERRUPT_NUM_RETRAC (1 - 8).

The interrupt program (ASUB) CYCLE3106.spf must always be available for the rapid retraction with G10.6. If the CYCLE3106.spf is not available in the part program memory, the Alarm 14011 "Program CYCLE3106 not available or not released for processing" is output with G10.6 in a part program block.

The response of the control system after the rapid retraction is defined in ASUB CYCLE3106.spf. If the axes and the spindle are stopped after the rapid retraction, M0 and M5 must be programmed in CYCLE3106.spf. If CYCLE3106.spf is a dummy program that contains only M17, the part program is continued without any interruption after the rapid retraction.

If the rapid retraction is activated with the programming G10.6 <Axis position>, then the change in the input signal of the 2nd NC rapid input from 0 to 1 aborts the current movement and the position programmed in the G10.6 block is moved at rapid traverse. Here, the positions are approached as absolute or incremental, as programmed in the G10.6 block.

The function is deactivated with G10.6 (without position specification). Rapid retraction via the input signal of the 2nd rapid NC input is blocked.

Restrictions

Only one axis can be programmed for rapid retraction.

4.6.2 Measuring with "delete distance-to-go" (G31)

Measuring with "Delete distance-to-go possible" is activated by specifying "G31 X... Y... Z... F... ;". The linear interpolation is interrupted and the distance-to-go of the axes is deleted if, during the linear interpolation, the measurement input of the 1st probe is active. The program is continued with the next block.

Format

G31 X... Y... Z... F... ;

G31: Non-modal G function (operates only in the block in which it is programmed)

PLC signal "Measurement input = 1"

With the rising edge of the measurement input 1, the current axis positions are stored in the axial system parameters or $AA_MM[<Axis>], $AA_MW[<Axis>]. These parameters can be read in the Siemens mode.

$AA_MW[X]	Saving the coordinate value of the X axis in the workpiece coordinate system
$AA_MW[Y]	Saving the coordinate value of the Y axis in the workpiece coordinate system
$AA_MW[Z]	Saving the coordinate value of the Z axis in the workpiece coordinate system
$AA_MM[X]	Saving the coordinate value of the X axis in the machine coordinate system
$AA_MM[Y]	Saving the coordinate value of the Y axis in the machine coordinate system
$AA_MM[Z]	Saving the coordinate value of the Z axis in the machine coordinate system

Note

Alarm 21700 is output if G31 is activated when the measuring signal is still active.

Program continuation after the measuring signal

If incremental axis positions are programmed in the next block, these axis positions are related to the measuring point, i.e. the reference point of the incremental position is the axis position at which the delete distance-to-go was executed by the measuring signal.

If the axis positions in the next block are programmed as absolute, then the programmed positions are approached.

Note

No cutter radius compensation should be active in a block containing G31. Hence, the cutter radius compensation is to be deselected before programming of G31, with G40.

Example

G31 with incremental position specification

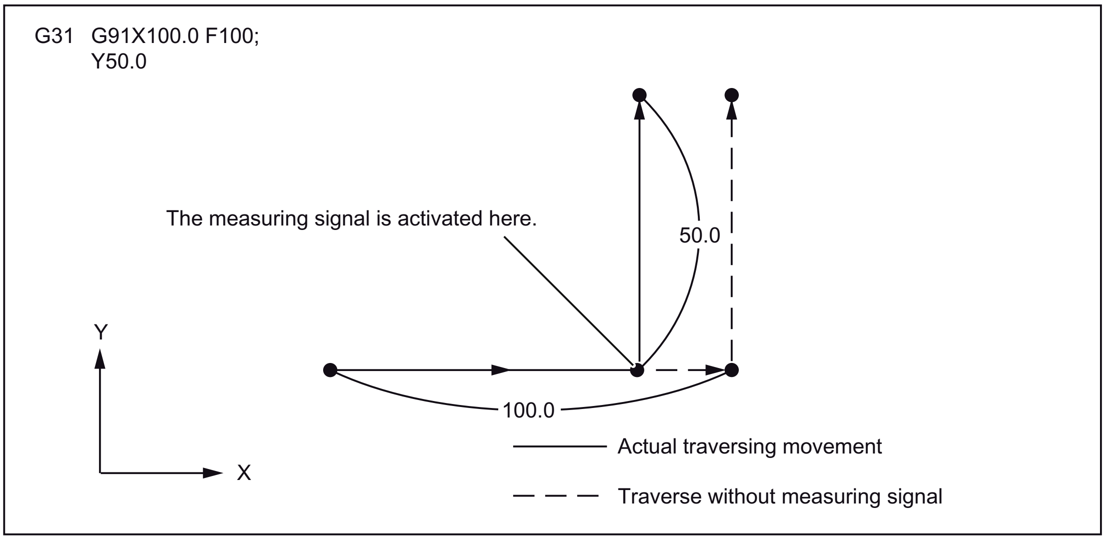

G31 G91X100.0 F100;
Y50.0
The measuring signal is activated here.
50.0
Y
X
100.0
Actual traversing movement
Traverse without measuring signal

Figure 4-20 G31 with incremental position specification of one axis

G31 is an absolute position specification

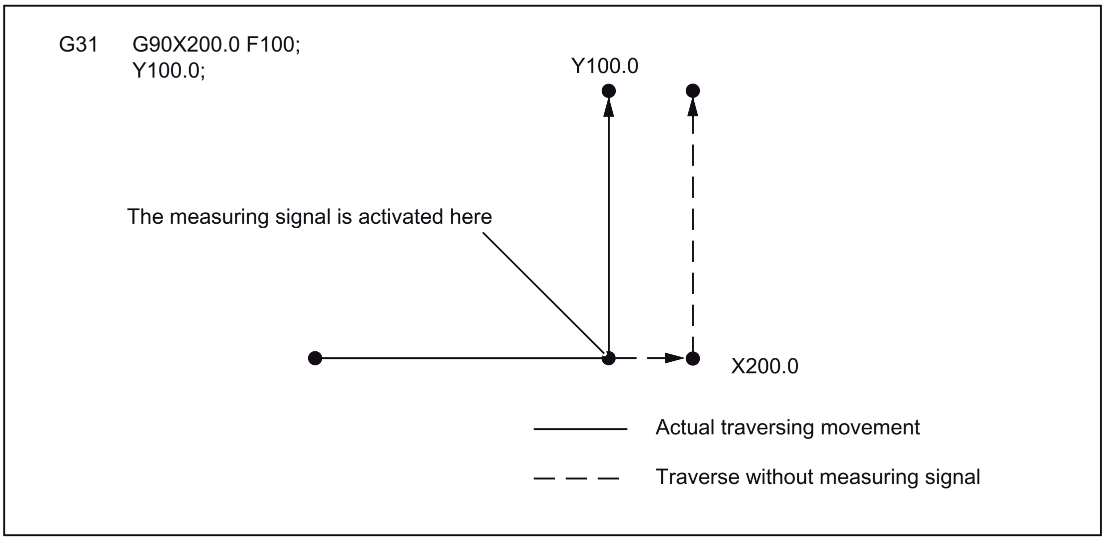

G31 G90X200.0 F100;
Y100.0;
Y100.0
The measuring signal is activated here
X200.0
Actual traversing movement
Traverse without measuring signal

Figure 4-21 G31 with absolute position specification of one axis

G31 is an absolute command for 2 axes.

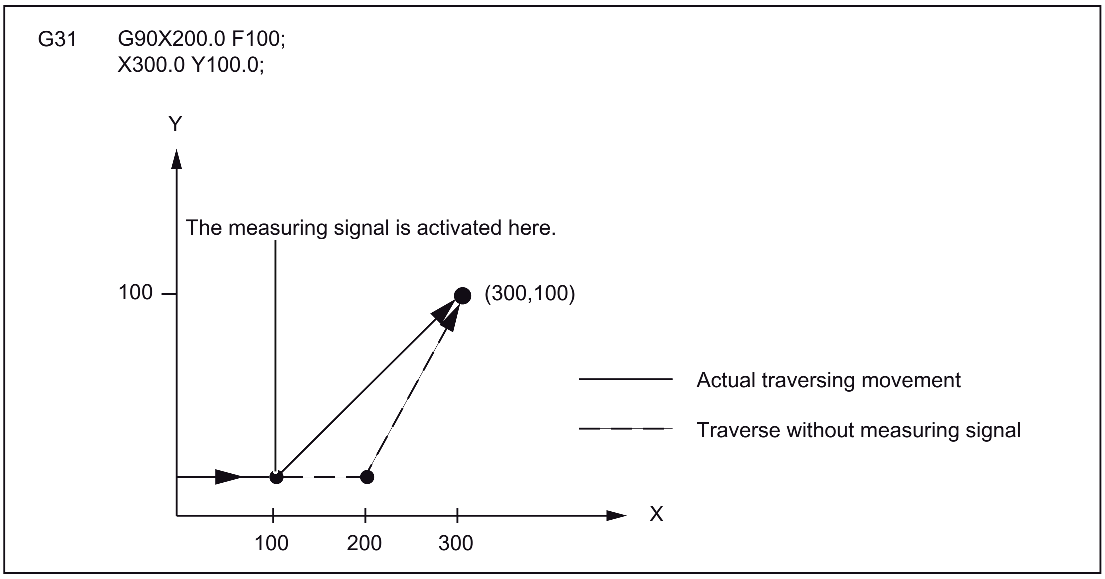

G31 G90X200.0 F100;
X300.0 Y100.0;
Y
The measuring signal is activated here.
100
(300,100)
Actual traversing movement
Traverse without measuring signal
X
100
200
300

Figure 4-22 G31 is an absolute command for 2 axes

4.6.3 Measuring with G31, P1 - P4

The function G31 P1 (.. P4) is different from G31 in that different inputs for the measuring signal can be selected with P1 to P4. Several inputs can also be monitored on a rising edge of a measuring signal simultaneously. The assignment of the inputs to the addresses P1 to P4 is defined through machine data.

Format

G31 X... Y... Z... F... P... ;

X, Y, Z: End point

F...: Feedrate

P...: P1 - P4

Explanation

The digital inputs are assigned to the Addresses P1 to P4 via machine data as follows:

P1: $MN_EXTERN_MEAS_G31_P_SIGNAL[0]

P2: $MN_EXTERN_MEAS_G31_P_SIGNAL[1]

P3: $MN_EXTERN_MEAS_G31_P_SIGNAL[2]

P4: $MN_EXTERN_MEAS_G31_P_SIGNAL[3]

Explanations for selection (P1, P2, P3 or P4) can be found in the documentation of your machine manufacturer.

4.6.4 Interrupt program with M96, M97

M96

A subprogram can be defined as an interrupt routine with the M96 P<Program No.>.

The start of this program is triggered by an external signal. To start the interrupt routine, the 1st rapid NC input is used from among the eight inputs available in the Siemens mode. Another rapid input (1 to 8) can also be selected with MD10818 $MN_EXTER_INTERRUPT_NUM_ASUP.

Format

M96 Pxxxx	;Activation of program interrupt
M97	;Deactivation of program interrupt

M97 and M96 P_ must be alone in the block.

So that on triggering the interrupt, the cover cycle CYCLE396 is called first and it calls the interrupt program programmed with Pxxxx in the ISO mode. At the end of the cover cycle, the machine data 10808 $MN_EXTERN_INTERRUPT_BITS_M96, Bit 1 is evaluated and either positioned on the interruption point with REPOS or continued with the next block.

End of interruption (M97)

The interrupt program is deactivated with M97. Only after the next activation with M96 can the interrupt routine be started with the external signal.

If the interrupt program programmed with M96 Pxx is to be called directly with the interrupt signal (without intermediate step with CYCLE396), then machine data 20734 $MC_EXTERN_FUNCTION_MASK, Bit 10 must be set. The subprogram programmed with Pxx is then called in the Siemens mode during a signal change from 0 -> 1.

The M function numbers for the interrupt function are set through machine data. Machine data 10804 $MN_EXTERN_M_NO_SET_INT is used to determine the M number for activating an interrupt routine, machine data 10806 $MN_EXTERN_M_NO_DISABLE_INT is used to determine the M number for suppressing an interrupt routine.

Only the M functions not reserved for standard M functions can be used. The default of the M functions is M96 and M97. To activate the function, one must set bit 0 in machine data 10808 $MN_EXTERN_INTERRUPT_BITS_M96. The M functions are not output to the PLC. The M functions are interpreted as normal auxiliary functions if Bit 0 is not set.

At the end of the interrupt program, one normally traverses to the end position of the part program block following the interruption block. If the part program is to be processed further from the interruption point, there must be a REPOS instruction at the end of the interrupt program, e.g. REPOSA. For this, the interrupt program must be written in the Siemens mode.

The M function for activating and deactivating an interrupt program must be alone in the block. The system issues Alarm 12080 (syntax error) if addresses other than "M" and "P" are programmed in the block.

Machine data

The response of the interrupt program function can be determined from the following machine data:

MD10808 $MN_EXTERN_INTERRUPT_BITS_M96:
Bit 0 = 0
Interrupt program is not possible as M96/M97 are normal M functions.
Bit 0 = 1
Activation of an interrupt program with M96/M97 is allowed.

Bit 1 = 0
The part program is processed further with the end position of the block immediately after the interruption block (REPOSL RMEBL).
Bit 1 = 1
The part program is continued from the interruption position (REPOSL RMIBL).

Bit 2 = 0
The interrupt signal interrupts the current block immediately and starts the interrupt routine.
Bit 2 = 1
The interrupt routine is started only at the end of the block.

Bit 3 = 0
The execution cycle is interrupted immediately after an interrupt signal arrives.
Bit 3 = 1
The interrupt program is started only at the end of the execution cycle (evaluation in the shell cycles).

Bit 3 is evaluated in the shell cycles,and the cycle sequence is adapted accordingly.

Bit 1 is evaluated in cover cycle CYCLE396.

If the interrupt program is not called via the cover cycle CYCLE396, ($MC_EXTERN_FUNCTION_MASK, Bit 10 = 1) must be evaluated with Bit 1. If Bit 1 = TRUE, REPOSL RMIBL must be used for positioning on the interruption point, otherwise REPOSL RMEBL must be used for positioning on the block end position.

Example:

`N100 M96 P1234`	`;Activate ASUB 1234spf. In the case of a rising edge of`
	`;1st rapid input, the program`
	`;1234.spf is started`

`....`	
`....`	
`N300 M97`	`;Deactivation of ASUB`

Restrictions

The interrupt routine is treated as a normal subprogram. In other words, to be able to execute interrupt routines, at least one subprogram level must be free. (16 program levels are available, plus two levels that are reserved for the ASUB interrupt programs.)

The interrupt routine is started only during an edge change of the interrupt signal from 0 to 1. If the interrupt signal remains permanently on 1, then the interrupt program is not restarted any more.

4.6.5 "Tool life control" function

Tool life monitoring and workpiece count can be undertaken with Siemens Tool Management.

4.7 Macro programs

Macros may consist of several part program blocks that are completed with M99. In principle, macros are subroutines that are called with G65 Pxx or G66 Pxx in the part program.

Macros that are called with G65 are non-modal. Macros that are called with G68 are modal and are deselected again with G67.

4.7.1 Differences with subroutines

Macro programs (G65, G66) can be used to specify parameters that can be evaluated in the macro programs. No parameters can be specified in subroutine calls (M98).

4.7.2 Macro program call (G65, G66, G67)

Macro programs are generally executed immediately after their call.

The procedure of calling a macro program is described in the following table.

Table 4- 4 Format for calling a macro program

Call method	Command code	Remarks
Simple call	G65	
Modal call (a)	G66	Deselection through G67

Simple call (G65): Format

G65 P_ L_ ;

A macro program to which a program number was assigned with "P" is called and executed "L" times by specifying "G65 P ... L... <Argument>; ".

The required parameters must be programmed in the same block (with G65).

Explanation

In a part program block containing G65 or G66, the address Pxx is interpreted as program number of the subroutine in which the macro functionality is programmed. The number of passes of the macro can be defined with the address Lxx. All other addresses in this part program block are interpreted as transfer parameters and their programmed values are stored in the system variables $C_A to $C_Z. These system variables can be read in the subroutine and evaluated for the macro functionality. If other macros with parameter transfer are called in a macro (subroutine), then the transfer parameters in the subroutine must be saved in internal variable before the new macro call.

To enable internal variable definitions, one must switch automatically to the Siemens mode during macro call. One can do this by inserting the instruction PROC<Program name> in the first line of the macro program. If another macro call is programmed in the subroutine, then the ISO-dialect-mode must be reselected in advance.

Table 4- 5 The P and L command

Address	Description	Number of digits
P	Program number	4 to 8 digits
L	Number of repetitions	

System variables for the addresses I, J, K

As the addresses I, J, and K can be programmed up to 10 times in a block containing macro call, the system variables of these addresses must be accessed with an array index. The syntax of these three system variables thus is $C_I[..], $C_J[..], $C_K[..]. The values remain in the programmed sequence in the array. The number of I, J, K addresses programmed in the block is given in the variables $C_I_NUM, $C_J_NUM, $C_K_NUM.

The transfer parameters I, J, K for macro calls are treated in each case as one block even if the individual addresses are not programmed. If a parameter is reprogrammed, or a following parameter based on the I, J, K sequence was programmed, it belongs to the next block.

The system variables $C_I_ORDER, $C_J_ORDER, $C_K_ORDER are set to detect the programming sequence in the ISO mode. These are identical arrays of $C_I, $C_K and they contain the associated numbers of the parameters.

Note

The transfer parameters can be read only in the subroutine in the Siemens mode.

Example:

```
N5 I10 J10 K30 J22 K55 I44 K33

  Block1  Block2  Block3

$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 the ISO-dialect-0 mode, the programmed values can be evaluated in different ways, depending on the programming method (integer or actual value). The different evaluation is activated through a machine data.

If the MD is set, the control system responds as in the following example:

X100 ; X axis is traversed by 100 mm (100. with point) => actual value

Y200 ; Y-axis is traversed by 0.2 mm (200 without point) => integer value

If the addresses programmed in the block are used as transfer parameters of cycles, then the programmed values always exist as real values in the $C_x variables. For integer values, one cannot take recourse to the programming method (real/integer) in the cycles any more, and therefore there is no evaluation of the programmed values with the correct conversion factor.

There are two system variables $C_TYP_PROG. $C_TYP_PROG for information as to whether REAL or INTEGER programming was undertaken. The structure is the same as that of $C_ALL_PROG and $C_INC_PROG. If the value is programmed as INTEGER, then Bit is set to 0, for REAL it is set to 1. If the value is programmed over a variable $<Number>, then the corresponding bit is also set to 1.

Example:

P1234 A100. X100 -> $C_TYP_PROG == 1.

Only Bit 0 is present, because only A was programmed as REAL.

P1234 A100. C20. X100 -> $C_TYP_PROG == 5.

Bit 1 and Bit 3 (A and C) are present.

Restrictions:

A maximum of ten I, J, K parameters can be programmed in each block. Only one bit each is provided for I, J, K in the variable $C_TYP_PROG. Hence in $C_TYP_PROG the corresponding bit for I, J and K is always set to 0. Therefore it cannot be derived whether I, J or K is programmed as REAL or as INTEGER.

Modal call (G66, G67)

A modal macro program is called with G66. The specified macro program is executed only if the specified conditions are fulfilled.

- The modal macro program is activated on specifying "G66 P... L... <Parameters>;". The transfer parameters are handled as in G65.
- G66 is deselected by G67.

Table 4- 6 Modal call conditions

Call conditions	Function for mode selection	Function for mode deselection
after executing a traversing command	G66	G67

Specification of a parameter

The transfer parameters are defined by programming an Address A - Z.

Interrelation between address- and system variables

Table 4- 7 Interrelation between addresses and variables and addresses that can be used to call commands

Interrelation between addresses and variables	
Address	System variable
A	$C_A
B	$C_B
C	$C_C
D	$C_D
E	$C_E
F	$C_F
H	$C_H
I	$C_I[0]
J	$C_J[0]
K	$C_K[0]
M	$C_M
Q	$C_Q
R	$C_R
S	$C_S
T	$C_T
U	$C_U
V	$C_V
W	$C_W
X	$C_X
Y	$C_Y
Z	$C_Z

Interrelation between address- and system variables

To be able to use I, J and K, these must be specified in the I, J, K sequence.

As the I, J and K addresses in a block containing a macro call can be programmed up to 10 times, access to the system variables within the macro program for these addresses must take place with an index. The syntax of these three system variables thus is $C_I[..], $C_J[..], $C_K[..]. The corresponding values are saved in the matrix in the sequence in which they were programmed. The number of I, J, K addresses programmed in the block is saved in the variables $C_I_NUM, $C_J_NUM and $C_K_NUM.

Unlike for the remaining variables, one index must always be specified while reading the three variables. The index "0" is always used for cycle calls (e.g. G81), for example, N100 R10 = $C_I[0]

Table 4- 8 Interrelation between addresses and variables and addresses that can be used to call commands

Interrelation between addresses and variables	
Address	System variable
A	$C_A
B	$C_B
C	$C_C
I1	$C_I[0]
J1	$C_J[0]
K1	$C_K[0]
I2	$C_I[1]
J2	$C_J[1]
K2	$C_K[1]
I3	$C_I[2]
J3	$C_J[2]
K3	$C_K[2]
I4	$C_I[3]
J4	$C_J[3]
K4	$C_K[3]
I5	$C_I[4]
J5	$C_J[4]
K5	$C_K[4]
I6	$C_I[5]
J6	$C_J[5]
K6	$C_K[5]
I7	$C_I[6]
J7	$C_J[6]
K7	$C_K[6]
I8	$C_I[7]
J8	$C_J[7]
K8	$C_K[7]

Interrelation between addresses and variables	
I9	$C_I[8]
J9	$C_J[8]
K9	$C_K[8]
I10	$C_I[9]
J10	$C_J[9]
K10	$C_K[9]

Note

If more than one block of I, J or K addresses are specified, then the sequence of the addresses for each block of I/J/K is determined in such a way that the numbers of the variables are defined according to their sequence.

Example of entering a parameter

The value of the parameter contains a sign and a decimal point, independently of the address.

The value of the parameters is always saved as actual value.

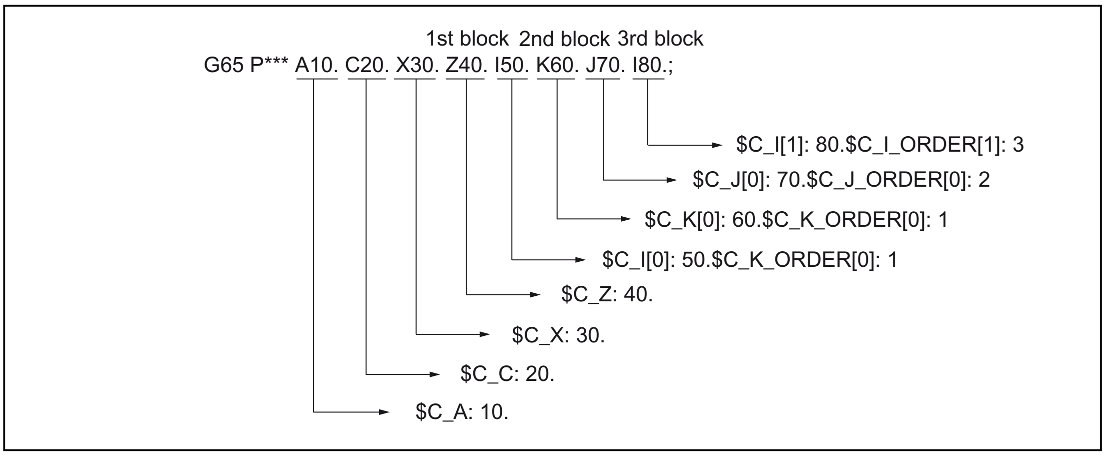

1st block 2nd block 3rd block
G65 P*** A10. C20. X30. Z40. I50. K60. J70. I80.;
$C_I[1]: 80.$C_I_ORDER[1]: 3
$C_J[0]: 70.$C_J_ORDER[0]: 2
$C_K[0]: 60.$C_K_ORDER[0]: 1
$C_I[0]: 50.$C_K_ORDER[0]: 1
$C_Z: 40.
$C_X: 30.
$C_C: 20.
$C_A: 10.

Figure 4-23 Example of entering an argument

Execution of macro programs in the Siemens and ISO modes

A called macro program can be called either in the Siemens mode or in the ISO mode. The language mode in which the program is executed is defined in the first block of the macro program.

If a PROC <Program name> instruction exists in the first block of a macro program, then an automatic changeover to the Siemens mode is conducted. If this instruction is missing, the processing is done in the ISO mode.

The transfer parameters can be saved in local variables by executing a program in the Siemens mode. In the ISO mode however, it is not possible to store transfer parameters in local variables.

To read transfer parameters in a macro program executed in the ISO mode, one must first change over to the Siemens mode with the G290 command.

Examples

Main program with 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
```

Tool Macro program in the Siemens mode:

```
_N_0010_SPF:

PROC 0010 ; Changeover to the Siemens mode

N10 DEF REAL X_AXIS ,Y_AXIS, S_SPEED, FEED

N15 X_AXIS = $C_X Y_AXIS = $C_Y S_SPEED = $C_S FEED = $C_F

N20 G01 F=FEED G95 S=S_SPEED

...

N80 M17
```

Macro program in the ISO mode:

```
_N_0010_SPF:

G290; Changeover to the Siemens mode,

    ; to read the transfer parameters

N15 X_AXIS = $C_X Y_AXIS = $C_Y S_SPEED = $C_S FEED = $C_F

N20 G01 F=$C_F G95 S=$C_S

N10 G1 X=$C_X Y=$C_Y

G291; Changeover to the ISO mode,

N15 M3 G54 T1

N20

...

N80 M99
```

4.7.3 Macro call via G function

Macro call

A macro can be called with a G number analogous to G65.

The replacement of 50 G functions can be configured via 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 stored in the $C_Variables. The number of macro repetitions is programmed with Address L. The number of the programmed G macros is stored in the variable $C_G. All the other G functions programmed in the block are treated as normal G functions. The programming sequence of the addresses and G functions in the block is random, and it does not have any effect on the functionality.

Further information about the parameters programmed in this block is available in Chapter "Macro Program Call (G65, G66, G67)".

Restrictions

- The macro call with a G function can be executed only in the ISO mode (G290).
- Only one G function can be replaced per part program line (or in general, only one subroutine call). If there are possible conflicts with other subroutine calls, e.g. if a modal subroutine is active, the system outputs Alarm 12722 "Several ISO_M/T macro- or cycle calls in block".
- No other G or M macro or M subroutine can be called if a G macro is active. In this case, M macros or M subroutines are executed as M functions. G macros are executed as G functions, provided a corresponding G function exists; otherwise Alarm 12470 "Unknown G function" is output.
- Otherwise the same restrictions are applicable as for G65.

Configuration examples

Calling the subroutine G21_MAKRO via G function G21

```
$MN_EXTERN_G_NO_MAC_CYCLE[0] = 21
$MN_EXTERN_G_NO_MAC_CYCLE_NAME[0] = "G21_MAKRO"
$MN_EXTERN_G_NO_MAC_CYCLE[1] = 123
$MN_EXTERN_G_NO_MAC_CYCLE_NAME[1] = "G123_MAKRO"
$MN_EXTERN_G_NO_MAC_CYCLE[2] = 421
$MN_EXTERN_G_NO_MAC_CYCLE_NAME[2] = "G123_MAKRO"
```

Programming example

```
PROC MAIN
. . .
N0090 G291                                    ; ISO mode
N0100 G1 G21 X10 Y20 F1000 G90                ; Call of G21_MAKRO.spf,
                                              ; G1 and G90 are activated
                                              ; before the call of
                                              ; G21_MAKRO.spf
. . .
N0500 G90 X20 Y30 G123 G1 G54                 ; Call of G123_MAKRO.spf,
                                              ; G1, G54 and G90 are activated
                                              ; before the call of
                                              ; G123_MAKRO.spf
. . .
N0800 G90 X20 Y30 G421 G1 G54                 ; Call of G421_MAKRO.spf,
                                              ; G1, G54 and G90 are activated
                                              ; before the call of
                                              ; G123_MAKRO.spf
. . .
N0900 M30
PROC G21_MAKRO
. . .
N0010 R10 = R10 + 11.11
N0020 IF $C_X_PROG == 0
N0030 SETAL(61000)                            ; programmed variable not transferred
                                              ; correctly
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->activate metric measuring
                                              ; system (no macro call)
N0140 G290
. . .
N0150 M17
PROC G123_MAKRO
. . .
N0010 R10 = R10 + 11.11
N0020 IF $C_G == 421 GOTOF label_G421         ; Macro functionality for G123
N0040 G91 X=$C_X Y=$C_Y F500
. . .
. . .
N1990 GOTOF label_end
N2000 label_G421:                              ; Macro functionality for G421
N2010 G90 X=$C_X
Y=$C_Y F100
N2020
. . .
. . .
N3000 G291
N3010 G123                                     ; Alarm 12470, because G123 is not a
                                               ; G function and a
                                               ; macro call is not possible for
                                               ; active macro
                                               ;
                                               ; Exception: The macro was called
                                               ; as subroutine with CALL
                                                 G123_MAKRO.
N4000 label_end: G290
N4010 M17
```

4.8 Special functions

4.8.1 Contour repetition (G72.1, G72.2)

A contour programmed once can be repeated easily with G72.1 and G72.2. This function can be used to create either a linear copy (G72.2) or a rotational copy (G72.1).

Format

G72.1 X... Y... (Z...) P... L... R...

X, Y, Z: Reference point for coordinate rotation

P: Subprogram number

L: Number of subprogram repetitions

R: Roll angle

A subprogram containing the contour to be copied can be called multiple times with G72.1. The coordinate system is rotated by a certain angle before calling each subprogram. The coordinate rotation is executed around a vertical axis on the selected plane.

G72.2 I... J... K... P... L...

I, J, K: Position to which the X, Y Z axes are traversed before calling the subprogram.

P: Subprogram number

L: Number of subprogram repetitions

A subprogram containing the contour to be repeated can be called multiple times with G72.2. The axes programmed with I, J and K must be traversed incrementally before each subprogram call. The cycle (CYCLE3721) is used to call the subprogram as often as is specified in address "L". A distance programmed in I, J and K and calculated from the starting point is traversed before each subprogram call.

Examples

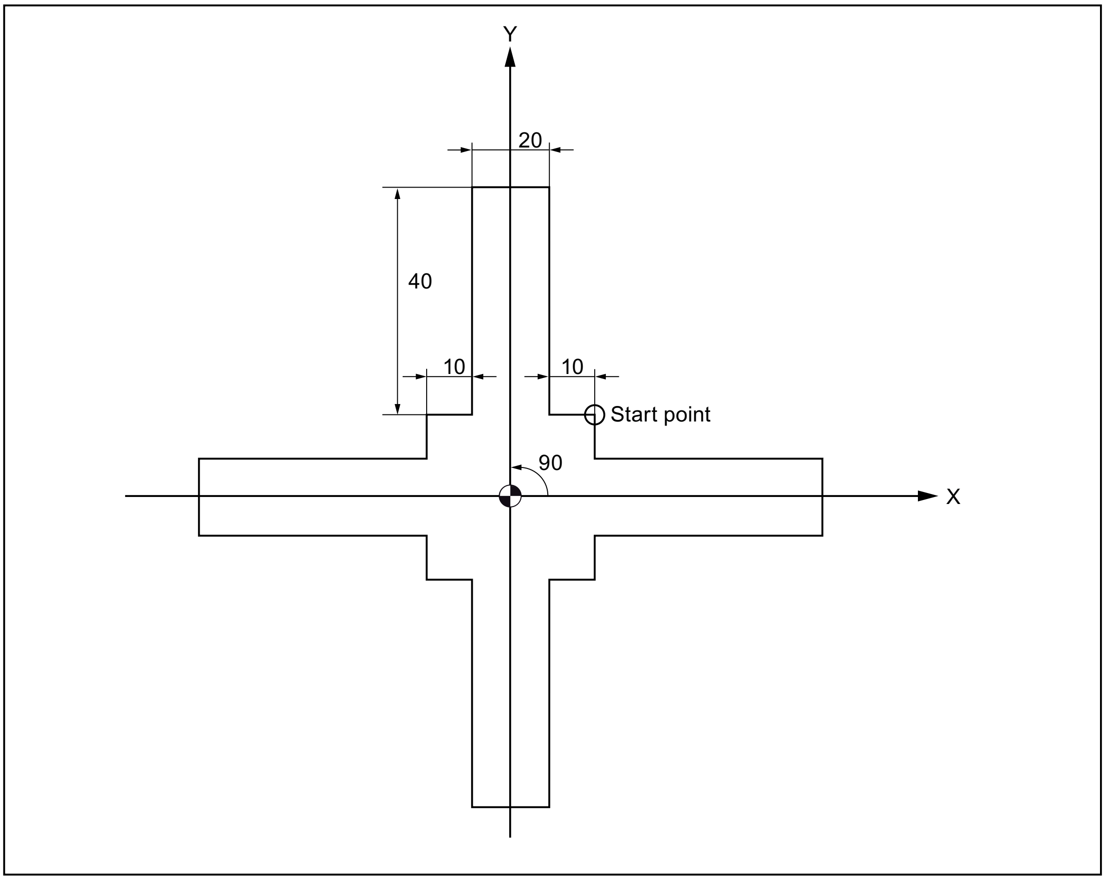


Figure 4-24 Contour repetition with G72.1

Main program

```
N10 G92 X40.0 Y50.0

N20 G01 G90 G17 G41 20 Y20 G43H99 F1000

N30 G72.1 P123 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
```

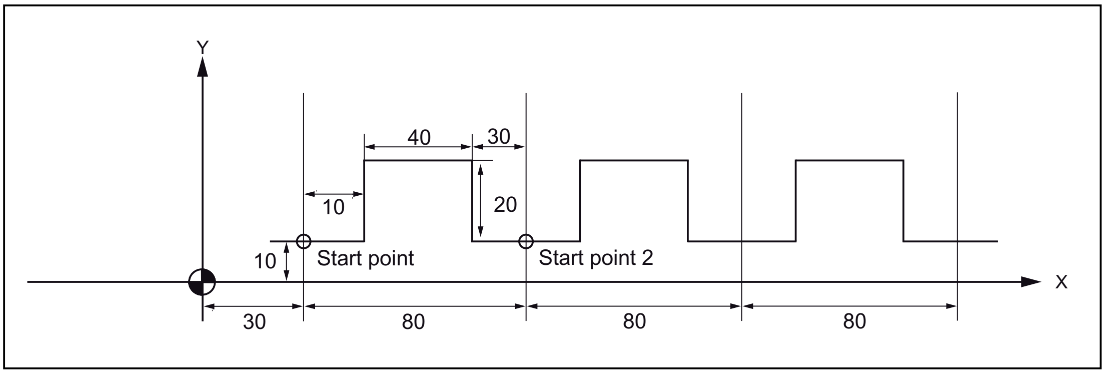


Figure 4-25 Contour repetition with G72.2

Main program

```
N10 G00 G90 X0 Y0

N20 G01 G17 G41 X30. Y0 G43H99 F1000

N30 Y10.

N40 X30.

N50 G72.2 P2000 L3 I80. J0
```

Subprogram 2000.mpf

```
G90 G01 X40.

N100 Y30.

N200 G01 X80.

N300 G01 Y10.

N400 X110.

500 M99
```

4.8.2 Switchover modes for DryRun and skip levels

Changeover of the skip levels (DB21.DBB2) always represents an intervention in the program run, which has led to a short-term drop in velocity on the path. The same is true of the changeover of the DryRun mode (DryRun = dry run feedrate DB21.DBB0.BIT6) from DryRunOff to DryRunOn or vice-versa.

All the drops in velocity can be avoided with a changeover mode that is limited in its function.

No drop in velocity is required with a setting machine data 10706 $MN_SLASH_MASK==2 while changing the skip levels (i.e., a new value in the PLC->NCK-Chan interface DB21.DBB2).

Note

The NCL processes blocks in two steps, the preprocessing and main runs (also pre-travel and main run). The result of the premachining changes to the preprocessing memory. The main machining takes the relevant oldest block out of the preprocessing memory and traverses its geometry.

Note

Changing the skip levels

The premachining is changeover with the setting machine data $MN_SLASH_MASK==2 during a change of the skip level! All blocks located in the preprocessing memory are traversed with the old skip level. The user normally does not have any control over the fill level of the preprocessing memory. The user can see the following effect: A new skip level is effective "some time" after the changeover!

Note

The part program command STOPRE vacates the preprocessing memory. If one switches the skip level before STOPRE, then all the blocks after STOPRE are changed over securely. The same is valid for an implicit STOPRE.

No drop in velocity is required while changing the DryRun mode with the setting machine data 10704 $MN_DRYRUN_MASK==2. Here too, only the premachining that leads to the above-mentioned restrictions is switched. The following analogy is apparent from this: Notice! This will also be active "sometime" after the changeover of the DryRun mode!

Abbreviations

A

ASCII	American Standard Code for Information Interchange
ASUB	Asynchronous subprogram
AV	Job planning
BA	Operating mode
BCD	Binary Coded Decimals
BCS	Basic coordinate system
BIN	Binary files
BP	Basic program
C Bus	Communication bus
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CNC	Computerized Numerical Control
COM	Communication
COR	Coordinate rotation
CPU	Central Processing Unit
CR	Carriage Return
CRC	Cutter radius compensation (tool radius compensation)
CSF	Function plan (PLC programming method)
CTS	Clear To Send (clear to send messages for serial data interfaces)

CUTOM	Cutter radius Compensation: Tool radius compensation
DB	Data Block in the PLC
DBB	Data Block Byte in the PLC
DBW	Data Block Word in the PLC
DBX	Data block bit in the PLC
DC	Direct Control: Movement of the rotary axis via the shortest path to the absolute position within one revolution.
DCE	Data transmission equipment
DDE	Dynamic Data Exchange
DIO	Data Input/Output
DIR	Directory
DLL	Dynamic Link Library: Module which a program can access at runtime. Often contains program sections that are required by different programs.
DOE	Data transmission equipment
DOS	Disk Operating System
DPM	Dual-Port Memory
DPR	Dual-Port RAM: Dual port write/read-only memory
DRAM	Dynamic Random Access Memory: Dynamic write/read-only memory
DRF	Differential Resolver Function (handwheel)
DRY	Dry Run: Dry run feedrate
DSB	Decoding Single Block
DTE	Data Terminal Equipment
DW	Data word

EIA code	Special punched-tape code, number of holes per character always odd
ENC	Encoder: Actual value encoder
EPROM	Erasable Programmable Read Only Memory
FB	Function block
FC	Function Call: Function block in the PLC
FDB	Product database
FDD	Floppy Disk Drive: Diskette drive
FDD	Feed drive
FEPROM	Flash-EPROM: Read and write memory
FIFO	First In First Out: Memory that works without address specification and whose data is read in the same order in which it is stored.
FM	Function module
FM-NC	Function Module - Numerical Control
FPU	Floating Point Unit
FRA	Frame block
FRAME	Data block (FRAME)
FST	Feed Stop
GUD	Global User Data
Hardware	Hardware
HD	Hard Disk
HEX	Abbreviation for hexadecimal number
HMI	Human Machine Interface: Operator functionality of SINUMERIK for operation, programming and simulation.

I	Input
I/O	Input/output
IBN	Commissioning
ICA	Interpolatory Compensation
IF	Pulse enable for drive modules
IK (GD)	Implicit communication (global data)
IM	Interface Module
IMR	Interface Module Receive: Interface module for receiving data
IMS	Interface Module Send: Interface module for transmit mode
INC	Increment
INI	Initializing Data
IPO	Interpolator
IS	Interface Signal
ISO code	Special punched-tape code, number of holes per character always even
JOG	Jogging: Setup mode
K1 .. K4	Channel 1 to channel 4
Kv	Loop gain factor
LAD	Ladder logic (PLC programming method)
LEC	Leadscrew error compensation
LR	Position controller
LUD	Local User Data
MB	Megabyte

MC	Measuring circuit
MCP	Machine Control Panel
MCS	Machine coordinate system
MD	Machine data
MDA	Manual Data Automatic: Manual input
Mode group	Mode group
MPF	Main Program File: NC part program (main program)
MSD	Main Spindle Drive
NC	Numerical Control
NCK	Numerical Control Kernel: with block preparation, traversing range, etc.
NCU	Numerical Control Unit: Hardware unit of NCK
NURBS	Non-Uniform Rational B-Spline: Rational B-spline curves
O	Output
OB	Organization block in the PLC
OEM	Original Equipment Manufacturer: Manufacturer whose products are marketed under an external company name.
OP	Operator Panel: Operating equipment
OPI	Operator Panel Interface: Operator panel front connection
P-Bus	Peripheral Bus
PC	Personal Computer
PCIN	Name of the SW for data exchange with the control
PCMCIA	Personal Computer Memory Card International Association: Memory plug-in card standardization

PG	Programming device
PLC	Programmable Logic Control: Interface control
PMS	Position measuring system
RAM	Random Access Memory: Data memory that can be read and written to
RD	Line Feed
REF	"Reference point approach" function
REPOS	"Reposition" function
ROV	Rapid override: Rapid traverse override
RPA	R Parameter Active: Memory area in the NCK for R-NCK for R parameter numbers
RPY	Roll Pitch Yaw: Rotation type of a coordinate system
RTS	Clear To Send (clear to send message for serial data interfaces, activate sending part, control signal from serial data interfaces)
SBL	Single Block
SD	Setting data
SDB	System Data Block
SEA	Setting Data Active: Identifier (file type) for setting data
SFB	System function block
SFC	System function call
SK	Softkey
SKP	Skip block
SM	Stepper motor
SPF	Sub Routine File: Subprogram

SRAM	Static read-only memory (battery buffered)
SRT	Transformation ratio
SS	Interface signal
SSI	Serial Synchronous Interface
STL	Statement list
SW	Software
SYF	System Files
T	Tool
TC	Tool change
TEA	Testing Data Active: Designation for machine data
TLC	Tool length compensation
TNRC	Tool nose radius compensation
TO	Tool Offset:
TO	Tool change
TOA	Tool Offset Active: Identifier (file type) for tool offsets
TRANSMIT	Transform Milling Into Turning: Coordinate conversion in turning machine for milling
TRC	Tool radius compensation
UFR	User Frame: Work offset
UI	User interface
UP	Subprogram
V.24	Serial interface (definition of the interchange lines between DTE and DCE)
WCS	Workpiece coordinate system

WOP	Workshop-oriented programming
WPD	Workpiece Directory
ZO	Work offset
ZOA	Zero Offset Active: Identifier (file type) for work offset data

G code table

B

Table B- 1 G code table

G code		Description	System A	System C
Group 1				
G00[1)]	1	Rapid traverse	G00	G00
G01	2	Linear movement	G01	G01
G02	3	Circle/helix in clockwise direction	G02	G02
G02.2	6	Involute in clockwise direction		
G03	4	Circle/helix in counterclockwise direction	G03	G03
G03.2	7	Counterclockwise involutes		
G33	5	Thread cutting with constant lead	G32	G33
Group 2				
G17[1)]	1	XY plane		
G18	2	ZX plane		
G19	3	YZ plane		
Group 3				
G90[1)]	1	Absolute programming		
G91	2	Incremental programming		
Group 4				
G22	1	Working area limitation, protection zone 3 on	G22	G22
G23[1)]	2	Working area limitation, protection zone 3 off	G23	G23
Group 5				
G93	3	Inverse-time feedrate (rpm)		
G94[1)]	1	Feedrate in [mm/min, inch/min]	G98	G94
G95	2	Revolutional feedrate in [mm/rev, inch/rev]	G99	G95
Group 6				
G20[1)]	1	Inch input system	G20	G70
G21	2	Metric input system	G21	G71
Group 7				
G40[1)]	1	Deselection of cutter radius compensation	G40	G40
G41	2	Compensation left of contour	G41	G41
G42	3	Compensation to right of contour	G42	G42
Group 8				
G43	1	Positive tool length compensation on		
G44	2	Negative tool length compensation on		
G49[1)]	3	Tool length compensation off		

G code		Description	System A	System C
Group 9				
G73	1	Deep-hole drilling cycle with chip breakage	G73	G75
G74	2	Left tapping cycle	G74	G76
G76	3	Fine drill cycle	G76	G78
G80 1)	4	Cycle off	G80	G80
G81	5	Drilling cycle counterboring		
G82	6	Countersink drilling cycle		
G83	7	Deep-hole drilling cycle with chip removal	G83	G83
G84	8	Right tapping cycle	G84	G84
G85	9	Drilling cycle	G85	G85
G86	10	Drilling cycle, retraction with G00		
G87	11	Reverse countersinking	G87	G87
G89	12	Drilling cycle, retraction with machining feedrate	G89	G89
Group 10				
G98 1)	1	Return to starting point in fixed cycles	not	G98
G99	2	Return to point R in fixed cycles	not	G99
Group 11				
G50 1)2)	1	Scaling off		
G51 2)	2	Scaling on		
Group 12				
G66 2)	1	Macro module call	G66	G66
G67 1)2)	2	Delete macro module call	G67	G67
Group 13				
G96	1	Constant cutting rate on		
G97 1)	2	Constant cutting rate off		
Group 14				
G54 1)	1	Selecting work offset	G54	G54
G55	2	Selecting work offset	G55	G55
G56	3	Selecting work offset	G56	G56
G57	4	Selecting work offset	G57	G57
G58	5	Selecting work offset	G58	G58
G59	6	Selecting work offset	G59	G59
G54P{1...48}	1	Extended work offsets		
G54.1	7	Extended work offset	G54.1	G54.1
G54 P0	1	External work offset		
Group 15				
G61	1	Exact stop modal		
G62	4	Automatic corner override		
G63	2	Tapping mode		
G64 1)	3	Continuous-path mode		

G code		Description	System A	System C
Group 16				
G68 2)	1	Rotation ON, 2D/3D	G68	G68
G69 2)	2	Rotation OFF	G69	G69
Group 17				
G15 1)	1	Polar coordinates off		
G16	2	Polar coordinates on		
Group 18 (non-modal effective)				
G04	1	Dwell time in [s] or spindle revolutions	G04	G04
G05	18	High-speed cycle cutting		
G05.1 2)	22	High-speed cycle -> Call CYCLE305	G05.1	G05.1
G07.1 2)	16	Cylindrical interpolation		
G08	12	Feedforward control ON/OFF		-
G09	2	Exact stop		
G10 2)	3	Write work offset / tool offset	G10	G10
G10.6	17	Retraction from contour (POLF)		
G11	4	End parameter entry		
G27	13	Checking the reference position	G27	G27
G28	5	1st approach reference point	G28	G28
G30	6	2nd/3rd/4th approach reference point	G30	G30
G30.1	19	Reference point position	G30.1	G30.1
G31	7	Measuring with touch-trigger probe	G31	G31
G52	8	Programmable work offset	G52	G52
G53	9	Approach position in machine coordinate system	x	x
G60	22	Directed positioning	x	x
G65 2)	10	Macro call	G65	G65
G72.1 2)	14	Contour repetition with rotation		-
G72.2 2)	15	Linear contour repetition		-
G92	11	Setting actual value, spindle speed limitation		x
G92.1	21	Delete actual value, reset the WKS		
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 OFF	G13.1	G13.1
G12.1	2	Polar coordinates interpolation ON	G12.1	G12.1
Group 31				
G290 1)	1	Selection of Siemens mode	x	x
G291	2	Selection of ISO dialect mode	x	x
x means G code can be used, -- means G code cannot be used				

Note

In general, the G functions mentioned in [1)] are defined by the NC during activation of the control system or during RESET. Data about the actual settings can be found in the documentation of your machine manufacturer.

The G functions mentioned in [2)] are optional. Whether the relevant function is available on your control system can be found out from the documentation of your machine manufacturer.

Data description C

C.1 General machine data

Note

All the machine data described here relates to the SINUMERIK 840D sl. Please use the associated List Manuals for the SINUMERIK 828D control system.

10604	WALIM_GEOAX_CHANGE_MODE	
SD Number	Working area limitation during switchover of geometry axes	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Changes effective after POWER ON	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	The machine data is used to define whether a possibly active working field limitation is retained or deactivated during a geo axis interchange. The MD is bit-coded with following significance: Bit 0=0: Working field limitation is deactivated during geo axis interchange Bit 0=1: Active working area limitation remains active during geo axis interchange	

10615	NCBFRAME_POWERON_MASK	
MD number	Reset global basic frames after power on	
Default setting: 0	Min. input limit: 0	Max. input limit: 0
Changes effective after POWER ON	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	This machine data is used to define whether global basic frames are reset in the data management during power on. This means - offsets are set to 0. - scaling is set to 1. - mirroring is switched off. The individual basic frames can be selected separately. Bit 0 corresponds to basic frame 0, bit 1 to basic frame 1, etc. 0: Basic frame is retained at Power On 1: Basic frame is deleted at Power On.	
Corresponds to:	MD24004 CHBFRAME_POWERON_MASK	

10652	CONTOUR_DEF_ANGLE_NAME	
MD number	Settable name for an angle in the contour brief description	
Default setting: "ANG"	Min. input limit: -	Max. input limit: -
Changes effective after POWER ON	Protection level: 0/0	Unit: -
Data type: STRING		
Meaning:	Identifier for contour angle The identifier must be selected ensuring that there is no conflict with other identifiers (e.g. axes, Euler angle, normal vector, direction vector, coordinate of intermediate point).	

10654	RADIUS_NAME	
MD number	Adjustable name for radius non-modal in the contour brief description	
Default setting: "RND"	Min. input limit: -	Max. input limit: -
Changes effective after POWER ON	Protection level: 0/0	Unit: -
Data type: STRING		
Meaning:	Identifier for contour definition radius. The identifier must be selected ensuring that there is no conflict with other identifiers (e.g. axes, Euler angle, normal vector, direction vector, coordinate of intermediate point).	

10656	CHAMFER_NAME	
MD number	Adjustable name for chamfer in the contour brief description	
Default setting: "CHR"	Min. input limit: -	Max. input limit: -
Changes effective after POWER ON	Protection level: 0/0	Unit: -
Data type: STRING		
Meaning:	Identifier for contour definition chamfer The identifier must be selected ensuring that there is no conflict with other identifiers (e.g. axes, Euler angle, normal vector, direction vector, coordinate of intermediate point).	

10704	DRYRUN_MASK	
MD number	Activation of dry run feedrate	
Default setting: 0	Min. input limit: 0	Max. input limit: 2
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	DRYRUN_MASK = 0 DRYRUN should be switched on and off only at end of block. If DRYRUN_MASK = 1 , the dry run feedrate can also be activated during execution of the program (in the part program block). Notice: The axes are stopped for the duration of the reorganization operation after the dry run feedrate is activated. DRYRUN_MASK = 2 DRYRUN can be switched on and off in each phase, and the axes are not stopped. Notice: However, the function only becomes effective with a block which comes "later" in the program sequence and that occurs with the next (implicit) StopRe block.	
Corresponds to:	SD42100 DRY_RUN_FEED	

10706	SLASH_MASK	
MD number	Activation of block skip function	
Default setting: 0	Min. input limit: 0	Max. input limit: 2
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	When SLASH_MASK = 0, the block skip function can only be activated at the end of a block. In SLASH_MASK = 1, the activation of the block skip function will also be possible during program processing. Notice: The axes are stopped for the duration of the reorganization operation after block skip function. SLASH_MASK = 2 Block switchover is possible in each phase. Notice: The function only becomes effective with a block which comes "later" in the program run! The function is effective from the next (implicit) stop reset block.	

Note

The number of field elements of machine data 10715 $MN_M_NO_FCT_CYCLE[ ],

10716 $MN_M_NO_FCT_CYCLE_NAME[ ], 10814 $MN_EXTERN_M_NO_MAC_CYCLE[ ],

10815 $MN_EXTERN_M_NO_MAC_CYCLE_NAME[ ] has been increased from 10 to 30. In this way,

it is possible to assign 30 M functions to a subprogram call.

10715	M_NO_FCT_CYCLE[0]	
MD number	M function to be replaced by a subprogram	
Default setting: -1	Min. input limit: --	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	M number with which the subprogram is called. The name of the subprogram is stored in MD10716 $MN_M_NO_FCT_CYCLE_NAME[n]. If the M function defined using $MN_M_NO_FCT_CYCLE[n] is programmed in a part program, then the subprogram defined in M_NO_FCT_CYCLE_NAME[n] is started at the end of block. If the M function is reprogrammed in the subprogram, then there is no longer any replacement by a subprogram call. $MN_M_NO_FCT_CYCLE[n] acts in the Siemens mode G290 as well as in the external language mode G291. Restrictions: The subprograms configured with MD10716 $MN_M_NO_FCT_CYCLE_NAME[n] and MD10717 $MN_T_NO_FCT_CYCLE_NAME[ ] must not be effective in one block (part-program line) at the same time, i.e. max. one M/T function replacement can be effective per block. Neither an M98 nor a modal subprogram call may be programmed in the block with the M function replacement. A subprogram return jump or end of part program is also not permitted. Alarm 14016 is generated in the case of a conflict. A subprogram call must not be superimposed on M functions with predetermined significance. Alarm 4150 is generated in the event of a conflict: The following M functions are checked: M0 to M5, M17, M30, M19, M40 to M45, M functions for switching over spindle/axis mode according to MD20094 $MC_SPIND_RIGID_TAPPING_M_NR (default M70). M functions for nibbling/punching that were activated via MD26008 $MC_NIBBLE_PUNCH_CODE according to configuration. For applied external language (MD18800 $MN_MM_EXTERN_LANGUAGE) also M19, M96 to M99. Exception: The M functions defined by MD22560 $MC_TOOL_CHANGE_M_CODE for the tool change.	

10716	M_NO_FCT_CYCLE_NAME[0]	
MD number	Subprogram name for M function replacement	
Default setting: -	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: STRING		
Meaning:	The cycle name is stored in the machine data. This cycle is called if the M function was programmed from MD10715 $MN_M_NO_FCT_CYCLE. If the M function is programmed in a motion block, then the cycle is executed after the motion. MD10715 $MN_M_NO_FCT_CYCLE acts in the Siemens mode G290 as well as in the external language mode G291. If a T number is programmed in the calling block, the programmed T number can be queried in the cycle under the variable $P_TOOL. M and T function replacement must not be programmed in one block at the same time, i.e. max. one M/T function replacement can be effective per block. Neither an M98 nor a modal subprogram call may be programmed in the block with the M function replacement. A subprogram return jump or end of part program is also not permitted. Alarm 14016 is generated in the case of a conflict.	
Corresponds to..	MD10715 $MN_M_NO_FCT_CYCLE MD10717 $MN_T_NO_FCT_CYCLE_NAME	

10717	T_NO_FCT_CYCLE_NAME	
MD number	Name of the tool change cycle for T function replacement	
Default setting: -	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: STRING		
Meaning:	Cycle name of the tool change routine during call via T function. If a T function is programmed in a part program block, then the subprogram defined in T_NO_FCT_CYCLE_NAME is called at the end of the block. The programmed T number can be queried in the cycle via the system variables $C_T/$C_T_PROG as decimal value and via $C_TS/$C_TS_PROG as string (only in tool management). MD10717 $MN_T_NO_FCT_CYCLE_NAME acts in the Siemens mode G290 as well as in the external language mode G291. The subprograms configured with MD10716 $MN_M_NO_FCT_CYCLE_NAME and MD10717 $MN_T_NO_FCT_CYCLE_NAME must not be effective in one block at the same time, i.e. max. one M/T function replacement can be effective per block. Neither an M98 nor a modal subprogram call may be programmed in the block with the T function replacement. It is also illegal to program a subprogram return jump or end of part program. Alarm 14016 is generated in the case of a conflict.	
Corresponds to:	MD10715 $MN_M_NO_FCT_CYCLE MD10717 $MN_T_NO_FCT_CYCLE_NAME	

10718	M_NO_FCT_CYCLE_PAR	
MD number	M function replacement with parameters	
Default setting: -1	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	If a M function replacement was configured with MD10715 $MN_M_NO_FCT_CYCLE[n], MD10716 $MN_M_NO_FCT_CYCLE_NAME[n], then MD10718 $MN_M_NO_FCT_CYCLE_PAR can be used for specifying parameter transfer for one of these M functions per system variable as is the case for the T function replacement. The parameters stored in the system variables always refer to the part program line in which the M function to be replaced is programmed. The following system variables are available: $C_ME: Address extension of the substituted M function $C_T_PROG: TRUE if address T has been programmed $C_T: Value of address T (integer) $C_TE: Address extension of address T $C_TS_PROG: TRUE if address TS has been programmed $C_TS: Value of address TS (string, with tool management only) $C_D_PROG: TRUE if address D has been programmed $C_D: Value of address D $C_DL_PROG: TRUE if address DL has been programmed $C_DL: Value of address DL	

10719	T_NO_FCT_CYCLE_MODE	
MD number	Parameter assignment for T function replacement	
Default setting: 0	Min. input limit: 0	Max. input limit: 7
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	Processing of the substitution subprogram is parameterized in this machine data for the tool or tool offset selection. Bit 0 = 0: The D or DL number is transferred to the substitution subprogram (default value) Bit 0 = 1: The D or DL number is not transferred to the substitution subprogram if the following conditions are fulfilled: $MC_TOOL_CHANGE_MODE = 1 programming of D/DL with T or M function with which the tool-changing cycle is called, in one part program line Bit 1 = 0 Processing of the substitution subprogram at end of block (default value) Bit 1 = 1 Processing of the substitution subprogram at start of block Bit 2 = 0: Processing of the substitution subprogram according to setting for bit 1 Bit 2 = 1: Processing of the substitution subprogram at start of block and at end of block	

10760	G53_TOOLCORR	
MD number	Method of functioning with G53, G153 and SUPA	
Default setting: 0	Min. input limit: 0	Max. input limit: 3
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	This machine data is used to define whether tool length and tool radius compensations are to be suppressed in the language commands G53, G153 and SUPA. The machine data is bit-coded. Bit 0 = 0: G53, G153 and SUPA is a non-modal suppression of work offsets. Active tool length and tool radius compensation is retained. Bit 0 = 1: G53, G153 and SUPA is a non-modal suppression of work offsets, active tool length and tool radius compensation. The behavior regarding the tool lengths can be modified with bit 1. Bit 1 is only evaluated if bit 0 has the value 1. Bit 1 = 0: If bit 0 is set, the tool length for G53, G153 and SUPA is always suppressed. Bit 1 = 1: If bit 0 is set, the tool length is only suppressed for G53, G153 and SUPA if a cutting edge is selected but not in the same block (this may even be the cutting edge which is already active).	

10800	EXTERN_CHAN_SYNC_M_NO_MIN	
MD number	1st M function for channel synchronization	
Default setting: -1	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	M number of the first M function with which channel program synchronization can be undertaken in ISO2/3 mode. In order to avoid conflicts with standard M functions, 100 is the smallest value permitted. If a value between 0 - 99 is entered, alarm 4170 is output.	

10802	EXTERN_CHAN_SYNC_M_NO_MAX	
SD Number	Last M number for the channel synchronization	
Default setting: -1	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	M number of the last M function with which channel program synchronization can be undertaken in ISO2/3 mode. Along with MD10800 $MN_EXTERN_CHAN_SYNC_M_NO_MIN, the machine data defines a range of M numbers which is reserved for channel synchronization. The range may be a maximum of 10*number of channels in size because only 10 WAIT marks may be set for each channel. If a value between 0-99 or a value less than MD10800 $MN_EXTERN_CHAN_SYNC_M_NO_MIN is specified, alarm 4170 is output.	

10804	EXTERN_M_NO_SET_INT	
MD number	M function for ASUB activation	
Default setting: 96	Min. input limit: 0	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	M function number with which an interrupt program (ASUB) is activated in the ISO_T/M mode. The interrupt program is always started with the first fast NC input. The M number defined in the machine data takes the place of M96 in the external language mode. For constraints, see MD10715 $MN_M_NO_FCT_CYCLE	
Corresponds to:	MD10814 $MN_EXTERN_M_NO_MAC_CYCLE MD10804 $MN_EXTERN_M_NO_SET_INT MD10806 $MN_EXTERN_M_NO_DISABLE_INT MD10800 $MN_EXTERN_CHAN_SYNC_M_NO_MIN MD10802 $MN_EXTERN_CHAN_SYNC_M_NO_MAX MD20095 $MC_EXTERN_RIGID_TAPPING_M_NR	

10806	EXTERN_M_NO_DISABLE_INT	
MD number	M function for ASUB deactivation	
Default setting: 96	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	M function number with which an interrupt program (ASUB) is deactivated in the ISO_T/M mode. The M number defined in the machine data takes the place of M97 in the external language mode. For constraints, see MD10715 $MN_M_NO_FCT_CYCLE	
Corresponds to:	MD10814 $MN_EXTERN_M_NO_MAC_CYCLE MD10804 $MN_EXTERN_M_NO_SET_INT MD10806 $MN_EXTERN_M_NO_DISABLE_INT MD10800 $MN_EXTERN_CHAN_SYNC_M_NO_MIN MD10802 $MN_EXTERN_CHAN_SYNC_M_NO_MAX MD20095 $MC_EXTERN_RIGID_TAPPING_M_NR	

10808	EXTERN_INTERRUPT_BITS_M96	
MD number	Interrupt program (ASUB)	
Default setting: 0	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	The flow of the interrupt routine activated with M96P can be affected by setting the different bits. Bit 0=0: Interrupt program is not possible, M96/97 are normal M functions Bit 0=1: Activation of an interrupt program with M96/M97 is allowed Bit 1=0: Process part program with the end position of the next block after the interruption block Bit 1=1: Continue processing part program from the interruption position bit 2 =0: The interrupt signal interrupts the current block immediately and starts the interrupt routine Bit 2=1: the interrupt routine is started only at the end of the block Bit 3=0: Interrupt execution cycle during an interrupt signal Bit 3=1: Start interrupt program only at the end of the execution cycle	

10810	EXTERN_MEAS_G31_P_SIGNAL	
MD number	Assignment of the measuring inputs for G31 P..	
Default setting: 1	Min. input limit: 0	Max. input limit: 3
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	The machine data is used to define an assignment of the measuring inputs 1 and 2 to the P number programmed with G31 P1 (-P4). The MD is bit-coded. Only Bit 0 and Bit 1 are evaluated. If, e.g. the Bit 0=1 is in $MN_EXTERN_MEAS_G31_P_SIGNAL[1], it is activated with G31 P2 of the first measuring input. The second measuring input is activated with G31 P4 with $MN_EXTERN_MEAS_G31_P_SIGNAL[3] = 2. Bit 0=0: Do not evaluate measuring input 1 in G31 P1 (-P4) Bit 0=1: Activate measuring input 1 in G31 P1 (-P4) Bit 1=0: Do not evaluate measuring input 2 in G31 P1 (-P4) Bit 1=1: Activate measuring input 2 in G31 P1 (-P4)	

10812	EXTERN_DOUBLE_TURRET_ON	
MD number	Double turret head with G68	
Default setting: FALSE	Min. input limit:	Max. input limit:
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BOOLEAN		
Meaning:	This MD is used to define whether a double-slide machining is to be started with G68 (channel synchronization for the first and second channel), or the second tool of a double turret (= 2, tool permanently connected at the distance defined in setting data 42162 $SC_EXTERN_DOUBLE_TURRET_DIST) is to be activated. FALSE: Channel synchronization for double-slide machining TRUE: Change 2nd tool of a double turret (= activate $SC_EXTERN_DOUBLE_TURRET_DISTANCE as additive work offset and mirroring at Z axis)	

10814	EXTERN_M_NO_MAC_CYCLE	
MD number	Macro call via M function	
Default setting: -1	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	M number with which a macro is called. The name of the subprogram is stored in $MN_EXTERN_M_NO_MAC_CYCLE_NAME[n]. If the M function is programmed in a part program block defined with $MN_EXTERN_M_NO_MAC_CYCLE[n], the subprogram defined in EXTERN_M_NO_MAC_CYCLE_NAME[n] is started, all the addresses programmed in the block are written in the associated variables. If the M function is reprogrammed in the subprogram, then there is no longer any replacement by a subprogram call. $MN_EXTERN_M_NO_MAC_CYCLE_NAME[n] operates only in the external language mode G291. The subprograms configured with $MN_EXTERN_M_NO_MAC_CYCLE_NAME[n] must not be effective in one block (part program line) at the same time, i.e. max. one M function replacement can be effective per block. Neither an M98 nor a modal subprogram call may be programmed in the block with the M function replacement. A subprogram return jump or end of part program is also not permitted. Alarm 14016 is generated if these conventions are not observed. For constraints, see MD10715 $MN_M_NO_FCT_CYCLE.	

10815	EXTERN_M_NO_MAC_CYCLE_NAME	
MD number	Subprogram name for macro call via M function	
Default setting: -	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: STRING		
Meaning:	Name of subprogram which is started during call via the M function defined with $MN_EXTERN_M_NO_MAC_CYCLE[n].	

10818	EXTERN_INTERRUPT_NUM_ASUP		
MD number	Interrupt number for ASUB start (M96)		
Default setting: 1	Min. input limit: 1		Max. input limit: 8
Changes effective after Power On		Protection level: 7/2	Unit: -
Data type: BYTE			
Meaning:	Number of the interrupt input with which an asynchronous subprogram activated in the ISO mode is started. (M96<Program number>)		

10820	EXTERN_INTERRUPT_NUM_RETRAC		
MD number	Interrupt number for rapid retraction (G10.6)		
Default setting: 2	Min. input limit: 1		Max. input limit: 8
Changes effective after Power On		Protection level: 7/2	Unit: -
Data type: BYTE			
Meaning:	Number of the interrupt input with which a rapid retraction is triggered in the ISO mode at the position programmed with G10.6.		

10880	MM_EXTERN_CNC_SYSTEM		
MD number	Definition of the control system to be adapted		
Default setting: 1	Min. input limit: 1		Max. input limit: 3
Changes effective after Power On		Protection level: 7/2	Unit: -
Data type: WORD			
Meaning:	Selecting the external language 1 = ISO_2.1: System Fanuc0 Milling (shared compensation memory) 2 = ISO_3.1: System Fanuc0 Turning (shared compensation memory) 3: External memory via OEM application 4: ISO_2.2: System Fanuc0 Milling (separate compensation memory) 5: ISO_3.2: System Fanuc0 Turning (separate compensation memory)		

10882	NC_USER_EXTERN_GCODES_TAB [n]:0...59	
MD number	List of user-specific G commands of an external NC language	
Default setting: -	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 2/2	Unit: -
Data type: STRING		
Meaning:	List of G commands reconfigured by the user, external NC languages. The implemented G commands can be found in the current Siemens documentation for this programming language. The list should be structured as follows: Even address: G command to be changed Following odd address: New G command Only G codes can be reconfigured, e.g. G20, G71.	

10884	EXTERN_FLOATINGPOINT_PROG	
MD number	Evaluation of programmed values without decimal point	
Default setting: TRUE	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BOOLEAN		
Meaning:	This machine data is used to define how programmed values without decimal points are evaluated. 0: Values without a decimal point are interpreted in internal units, e.g. X1000 = 1 mm (with 0.001 input resolution) X1000.0 = 1000 mm 1: Values without a decimal point are interpreted as mm, inch or degree, e.g. X1000 = 1000 mm, X1000.0 = 1000 mm	

10886	EXTERN_INCREMENT_SYSTEM	
MD number	Increment system	
Default setting: FALSE	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BOOLEAN		
Meaning:	This machine data is effective for external programming languages, i.e. if MD18800 $MN_MM_EXTERN_LANGUAGE = 1. This machine data is used to define which increment system is active: 0: Increment system IS-B= 0.001 mm/degree = 0.0001 inch 1: Increment system IS-C= 0.0001 mm/degree = 0.00001 inch	

10888	EXTERN_DIGITS_TOOL_NO	
MD number	T number in ISO mode	
Default setting: 2	Min. input limit: 0	Max. input limit: 8
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	The machine data is active only in $MN_EXTERN_CNC_SYSTEM = 2. Number of digits in tool number in the programmed T value. The number of leading digits specified as tool number is interpreted with $MN_EXTERN_DIGITS_TOOL_NO from the programmed T value. The following positions address the offset memory. If a value > 0 is entered in MD $MN_EXTERN_DIGITS_OFFSET_NO, MD $MN_EXTERN_DIGITS_TOOL_NO has no effect. $MN_EXTERN_DIGITS_OFFSET_NO has priority over $MN_EXTERN_DIGITS_TOOL_NO.	

18800	MM_EXTERN_LANGUAGE	
MD number	Activation of external NC languages	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	To process part programs of other system manufacturers, one must activate the corresponding NC language. Only one external language is to be selected. The already defined relevant instruction set can be taken from the current documentation. Bit 0 (LSB): Processing of part programs ISO_2 or ISO_3. For encoding, see $MN_MM_EXTERN_CNC_SYSTEM (10880)	

C.2 Channel-specific machine data

20050	AXCONF_GEOAX_ASSIGN_TAB	
MD number	Assignment of geometry axis to channel axis	
Default setting: 1, 2, 3	Min. input limit: 0	Max. input limit: 20
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	The channel axis to which the geometry axis is assigned is specified in this MD. The assignment for all the geometry axes is to be done channel-specifically. If no assignment is defined for a geometry axis, then this geometry axis does not exist and cannot be programmed (with the names defined in AXCONF_GEOAX_NAME_TAB). E.g.: Turning machine without transformation: $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1 ; 1st Geo axis = 1st channel axis $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0 ; 2nd Geo axis not defined $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 2 ; 3rd Geo axis = 2nd channel axis The assignment defined here is valid if no transformation is active. In case of active transformation n, the transformation-specific assignment table TRAFO_GEOAX_ASSIGN_TAB_n is active.	

20060	AXCONF_GEOAX_NAME_TAB	
MD number	Geometry axis name in channel	
Default setting: X, Y, Z	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: STRING		
Meaning:	The names of the geometry axes for the channel are input separately in this MD. Geometry axes can be programmed in the part program with the names specified here. Special situations: - The entered geometry axis name must not collide with the designation and assignment of the machine and channel axis names. - The entered geometry axis name must not overlap with the names for the Euler angle (MD10620: EULER_ANGLE_NAME_TAB), names for the direction vectors (MD10640: DIR_VECTOR_NAME_TAB), names for the coordinates of the intermediate circuit point for CIP (MD10660: INTERMEDIATE_POINT_NAME_TAB) and the names for the interpolation parameters (MD10650: IPO_PARAM_NAME_TAB). - The geometry axis name entered must not use the following reserved address letters: - D tool compensation (D function) - E reserved - F feedrate (F function) - G distance condition - H auxiliary function (H function) - L subprogram call - M additional function (M function) - N subordinate block - P number of subprogram repetitions - R arithmetic parameter - S spindle number (S function) - T tool (T function) - Keywords (e.g. DEF, SPOS, etc.) and predefined identifiers (e.g. ASPLINE, SOFT) are also not permitted. - The use of an axis identifier, consisting of a valid address letter (A, B, C, I, J, K, Q, U, V, W, X, Y, Z), followed by an optional numerical extension (1-99), offers slight benefits in the block cycle time compared to the assignment of a general identifier. - The geometry axes in different channels can have the same names	
Corresponds to:	MD 10000: AXCONF_MACHAX_NAME_TAB MD 20080: AXCONF_CHANAX_NAME_TAB	

20070	AXCONF_MACHAX_USED	
MD number	Machine axis number valid in channel	
Default setting: 1, 2, 3, 4	Min. input limit: 0	Max. input limit: 31
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	The machine axis to which the channel axis / special axis is assigned is input in this MD. The assignment for all the channel axes is to be done channel-specifically. A machine axis that is not assigned to any channel is not active, i.e. the axis control is not processed, the axis is not displayed on the screen and it cannot be programmed in any channel.	

20080	AXCONF_CHANAX_NAME_TAB	
MD number	Name of channel axis in the channel	
Default setting: X, Y, Z, A, B, C, U, V, X11, Y11, ...	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: STRING		
Meaning:	The number of the channel axis / special axis is entered in this MD. Normally the first three channel axes are populated by the three assigned geometry axes (see also MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB). The remaining channel axes are also denoted as special axes. The channel axis / special axis is always displayed on the monitor of WCS (Workpiece Coordinate System) with the names input in this MD. Special situations: - The entered channel axis name / special axis name must not collide with the designation and assignment of the machine and channel axis names. - The entered channel axis name must not overlap with the names for the Euler angle (MD10620 $MN_EULER_ANGLE_NAME_TAB), names for the direction vectors (MD10640 $MN_DIR_VECTOR_NAME_TAB), names for the coordinates of the intermediate circuit point for CIP(MD10660 $MN_INTERMEDIATE_POINT_NAME_TAB) and the names for the interpolation parameters (MD10650 $MN_IPO_PARAM_NAME_TAB). - The entered channel axis name must not have the following reserved address letters: - D tool compensation (D function) - E reserved - F feedrate (F function) - G distance condition - H auxiliary function (H function) - L subprogram call - M additional function (M function) - N subordinate block - P number of subprogram repetitions - R arithmetic parameter - S spindle speed (S function) - T tool (T function) - Keywords (e.g. DEF, SPOS, etc.) and predefined identifiers (e.g. ASPLINE, SOFT) are also not permitted. - The use of an axis identifier, consisting of a valid address letter (A, B, C, I, J, K, Q, U, V, W, X, Y, Z), followed by an optional numerical extension (1-99), offers slight benefits in the block cycle time compared to the assignment of a general identifier. - A separate name must not be entered in this MD for channel axes that are assigned geometry axes (normally the first three channel axes). Illegal axis identifiers are rejected with power-up alarm.	

20094	SPIND_RIGID_TAPPING_M_NR	
MD number	M number for changing over to the controlled spindle mode (Siemens mode)	
Default setting: 70	Min. input limit: 0	Max. input limit: 0xFF
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	This machine data defines the M auxiliary function number with which the spindle is switched to the axis mode. The M number defined in the machine data takes the place of M70 in the Siemens language mode. Note: M70 with the corresponding address extension is always output at the VDI interface as identification for the switch to axis mode. Restrictions: See machine data 10715: $MN_M_NO_FCT_CYCLE	
Corresponds to:	MD10814 $MN_EXTERN_M_NO_MAC_CYCLE, MD10804 $MN_EXTERN_M_NO_SET_INT MD10806 $MN_EXTERN_M_NO_DISABLE_INT, MD 10800 $MN_EXTERN_CHAN_SYNC_M_NO_MIN, MD10802 $MN_EXTERN_CHAN_SYNC_M_NO_MAX MD20095 $MC_EXTERN_RIGID_TAPPING_M_NR	

20095	EXTERN_RIGID_TAPPING_M_NR	
MD number	M function for changing over to the controlled axis mode (external language mode)	
Default setting: 29	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	This machine data defines the M function number which is to be used to switch to the controlled spindle/axis mode. The M number defined in the machine data replaces M29 in the external language mode. Predefined M numbers such as M00, M1, M2, M3, etc. are not permitted as M number. Restrictions: See machine data 10715 $MN_M_NO_FCT_.CYCLE	
Corresponds to:	MD10814 $MN_EXTERN_M_NO_MAC_CYCLE, MD10804 $MN_EXTERN_M_NO_SET_INT MD10806 $MN_EXTERN_M_NO_DISABLE_INT, MD10800 $MN_EXTERN_CHAN_SYNC_M_NO_MIN, MD10802 $MN_EXTERN_CHAN_SYNC_M_NO_MAX MD20095 $MC_EXTERN_RIGID_TAPPING_M_NR	

20150	GCODE_RESET_VALUES	
MD number	Reset G groups	
Default setting: 2, 0, 0, 1, 0, ...	Min. input limit: -	Max. input limit: -
Changes effective after RESET	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	Definition of the G codes that become active during run-up and reset or at part program end and part program start. The G code index must be specified in the respective groups as preset value. Name - Group - Default value: GCODE_RESET_VALUES[0] - Group 1 - Default value 2 (G01) GCODE_RESET_VALUES[1] - Group 2 - Default value 0 (disabled) GCODE_RESET_VALUES[2] - Group 3 - Default value 0 (disabled) GCODE_RESET_VALUES[3] - Group 4 - Default value 1 (FIFO START) GCODE_RESET_VALUES[4] - Group 5 - Default value 0 (disabled) GCODE_RESET_VALUES[5] - Group 6 - Default value 1 (G17) for milling GCODE_RESET_VALUES[6] - Group 7 - Default value 1 (G40) GCODE_RESET_VALUES[7] - Group 8 - Default value 1 (G500) GCODE_RESET_VALUES[8] - Group 9 - Default value 0 (disabled) GCODE_RESET_VALUES[9] - Group 10 - Default value 1 (G60) GCODE_RESET_VALUES[10] - Group 11 - Default value 0 (disabled) GCODE_RESET_VALUES[11] - Group 12 - Default value 1 (G601) GCODE_RESET_VALUES[12] - Group 13 - Default value 2 (G71) GCODE_RESET_VALUES[13] - Group 14 - Default value 1 (G90) GCODE_RESET_VALUES[14] - Group 15 - Default value 2 (G94) GCODE_RESET_VALUES[15] - Group 16 - Default value 1 (CFC) ...	

20152	GCODE_RESET_MODE	
MD number	Reset behavior of G groups	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Changes effective after RESET	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	This Machine data is only evaluated if bit 0 is set in $MC_RESET_MODE_MASK. This MD is used to define for each entry in MD $MN_GCODE_RESET_VALUES (i.e. for each G group), whether a setting according to the $MC_GCODE_RESET_VALUES is taken again for a reset / part program end, (MD = 0) or the momentary current setting is retained (MD = 1). Example: The normal position of the 6th G group (current level) is read from the MD $MC_GCODE_RESET_VALUES for each reset / part program end here: $MC_GCODE_RESET_VALUE(5)=1; reset value of 6th G group is M17 $MC_GCODE_RESET_MODE(5)=0; normal position of the 6th G group after reset / part program end is according to $MC_GCODE_RESET_VALUES(5) However, if the current setting of the 6th G group (current level) is to be retained beyond reset / part program end, then we have the following setting: $MC_GCODE_RESET_VALUE(5)=1; reset value of 6th G group is M17 $MC_GCODE_RESET_MODE(5)=1; current setting of the 6th G group is retained even after reset / part program end	
Corresponds to:	MD20110 $MC_RESET_MODE_MASK MD 20112 $MC_START_MODE_MASK	

20154	**EXTERN_GCODE_RESET_VALUES[n]: 0, ..., 30**	
MD number	Definition of the G codes that are to be activated during reboot, if the NC channel is not running in the Siemens mode.	
Default setting: 1, 1, 1, 2, 1, 1...	Min. input limit: -	Max. input limit: -
Changes effective after RESET	Protection level: 2/2	Unit: -
Data type: BYTE		
Meaning:	When using an external NC programming language, definition of the G codes which are effective during a power up and reset and/or at the end of a part program depending on MD20110 $MC_RESET_MODE_MASK and at the start of a part program depending on MD20112 $MC_START_MODE_MASK The following external programming languages are possible: • ISO dialect milling ISO dialect turning The G group classification to be used is taken from the current SINUMERIK documentation. The following groups can be written within the MD EXTERN_GCODE_RESET_VALUES: **ISO dialect mode:** 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	

20156	EXTERN_GCODE_RESET_MODE	
MD number	Reset behavior of external G groups	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Changes effective after RESET	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	This MD is only evaluated if bit 0 is set in $MC_RESET_MODE_MASK (see there)! This MD is used to define for each entry in MD $MN_EXTERN_GCODE_RESET_VALUES (i.e. for each G group), whether a setting according to the MD $MC_EXTERN_GCODE_RESET_VALUES is taken again for a reset / part program end, (MD=0) or the momentary current setting is retained (MD=1). Example of ISO dialect mode: The normal position of the 14th G group (adjustable work offset) is read from the machine data $MC_EXTERN_GCODE_RESET_VALUES for each reset / part program end here: $MC_EXTERN_GCODE_RESET_VALUES[13]=1; reset value of the 14th group is G54 $MC_EXTERN_GCODE_RESET_VALUES(13)=0; normal position of the 14th G group after reset / part program end is defined via $MC_EXTERN_GCODE_RESET_VALUES[13] However, if the current setting of the 14th G group is to be retained beyond reset/part program end, then we have the following setting: $MC_EXTERN_GCODE_RESET_VALUES[13]=1; reset value of the 14th group is G54 $MC_G_CODE_GCODE_RESET_VALUES(13)=0; current setting of the 14th G group is retained even after reset/part program end	

20380	TOOL_CORR_MODE_G43/G44	
MD number	Handling the tool length compensation G43/G44	
Default setting: 0	Min. input limit: 0	Max. input limit: 2
Changes effective after RESET	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	The machine data is active only for $MN_MM_EXTERN_CNC_LANGUAGE = 1; If G43/G44 is active, it determines how length compensations programmed with H are processed. 0: Mode A The tool length H always operates on the Z axis, independent of the current plane 1: Mode B The tool length H always operates on one of the three geometry axes, depending on the active plane, i.e. in: G17 on the third geometry axis (normally Z) G18 on the second geometry axis (normally Y) G19 on the first geometry axis (as a rule X) Offsets can be installed in all three geometry axes in this mode through multiple programming, i.e. activation of a component does not delete the possibly already effective length compensation in a different axis. 2: Mode C The tool length works independently of the active plane in the axis which was simultaneously programmed with H. Moreover, the behavior is as in Variant B.	

20382	TOOL_CORR_MOVE_MODE	
MD number	Retract the tool length compensation	
Default setting: FALSE	Min. input limit: -	Max. input limit: -
Changes effective after RESET	Protection level: 7/2	Unit: -
Data type: BOOLEAN		
Meaning:	The machine data determines how the tool length compensations are retracted. 0: A tool length component is retracted only if the associated axis was programmed (behavior as in earlier software versions). 1: Too lengths are always retracted immediately irrespective of whether or not the associated axes are programmed.	

20732	EXTERN_G0_LINEAR_MODE	
MD number	Interpolation behavior with G00	
Default setting: TRUE	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BOOLEAN		
Meaning:	This MD is used to define the interpolation behavior for G00. 0: Axes are traversed as positioning axes 1: Mutual axis interpolation	

20734	EXTERN_FUNCTION_MASK	
MD number	Function mask for external language	
Default setting: 0	Min. input limit: 0	Max. input limit: 0xFFFF
Changes effective after RESET	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	Functions in the ISO mode are affected by this machine data. Bit 0=0: ISO Mode T: A" and "C" are interpreted as axes. If contour definition is programmed, there must be a comma before "A" or "C". Bit 0=1: "A" and "C" in the part program are always interpreted as contour definition. No Axis A or C should exist. Bit 1=0: ISO mode T G10 P<100 tool geometry P>100 tool wear Bit 1=1: G10 P<10 000 tool geometry P>10 000 tool wear Bit 2=0: Dwell time G04: Always [s] or [ms] Bit 2=1: If G95 is active, dwell time is in spindle revolutions Bit 3=0: Errors in the ISO scanner lead to alarm Example: N5 G291 ; ISO dialect mode N10 WAIT ; alarm 12080 "WAIT unknown" N15 G91 G500 ; alarm 12080 "G500 unknown" Bit 3=1: Errors in the ISO scanner are not output, the block is forwarded to the Siemens translator Example: N5 G291 ; ISO dialect mode N10 WAIT ; block is processed by Siemens translator N15 G91 G500 ; block is processed by Siemens translator N20 X Y ; block is processed by ISO translator due to G291, G91 from N15 is active Bit 4=0: G00 is traversed to the active exact stop function. Example: In G64, even G00 blocks are traversed with G64 Bit 4=1: G00 blocks are always traversed with G09, even if G64 is active Bit 5=0: Movements of the rotary axis are executed along the shortest path Bit 5=1: Movements of the rotary axis are executed in the positive or negative direction of rotation, depending on the sign Bit 6=0: Four-digit program numbers only permitted Bit 6=1: Eight-digit program number is permitted. If there are less than four digits, the figure is increased four digits	

20734	EXTERN_FUNCTION_MASK
Meaning:	Bit 7=0: Axis programming for geometry axis interchange/parallel axes is compatible with the ISO mode Bit 7=1: Axis programming in geo axis interchange/parallel axes is compatible with the Siemens mode, in the ISO mode Bit 8=0: For cycles, the F value is always transferred interpreted as feedrate Bit 8=1: For threading cycles, the F value is always interpreted as lead Bit 9=0: With ISO mode T for G84, G88 in the standard mode F for G95 is multiplied by 0.01 mm or 0.0001 inch Bit 9=1: In the ISO mode T for G84, G88 and in the standard mode F for G95, it is multiplied with 0.01 mm or 0.0001 inch Bit 10=0: In M96 Pxx, during interrupt, the program programmed with Pxx is always called. Bit 10=1: In M96 Pxx, CUCLE396.spf is always called in case of interrupt Bit 11=0: G54.1 is displayed during the programming of G54 Pxx. Bit 11=1: G54Px is always displayed while programming G54 Pxx or G54.1 Px. Bit 12=0: $P_ISO_STACK is not changed on calling the subprogram defined with M96 Pxx Bit 12=1: $P_ISO_STACK is incremented on calling the subprogram defined with M96 Pxx Bit 13=0: G10 is executed without internal STOPRE Bit 13=1: G10 is executed with internal STOPRE Bit 14 = 0: ISO dialect T: No alarm if a cutting edge has been programmed in the T command. Bit 14 = 1: ISO dialect T: Alarm 14185, if no cutting edge has been programmed in the T command.

22420	FGROUP_DEFAULT_AXES[n]: 0, ..., 7	
MD number	Default value of the FGROUP command	
Default setting: 0	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/7	Unit: -
Data type: BYTE		
Meaning:	Default setting for FGROUP command. One can specify up to eight channels whose resulting speed corresponds to the programmed path feedrate. If all eight values are zero (default), then as previously, the geometry axes recorded in MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB are active as the default setting for the FGROUP command.	

22512	EXTERN_GCODE_GROUPS_TO_PLC[n]: 0, ..., 7	
MD number	Send G code of an external language to PLC	
Default setting: 0	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	Specification of G code group of external languages whose G codes are output at the NCK/PLC interface during block change / reset. The interface is updated with each block change and after RESET. Notice: There are no systems in place to ensure that a PLC user program has a block-synchronous relation between the active NC block and pending G codes at all times (example: Path mode with very short blocks).	

22515	GCODE_GROUPS_TO_PLC_MODE	
MD number	Response of the G group transfer to PLC	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	To set the behavior of how the G groups are to be interpreted as data in the PLC. According to the current behavior (Bit 0=0), the G group is the array index of a 64-byte large array (DBB 208 - DBB 271). At the most, the 64th G group can be reached with it. In the new behavior (Bit 0=1), the data storage in the PLC is a maximum of eight bytes (DBB 208-DBB 215) large. In this process, the array index of this byte array is identical to the index of MD $MC_GCODE_GROUPS_TO_PLC[Index] and $MC_EXTERN_GCODE_GROUPS_TO_PLC[Index]. Here, each index (0-7) should only be entered for one of the two items of machine data, the value 0 must be entered for the relevant other MD. Bit 0 (LSB) = 0: Response as previously, the 64-byte large array is used for the display of the G code. Bit 0 (LSB) = 1: The user defines the G groups for which the first eight bytes are to be used	

22900	STROKE_CHECK_INSIDE	
MD number	Direction (inside/outside) in which the protection zone is effective	
Default setting: FALSE	Min. input limit: -	Max. input limit: -
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BOOLEAN		
Meaning:	It is defined whether the protection zone 3 is an inside protection zone or an outside protection zone. Meaning: 0: Protection zone 3 is an inside protection zone, i.e. the protection zone should not be overtraveled when working towards the interior 1: Protection zone 3 is an outside protection zone	

22910	WEIGHTING_FACTOR_FOR_SCALE		
MD number	Input resolution for scaling factor		
Default setting: FALSE	Min. input limit: -		Max. input limit: -
Changes effective after Power On		Protection level: 7/2	Unit: -
Data type: BOOLEAN			
Meaning:	Definition of the unit for the scaling factor P and for the axial scaling factors I, J, K Meaning: 0: Scaling factor in 0.001 1: Scaling factor in 0.00001		

22914	AXES_SCALE_ENABLE		
MD number	Activation for axial scaling factor (G51)		
Default setting: FALSE	Min. input limit: -		Max. input limit: -
Changes effective after Power On		Protection level: 7/2	Unit: -
Data type: BOOLEAN			
Meaning:	Axial scaling is disconnected with this MD. Meaning: 0: Axial scaling is not possible 1: Axial scaling is possible, i.e. MD DEFAULT_SCALE_FACTOR_AXIS is effective		
Corresponds to:	SD43120 DEFAULT_SCALE_FACTOR_AXIS		

22920	EXTERN_FIXED_FEEDRATE_F1_ON		
SD Number	Activation of fixed feedrate F1 - F9		
Default setting: FALSE	Min. input limit: -		Max. input limit: -
Changes effective after Power On		Protection level: 7/2	Unit:
Data type: BOOLEAN			
Meaning:	This MD is used to disconnect 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 become effective with the programming of F1 - F9		

22930	EXTERN_PARALLEL_GEOAX	
SD Number	Assignment of a parallel channel axis to the geometry axis	
Default setting: 0	Min. input limit: 0	Max. input limit: 20
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: BYTE		
Meaning:	Assignment tables of the axes that are parallel to the geometry axes. This table can be used to assign parallel channel axes to the geometry axes. The parallel axes can then be activated with the G functions of the plane selection (G17 - G19) and the axis name of the parallel axis as geometry axis. An axis interchange can then be executed with the axis defined via $MC_AXCONF_GEOAX_ASSIGN_TAB[ ]. Precondition: The used channel axes must be active (assigned list place in in AXCONF_MACHAX_USED). Entry of a zero deactivates the corresponding parallel geometry axis.	

24004	CHBFRAME_POWERON_MASK	
MD number	Reset channel-specific basic frame after Power On	
Default setting: 0	Min. input limit: 0	Max. input limit: 0xFFFF
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	This machine data is used to define whether channel-specific basic frames are reset to the data management during Power On Reset, i.e. shifts and rotations are set to 0, scaling to 1. Mirroring is switched off. The individual basic frames can be selected separately. Bit 0 corresponds to basic frame 0, bit 1 to basic frame 1, etc. 0: Basic frame is retained at Power ON 1: Basic frame is reset in the data management at Power ON.	
Corresponds to:	MD10651 $MN_NCBFRAME_POWERON_MASK	

24006	CHSFRAME_RESET_MASK	
MD number	Active system frames after Reset	
Default setting: 0	Min. input limit: 0	Max. input limit: 0x7FF
Changes effective after RESET	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	Bit mask for the reset setting of the channel-specific system frames calculated in the channel. Bit 0: System frame for actual value setting and scratching is active after Reset. 1: System frame for external work offset is active after Reset. 2: Reserved, TCARR and PAROT refer to $MC_GCODE_RESET_VALUES[ ]. 3: Reserved, TOROT and TORFRAME refer to $MC_GCODE_RESET_VALUES[ ]. 4: System frame for workpiece reference point is active after RESET. 5: System frame for cycles is active after Reset. 6: Reserved, reset behavior depends on $MC_RESET_MODE_MASK. 7: System frame $P_ISO1FR (ISO G51.1 Mirror) is active after Reset. 8: System frame $P_ISO2FR (ISO G68 2DROT) is active after Reset. 9: System frame $P_ISO3FR (ISO G68 3DROT) is active after Reset. 10: System frame $P_ISO4FR (ISO G51 Scale) is active after Reset. 11: System frame $P_RELFR is active after reset.	
Corresponds to:	MD28082 $MC_MM_SYSTEM_FRAME_MASK	

28082	MM_SYSTEM_FRAME_MASK	
MD number	System frames (SRAM)	
Default setting: 0x21, 0x21, ...	Min. input limit: 0	Max. input limit: 0x0000FF
Changes effective after Power On	Protection level: 7/2	Unit: -
Data type: DWORD		
Meaning:	Bit mask for the configuration of the channel-specific system frames calculated in the channel. Bit 0: System frame for actual value setting and scratching 1: System frame for external work offset 2: System frame for TCARR and PAROT 3: System frame for TOROT and TORFRAME 4: System frame for workpiece reference points 5: System frame for cycles 6: System frame for transformations 7: System frame for $P_ISO1FR for ISO G51.1 Mirror 8: System frame for $P_ISO2FR for ISO G68 2DROT 9: System frame for $P_ISO3FR for ISO G68 3DROT 10: System frame for $P_ISO4FR for ISO G51 Scale 11: System frame $P_RELFR is active after reset.	
Corresponds to:	MD28082 $MC_MM_SYSTEM_FRAME_MASK	

C.3 Axis-specific setting data

43120	DEFAULT_SCALE_FACTOR_AXIS	
MD number	Default axial scaling factor with active G51	
Default setting: 1	Min. input limit: -99999999	Max. input limit: 99999999
Change valid IMMEDIATELY	Protection level: 7/7	Unit: -
Data type: DWORD		
Meaning:	This machine data is valid in connection with external programming languages. It is effective for $MN_MM_EXTERN_LANGUAGE = 1. If no axial scaling factor I, J or K is programmed in the G51 block, then the DEFAULT_SCALEFAKTOR_AXIS is active. For the scaling factor to be effective, the MD AXES_SCALE_ENABLE must be set.	

43240	M19_SPOS	
MD number	Spindle position in degrees for spindle positions with M19	
Default setting: 0	Min. input limit: -359.999	Max. input limit: 359.999
Change valid IMMEDIATELY	Protection level: 7/7	Unit: -
Data type: DOUBLE		
Meaning:	The setting data is also effective in the Siemens mode.	

C.4 Channel-specific setting data

42110	DEFAULT_FEED	
SD Number	Default value for path feed	
Default setting: 0	Min. input limit: -	Max. input limit: -
Change valid IMMEDIATELY	Protection level: 7/7	Unit: -
Data type: DOUBLE		
Meaning:	The setting data is evaluated during part program start, giving due consideration to the feedrate type active at this point of time (refer to $MC_GCODE_RESET_VALUES or $MC_EXTERN_GCODE_RESET_VALUES).	

42140	DEFAULT_SCALE_FACTOR_P	
SD Number	Default scaling factor for address P	
Default setting: 1	Min. input limit: -	Max. input limit: -
Change valid IMMEDIATELY	Protection level: 7/7	Unit: -
Data type: DWORD		
Meaning:	If no scaling factor P is programmed in the block, then the value of this machine data is active.	
Corresponds to:	MD22910 $MC_WEIGHTING_FACTOR_FOR_SCALR	

42150	DEFAULT_ROT_FACTOR_R	
SD Number	Default rotation factor for address R	
Default setting: 0	Min. input limit: -	Max. input limit:
Change valid IMMEDIATELY	Protection level: 7/7	Unit: -
Data type: DOUBLE		
Meaning:	If no factor for rotation R is programmed with the selection of the rotation G68, the value of this setting data becomes effective.	

42160	EXTERN_FIXED_FEEDRATE_F1_F9	
SD Number	Fixed feedrates F1 - F9	
Default setting: 0	Min. input limit: -	Max. input limit: -
Change valid IMMEDIATELY	Protection level: 2/7	Unit: VELO
Data type: DOUBLE		
Meaning:	Fixed feedrate values for the programming of F1 -F9. If the machine data $MC_FEEDRATE_F!_F9_ON=TRUE is set, the feedrate values of the setting data $SC_EXTERN_FIXED_FEEDRATE_F1_F9[0] - $SC_EXTERN_FIXED_FEEDRATE_F1_F9[8] are read and activated as machining feedrate with the programming of F1 to F9. The rapid traverse feedrate must be entered in $SC_EXTERN_FIXED_FEEDRATE_F1_F9[0].	

42520	CORNER_SLOWDOWWN_START	
SD Number	Start of feedrate reduction in G62	
Default setting: 0	Min. input limit: -	Max. input limit: Any
Change valid IMMEDIATELY	Protection level: 7/7	Unit: mm
Data type: DOUBLE		
Meaning:	Path length from which the feedrate is reduced before the corner in G62.	

42522	CORNER_SLOWDOWN_END	
SD Number	End of feedrate reduction in G62	
Default setting: 0	Min. input limit: -	Max. input limit: Any
Change valid IMMEDIATELY	Protection level: 7/7	Unit: mm
Data type: DOUBLE		
Meaning:	Path length up to which the feedrate remains reduced after a corner in G62.	

42524	CORNER_SLOWDOWN_OVR	
SD Number	Override for feedrate reduction in G62	
Default setting: 0	Min. input limit: -	Max. input limit: Any
Change valid IMMEDIATELY	Protection level: 7/7	Unit: PERCENT
Data type: DOUBLE		
Meaning:	Override with which the feedrate is multiplied at the corner in G62.	

42526	CORNER_SLOWDOWN_CRIT	
SD Number	Corner detection in G62, G21	
Default setting: 0	Min. input limit: -	Max. input limit: Any
Change valid IMMEDIATELY	Protection level: 7/7	Unit: Degree
Data type: DOUBLE		
Meaning:	Angle from which a corner is considered for the feedrate reduction with G62, G21.	

43340	EXTERN_REF_POSITION_G30_1	
MD number	Reference point position for G30.1	
Default setting:	Min. input limit:	Max. input limit:
Change valid IMMEDIATELY	Protection level:	Unit:
Data type: DOUBLE		
Meaning:	Setting data Reference point position for G30.1. This setting data is evaluated in CYCLE328.	

C.5 Channel-specific cycle machine data

Table C- 1

52800	ISO_M_ENABLE_POLAR_COORD	
SD number	Polar coordinates	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Change valid IMMEDIATELY	Protection level: 7/3	Unit: -
Data type: BYTE		
Meaning:	Polar coordinates 0: OFF 1: ON	

52802	ISO_ENABLE_INTERRUPTS	
SD number	Interrupt processing	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Change valid IMMEDIATELY	Protection level: 7/3	Unit: -
Data type: BYTE		
Meaning:	Interrupt processing 0: OFF 1: ON	

52804	ISO_ENABLE_DRYRUN	
SD number	Processing skip during DRYRUN	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Change valid IMMEDIATELY	Protection level: 7/3	Unit: -
Data type: BYTE		
Meaning:	Processing skip when tapping G74/G84 during DRYRUN 0: OFF 1: ON	

52806	ISO_SCALING_SYSTEM	
SD number	Basic system	
Default setting: 0	Min. input limit: 0	Max. input limit: 2
Change valid IMMEDIATELY	Protection level: 7/3	Unit: -
Data type: BYTE		
Meaning:	Basic system 0: Not defined 1: METRIC 2: INCH	

52808	ISO_SIMULTAN_AXES_START	
SD number	Simultaneous drill position approach for all programmed axes	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Change valid IMMEDIATELY	Protection level: 7/3	Unit: -
Data type: BYTE		
Meaning:	Simultaneous drill position approach for all programmed axes 0: OFF 1: ON	

52810	ISO_T_DEEPHOLE_DRILL_MODE	
SD number	Deep-hole drilling with chip breaking/removal of chips	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Change valid IMMEDIATELY	Protection level: 7/3	Unit: -
Data type: BYTE		
Meaning:	Selection of deep-hole drilling type Deep-hole drilling with chip breaking Deep-hole drilling with removal of chips	

55800	$SCS_ISO_M_DRILLING_AXIS_IS_Z	
SD number	Drilling axis depends on plane / always Z	
Default setting: 0	Min. input limit: 0	Max. input limit: 1
Change valid IMMEDIATELY	Protection level: 7/6	Unit: -
Data type: BYTE		
Meaning:	Selection of drilling axis 0: Drilling axis is vertical to active plane 1: Drilling axis is always "Z" regardless of the active plane	

55802	$SCS_ISO_M_DRILLING_TYPE	
SD number	Tapping type	
Default setting: 0	Min. input limit: 0	Max. input limit: 3
Change valid IMMEDIATELY	Protection level: 7/6	Unit: -
Data type: BYTE		
Meaning:	Selection of drilling axis 0: Tapping without compensating chuck 1: Tapping with compensating chuck 2: Deep-hole tapping with chip breakage 3: Deep-hole tapping with chip removal	

55804	$SCS_ISO_M_RETRACTION_FACTOR	
SD number	Factor for retraction speed (0...200%)	
Default setting: 100	Min. input limit: 0	Max. input limit: 200
Change valid IMMEDIATELY	Protection level: 7/6	Unit: -
Data type: DWORD		
Meaning:	Factor for retraction speed (0...200%)	

55806	$SCS_ISO_M_RETRACTION_DIR	
SD number	Retraction direction for G76/G87	
Default setting: 0	Min. input limit: 0	Max. input limit: 4
Change valid IMMEDIATELY	Protection level: 7/6	Unit: -
Data type: DWORD		
Meaning:	Retraction direction for precision drilling and reverse countersinking G76/G87 0: G17(-X) G18(-Z) G19(-Y) 1: G17(+X) G18(+Z) G19(+Y) 2: G17(-X) G18(-Z) G19(-Y) 3: G17(+Y) G18(+X) G19(+Z) 4: G17(-Y) G18(-X) G19(-Z)	

Data lists D

D.1 Machine data

Number	Identifier	Name
General ($MN_ ...)		
10604	WALIM_GEOAX_CHANGE_MODE	Working area limitation during switchover of geometry axes
10615	NCFRAME_POWERON_MASK	Delete global basic frames during Power On
10652	CONTOUR_DEF_ANGLE_NAME	Adjustable name for angle in the contour short description
10654	RADIUS_NAME	Adjustable name for radius non-modal in the contour short description
10656	CHAMFER_NAME	Adjustable name for chamfer in the contour short description
10704	DRYRUN_MASK	Activation of dry run feedrate
10706	SLASH_MASK	Activation of block skip function
10715	M_NO_FCT_CYCLE[n]: 0, ..., 0	M function number for cycle call
10716	M_NO_FCT_CYCLE_NAME[ ]	Name for tool-changing cycle with M functions from MD $MN_NO_FCT_CYCLE
10717	T_NO_FCT_CYCLE_NAME	Name for tool-changing cycle with T number
10718	M_NO_FCT_CYCLE_PAR	M function replacement with parameters
10719	T_NO_FCT_CYCLE_MODE	Parameter assignment for T function replacement
10760	G53_TOOLCORR	Method of functioning with 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 ASUB activation
10806	EXTERN_M_NO_DISABLE_INT	M function for ASUB deactivation
10808	EXTERN_INTERRUPT_BITS_M96	Interrupt program execution (M96)
10810	EXTERN_MEAS_G31_P_SIGNAL	Assignment of the measuring inputs for G31 P..
10814	EXTERN_M_NO_MAC_CYCLE	Macro call via M function
10815	EXTERN_M_NO_MAC_CYCLE_NAME	Subroutine name for M 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)
10880	EXTERN_CNC_SYSTEM	External control system whose programs are processed
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 decimal point
10886	EXTERN_INCREMENT_SYSTEM	Increment system

Number	Identifier	Name
10888	EXTERN_DIGITS_TOOL_NO	Number of digits for T number in the external language mode
10890	EXTERN_TOOLPROG_MODE	Tool change programming in external programming language
18800	MM_EXTERN_LANGUAGE	External language is active in the control system
Channel-specific ($MC_ ...)		
20050	AXCONF_GEOAX_ASSIGN_TAB[ ]	Assignment of geometry axis to channel axis
20060	AXCONF_GEOAX_NAME_TAB[ ]	Geometry axis in channel
20070	AXCONF_MACHAX_USED[ ]	Machine axis number valid in channel
20080	AXCONF_CHANAX_NAME_TAB[ ]	Name of channel axis in the channel
20094	SPIND_RIGID_TAPPING_M_NR	M number for changing over to the controlled spindle mode (Siemens mode)
20095	EXTERN_RIGID_TAPPING_M_NR	M number for changing over to the controlled spindle mode (external language mode)
20150	GCODE_RESET_VALUES[n]: 0 to max. number of G codes	Reset G groups
20152	GCODE_RESET_MODE	Reset behavior of G groups
20154	EXTERN_GCODE_RESET_VALUES[n]: 0-30	Definition of the G codes that are effective during reboot, if the NC channel is not running in the Siemens mode
20380	TOOL_CORR_MODE_G43G44	Handling the tool length compensation G43/G44
20382	TOOL_CORR_MOVE_MODE	Retract the tool length compensation
20732	EXTERN_G0_LINEAR_MODE	Interpolation behavior with G00
20734	EXTERN_FUNCTION_MASK	Function mask for external language
22420	FGROUP_DEFAULT_AXES[ ]	Default value of the FGROUP command
22512	EXTERN_GCODE_GROUPS_TO_PLC[n]: 0-7	Specification of G groups that are output in the NCK PLC interface, if an external NC language is active
22900	STROKE_CHECK_INSIDE	Direction (inside/outside) in which the protection zone is effective
22910	WEIGHTING_FACTOR_FOR_SCALE	Input resolution for scaling factor
22914	AXES_SCALE_ENABLE	Activation for axial scaling factor (G51)
22920	EXTERN_FEEDRATE_F1_F9_ACTIV	Activation of fixed feedrate (F0 - F9)
22930	EXTERN_PARALLEL_GEOAX	Assignment of parallel channel geometry axis
24004	CHBFRAME_POWERON_MASK	Reset channel-specific basic frame after Power On
24006	CHSFRAME_RESET_MASK	Active system frames after Reset
28082	MM_SYSTEM_FRAME_MASK	System frames (SRAM)

D.2 Setting data

Number	Identifier	Name
Axis-specific		
43120	DEFAULT_SCALE_FACTOR_AXIS	Default axial scaling factor with active G51
43240	M19_SPOS	Spindle position in degrees for spindle positions with M19
43340	EXTERN_REF_POSITION_G30_1	Reference position for G30.1
Channel-specific		
42110	$SC_DEFAULT_FEED	Default value for path feed
42140	$SC_DEFAULT_SCALE_FACTOR_P	Default scaling factor for address P
42150	$SC_DEFAULT_ROT_FACTOR_R	Default for angle of rotation R
42520	$SC_CORNER_SLOWDOWN_START	Start of feedrate reduction in G62
42522	$SC_CORNER_SLOWDOWN_END	End of feedrate reduction in G62
42524	$SC_CORNER_SLOWDOWN_OVR	Override for feedrate reduction in G62
42526	$SC_CORNER_SLOWDOWN_CRIT	Corner detection in G62, G21

D.3 Variables

Identifier	Type	Description
$C_A	REAL	Value of programmed address A in the ISO dialect mode for cycle programming
$C_B	REAL	Value of programmed address B in the ISO-dialect-mode for cycle programming
....	....	.....
$C_G	INT	G number for cycle calls in the external mode
$C_H	REAL	Value of programmed address H in the ISO dialect mode for cycle programming
$C_I[ ]	REAL	Value of the programmed Address I in the ISO-dialect-mode for cycle programming and macro technology with G65/G66. A max. of 10 entries is possible in the block for the macro programming. The values remain in the programmed sequence in the array.
$C_I_ORDER[ ]	REAL	For description, refer to $C_I[ ], defines the programming sequence
$C_J[ ]	REAL	For description refer to $C_I[ ]
$C_J_ORDER[ ]	REAL	For description, refer to $C_I[ ], defines the programming sequence
$C_K[ ]	REAL	For description refer to $C_I[ ]
$C_K_ORDER[ ]	REAL	For description, refer to $C_I[ ], defines the programming sequence
$C_L	INT	Value of programmed address L in the ISO- dialect-mode for cycle programming
$C_M	REAL	Value of programmed address M in the ISO-dialect-mode for cycle programming
$C_P	INT	Value of programmed address P in the ISO-dialect-mode for cycle programming
$C_Q	REAL	Value of programmed address Q in the ISO-dialect-mode for cycle programming
....	....	....
$C_Z	INT	Value of programmed address Z in the ISO-dialect-mode for cycle programming
$C_TS	STRING	String of tool identifier programmed under the address T
$C_A_PROG	INT	Address A is programmed in a block with cycle call 0 = Not programmed 1 = Programmed (absolute) 3 = Programmed (incremental)
$C_B_PROG	INT	Address B is programmed in a block with cycle call 0 = Not programmed 1 = Programmed (absolute) 3 = Programmed (incremental)
....	....	....
$C_G_PROG	INT	Shell cycle is programmed over a G function
$C_Z_PROG	INT	Address Z is programmed in a block with cycle call 0 = Not programmed 1 = Programmed (absolute) 3 = Programmed (incremental)
$C_TS_PROG	INT	A tool identifier was programmed under Address T TRUE = programmed; FALSE = not programmed

Identifier	Type	Description
$C_ALL_PROG	INT	Bit pattern of all programmed addresses in a block with cycle call Bit 0 = Address A Bit 25 = Address Z Bit = 1 address is programmed Bit = 0 address is not programmed
$P_EXTGG[n]	INT	Active G code of the external language
$C_INC_PROG	INT	Bit pattern of all incrementally programmed addresses in a block with cycle call Bit 0 = Address A Bit 25 = Address Z Bit = 1 address is programmed incrementally Bit = 0 address is not programmed as absolute
$C_I_NUM	INT	Cycle programming: Value is always 1 if Bit 0 is set in $C_I_PROG. Macro programming: Number of address I programmed in the block (max. 10).
$C_J_NUM	INT	For description refer to $C_I_NUM
$C_K_NUM	INT	For description refer to $C_I_NUM
$P_AP	INT	Polar coordinates 0 = Off, 1 = On
$C_TYP_PROG	INT	Bit pattern of all programmed addresses in a block with cycle call Bit 0 = A Bit 25 = Z Bit = 0 Axis is programmed as INT Bit = 1 Axis is programmed as REAL
$C_PI	INT	Program no. of the interrupt routine that was programmed with M96

Interrupts E

An alarm is created on detecting error states in cycles and the cycle execution is interrupted.

Messages are output from the cycles in the status bar of the control system. These messages do not interrupt the machining.

The alarms having the numbers 61000 to 62999 are generated in the cycles. This large area is further subdivided according to alarm responses and cancel criteria.

Table E- 1 Alarm no. and alarm description

Alarm No.	Brief description	Cause	Explanation/remedy
General alarms			
61001	Incorrect thread lead	CYCLE376T	The thread lead is not specified correctly
61003	No feedrate programmed in cycle	CYCLE371T, CYCLE374T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	No F word was programmed in the calling block before the cycle call, refer to Siemens standard cycles.
61004	Configuration of geometry axis is not correct	CYCLE328	The sequence of the geometry axes is incorrect, refer to Siemens standard cycles
61101	Reference plane defined incorrectly	CYCLE375T, CYCLE81, CYCLE83, CYCLE84, CYCLE87	Refer to Siemens standard cycles
61102	No spindle direction programmed	CYCLE371T, CYCLE374T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Spindle direction M03 or M04 is missing, refer to Siemens standard cycles
61107	First drilling depth defined incorrectly		First drilling depth is opposite to total drilling depth
61603	Groove shape defined incorrectly	CYCLE374T	Value of groove depth equal to 0
61607	Starting point programmed incorrectly	CYCLE376T	The starting point lies outside the area to be machined.
61610	No infeed depth programmed	CYCLE374T	Infeed value = 0
ISO alarms			
61800	External CNC system is missing	CYCLE300, CYCLE328, CYCLE330, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Machine data for external language MD18800 $MN_MM_EX-TERN_ LANGUAGE or options bit 19800 $MN_EXTERN_LAN-GUAGE is not set.

Alarm No.	Brief description	Cause	Explanation/remedy
61801	Wrong G code selected	CYCLE300, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T	An impermissible value has been programmed in the program call CYCLE300<Value> or a wrong value has been specified in the cycle setting data for the G code system.
61802	Wrong 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 control system. Check MD20050-20080.
61804	Progr. Position exceeds reference point	CYCLE328, CYCLE330	The programmed intermediate position or the current position lies behind the reference point.
61805	Value programmed as absolute and incremental	CYCLE328, CYCLE330, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T	The intermediate position is programmed with absolute as well as incremental values.
61806	Wrong axis assignment	CYCLE328	The sequence of the axes is wrong.
61807	Incorrect spindle direction programmed	CYCLE384M	The programmed spindle direction contradicts the spindle direction provided for the cycle.
61808	Final drilling depth or single drilling depth is missing	CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Total drilling depth Z or single drilling depth Q is missing in the G8x block (first call of cycle)
61809	Drilling position not permissible	CYCLE383T, CYCLE384T, CYCLE385T	
61810	ISO G code not possible	CYCLE383T, CYCLE384T, CYCLE385T	
61811	ISO axis name not permissible	CYCLE328, CYCLE330. CYCLE371T, CYCLE374T, CYCLE376T, CLE383T, CYCLE384T, CYCLE385T	The calling NC block contains an impermissible ISO axis identifier.
61812	Value(s) in the external cycle call is defined incorrectly	CYCLE371T, CYCLE376T,	The calling NC block contains an impermissible numerical value.
61813	GUD value defined incorrectly	CYCLE376T	A wrong numerical value was entered in the cycle setting data.
61814	Polar coordinates with cycle not possible	CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	
61815	G40 not active	CYCLE374T, CYCLE376T	G40 was not active before cycle call.

Glossary

A spline

The Akima spline runs tangentially through the programmed interpolation points (3rd order polynomial).

Absolute dimensions

A specification for the target of axis motion is defined by a dimension that refers to the origin of the currently active coordinate system. See also -> incremental dimension.

AC (adaptive control)

A process variable (e.g. path- or axis-specific feedrate) can be controlled as a function of another measured process variable (e.g. depending on spindle current). Typical application: Maintenance of constant chip removal volume during grinding.

Acceleration and jerk limit

To achieve an optimal acceleration factor for the machine while preserving the mechanical parts, the machining program offers the option of switching between instantaneous (inertia-free) and constant (jerk-free) acceleration.

Access rights

The blocks of a CNC program are protected via the following 7-level system of access-authorization restrictions:

- Three password levels, one each for the system manufacturer, the machine manufacturer and the user;
- Four key-operated switch settings which can be evaluated via the PLC.

Activation / deactivation

Working area limitation is a means of limiting the axis motion over the restrictions set by the limit switch. A pair of values can be specified for each axis with which the protection zone is delimited.

Address

Addresses are fixed or variable identifiers for axes (X, Y, ...), spindle speed (S), feedrate (F), circle radius (CR), etc.

Analog input and output modules

Analog input and output modules are sensors for analog process signals.

Analog input modules are used to convert measured analog values into digital values, so that they can be processed in the CPU. Analog input modules convert digital values into manipulated variables.

Archiving

Exporting files or directories to an external memory medium.

Asynchronous subprogram

- A part program that can be started with an interrupt signal (e.g. "Rapid NC input signal") asynchronously (i.e. independently) while a part program is active.
- A part program that can be started with an interrupt signal (e.g. "Rapid NC input signal") asynchronously (i.e. independent of the current program status).

AUTOMATIC or Automatic mode

Operating mode of the control system (block sequence according to DIN): Operating mode of NC controls in which a -> part program is selected and continuously executed.

Auxiliary functions

Auxiliary functions can be used to transfer -> parameters in part programs to the -> PLC, whereupon reactions defined by the machine manufacturer are triggered.

Axes

CNC axes are set according to their functionality as follows:

- Axes: Interpolatory path axes
- Positioning axes: Non-interpolating infeed and positioning axes with axis-specific feedrates; these axes can be traversed beyond the block limits. Positioning axes need not be involved in the machining of the tool and they contain, e.g. tool feeder, tool magazine etc.

Axis identifier

According to DIN 66217, axes are denoted using X, Y, and Z for a clockwise, rectangular -> coordinate system.

-> Rotary axes that rotate around X, Y and Z, to which the identifiers A, B and C are assigned. Additional axes parallel to the mentioned ones can be denoted with other letters.

Axis/spindle replacement

An axis/spindle is permanently assigned to a specific channel via machine data. This assignment via machine data can be reset with program commands and the axis/spindle can be assigned to another channel.

B spline

The points programmed for the B spline are not interpolation points but simply "checkpoints". The generated curve only runs near the checkpoints, not directly through them (optional 1st, 2nd or 3rd order polynomials).

Backlash compensation

Compensation of the mechanical backlash of the machine, e.g. the backlash compensation of the spindles. The backlash compensation can be input separately for each axis.

Backup

A copy of the memory contents (hard disk) stored on an external device for the purpose of data backup/archiving.

Base axis

Axis whose setpoint value or actual value is used in the calculation of the offset value.

Basic coordinate system

Cartesian coordinate system which is mapped through a transformation onto the machine coordinate system.

The programmer works in the -> part program with the axis names of the basic coordinate systems. The basic coordinate system exists in parallel to the -> machine coordinate system if no -> transformation is active. The difference between the two systems lies only in the axis identifiers.

Baud rate

The speed at which the data transfer takes place (bits/s).

Block search

The block search function can be used to go to any convenient point in a part program at which the machining is to start or continue. This function is intended for testing the part programs or for continuing the machining after an interruption.

Booting

Loading the system program after power on.

C axis

An axis which the tool uses to describe a controlled rotary or positioning movement.

C spline

The C spline is the most well-known and widely used spline. The spline runs along a tangent and along the axis of curvature through all interpolation points. 3rd order polynomials are used for this.

Channel structure

The -> programs of the individual channels can be processed simultaneously or asynchronously with the channel structure.

Circular interpolation

In circular interpolation, the -> tool traverses between defined contour points at a definite feedrate during the machining of the workpiece on a circular path.

Clearance control (3D), sensor-driven

The positional shift for a certain axis can be controlled as a function of a measured process variable (e.g. analog input, spindle current...). This function can be used to maintain a fixed distance automatically so as to fulfill certain technological requirements of the relevant machining.

CNC programming language

The CNC programming language is based on DIN 66025 with high-level language extensions. The CNC programming language and the high-level language extensions support the definition of macros (sequenced statements).

COM

Part of the numerical control system for the implementation and coordination of communication.

Command axis

Command axes are started from synchronized actions as a reaction to an event (command). Command axes can be positioned, started and stopped fully asynchronous to the part program.

Compensation axis

Axis with a setpoint or actual value modified by a compensation value.

Compensation memory

Data storage area in the control system, in which the tool offset data is stored.

Compensation table

Table of interpolation points. It delivers the offset values of the offset axis for the selected positions of the basic axis.

Connecting cables

Connecting cables are either pre-assembled cables or cables that are pre-assembled, ready-to-use two-wire cables that have a connector at both ends. Connecting cables are used to connect the -> CPU to a -> programming device or another CPU with use of an -> "interface".

Continuous-path mode

The purpose of the continuous-path mode is to avoid an excessive acceleration of the -> path axes at the block limits of part programs, which may affect the operator, the machine or other assets of the plant. The continuous-path mode is expected to influence the transition to the next block in the NC program and to maintain the path velocity as uniformly as possible.

Contour

Contours of a workpiece.

Contour monitoring

The following error is monitored as a measure for contour precision within a defined tolerance range. Thus, for instance, an overloading of the drive can lead to a sequential fault, which is no longer acceptable. An alarm is output and the axes are stopped in this case.

Correction value

The distance between the axis position and the desired programmed axis position measured with a position encoder.

CPU

Central Processor Unit -> programmable logic controller

Cycle

Protected subprogram for executing machining operations which are always repeated at the -> workpiece.

Cycle support

The available cycles are listed in the "Cycle support" menu in the "Program" operating area. The required parameters for the assignment of the values are displayed in plain text after the desired execution cycle has been selected.

Data block

- The unit of data used in the -> PLC which can be accessed via -> HIGHSTEP program.
- Unit of data in the -> NC: Data blocks that contain data definitions for global user data. This data can be initialized directly during its definition.

Data word

Data unit inside a -> PLC data block having a size of 2 bytes

Deletion of distance-to-go

Command in a part program with which the machining is stopped and the distance-to-go is deleted.

Diagnostics

- Operating area of the control.
- The control system contains a self-diagnostics program and test routines for service: Status, alarm and service displays.

DRF

Differential Resolver Function. This is a function of an NC in which an incremental work offset is created in the automatic mode in conjunction with the electronic handwheel.

Dynamic feedforward control

The "Dynamic acceleration-dependent feedforward control" function can be used to fully overcome the contour inaccuracies that result from secondary faults. The feedforward control function enables an extremely high degree of machining accuracy at high tool path velocities. The feedforward control function allows selection or deselection of all axes with just one part program.

Editor

The editor is used to create, change, add, connect and insert programs/texts/blocks of a program.

Electronic handwheel

The selected axes can be operated simultaneously in manual mode with an electronic handwheel. The handwheel clicks are evaluated by the increment analyzer.

Exact stop

If exact stop is programmed, the position specified in the block is approached accurately and - if necessary - very slowly. To reduce the startup speeds, -> exact stop limits are defined for rapid traverse and feedrate.

Exact stop limit

After all the path axes reach their exact stop limits, the control system reacts as if they have reached the end point exactly. The -> part program continues with the machining from the next block.

External work offset

Work offset specified by the -> PLC.

Fast retraction from contour

When an interrupt occurs, a motion can be initiated via the CNC machining program, enabling the tool to be quickly retracted from the workpiece contour that is currently being machined. The retraction angle and the retraction path can also be parameterized. An interrupt routine can be carried out after a rapid retraction.

Feedrate override

In feedrate override, the current feedrate entered in the operator panel or specified by the PLC is overlaid on the programmed feedrate (0 - 200%). A feedrate override is also possible by means of a programmable percentage value (1 - 200%) in the machining program.
A feedrate correction can also be applied by synchronous actions regardless of the program currently running.

Fixed machine point

Point that is defined uniquely by the machine tool, e.g. the reference point.

Fixed-point approach

The machine tool can be used to approach defined fixed points such as tool change points, loading points, palette change points etc. The coordinates of these points are stored in the control system. If possible, the control system approaches these axes at -> rapid traverse.

Frame

A frame is an arithmetic rule, with which one Cartesian coordinate system is converted to another Cartesian coordinate system. A frame consists of the components -> work offset -> rotation -> scaling and -> mirroring.

Geometry

Description of a -> workpiece in the -> workpiece coordinate system.

Geometry axis

Geometry axes are used to describe a 2- or 3-dimensional range in the workpiece coordinate system.

Global main program / subprogram

Each global main program / subprogram can be stored only once under its name in the directory. However, the same name can be used several times in one and the same directory.

Ground

The term "Ground" is used for all electrically inactive, interconnected parts of one part of the system or a resource which cannot result in dangerous touch voltage in the event of errors.

Helical interpolation

The "Helical interpolation" function is especially suitable for machining female or male threads with form cutters and for milling lubricating grooves. The helix comprises two movements:

Circular movement in the plane
Linear movement vertical to this plane

High-speed digital inputs/outputs

Example: High-speed CNC program routines (interrupt routines) that can be started via digital inputs. Digital CNC outputs (SINUMERIK 840D sl) can be used to trigger switching functions operated by high-speed programs.

HIGHSTEP

Combination of different programming features for the -> PLC in the range S7-300/400.

I/O module

I/O modules are used to establish a connection between the CPU and the process.

I/O modules are:

Digital input and output modules
Analog input and output modules
Simulator modules

Identifiers

According to DIN 66025, identifiers (names) of variables (arithmetic variables, system variables, user variables), of subprograms, of vocabulary words and of words can contain several address characters. These characters have the same meaning as the words in the block syntax. Identifiers must always be unique. Different identifiers should always be used for different objects.

Inch measuring system

Measuring system with which the distance traversed and fractions thereof are specified in inches.

Inclined axis

Fixed angular interpolation with allowance for an inclined infeed axis or grinding wheel by specifying the angle. The inclined axes are programmed and displayed in the Cartesian coordinate system.

Inclined surface machining

The "Inclined surface machining" function supports drilling and milling operations on workpiece surfaces that are inclined with reference to the coordinate planes of the machine. The position of the inclined surface can be defined via the oblique position of the coordinate system (refer to FRAME programming).

Increment

The end point for the traversing of axes is defined by the distance to be covered and a direction that refers to a point already reached. See also -> Absolute dimension.
Specification of the length of the travel path in increments. The number of increments can be stored either in the -> setting data or it can be selected with the keys 10, 100, 1000 and 10 000.

Initialization block

Initialization blocks are special -> program blocks. These contain values that must be assigned before executing the program.
Initialization blocks are the preferred choice for initializing previously defined data or global user data.

Initialization files

An initialization file can be created for each -> workpiece. Different instructions for variable values that are valid exclusively for a workpiece, can be stored in the initialization file.

Intermediate blocks

Movements with a selected tool offset (G41/G42) can be interrupted by a limited number of intermediate blocks (blocks without traversing movements at the compensation plane). The tool offset can be calculated correctly while using intermediate blocks. The number of intermediate blocks that can be read in advance by the control system can be set in the system parameters.

Interpolation cycle

The interpolation cycle is a multiple of the basic system cycle. The interpolation cycle is used to specify the cycle time that is required to update the setpoint interface with the position controls. The resolution of the velocity profiles is determined with the interpolation cycle.

Interpolator

Logical unit of the -> NCK which can be used to determine the intermediate values for the movements of the individual axes to be executed on the basis of the target positions specified in the part program.

Interpolatory compensation

Interpolatory compensation is a means of compensating the leadscrew error (SSFK) and measuring system error (MSF) that result from the production process.

Interrupt routine

Interrupt routines are special -> subprograms which can be started by events (external signals) in the machining process. Here, the block of the part program just processed is interrupted and the axis position at the interruption point is stored automatically. See -> ASUP

Interrupts

All -> messages and alarms are displayed in plain text on the operator panel. The alarm text contains the date, time and a suitable symbol for the cancel criterion.

Alarms and messages are displayed separately according to the following criteria:

- 1. Alarms and messages in the part program:

  Alarms and messages can be displayed directly from the program as plain text.
- 2. PLC alarms and messages that concern the machine can be displayed directly from the PLC in plain text format. No additional function blocks are required for this.

Inverse-time feedrate

In the SINUMERIK 840D sl controls, the speed of the axis motion can be specified (G93) instead of the feedrate for traversing the distance stored in a block.

JOG

Operating mode of the CNC (in the setup operation): The machine can be setup in the JOG mode. The individual axes and spindles can be traversed in steps (in the JOG mode) with direction keys. Other functions offered by the JOG mode are -> Reference point approach, -> REPOS (reposition) -> Preset -> (actual value specification).

Keyswitch

S7-300: In the S7-300, the key-operated switch is the mode selector switch in the -> CPU. The key-operated switch is operated with a removable key.

840D sl: The key-operated switch on the -> machine control panel has four positions to which the operating system of the control system has assigned appropriate functions. Each key-operated switch has three differently colored keys which can be removed in the specified positions.

Languages

The texts of the user interface, system messages and alarms are available in five system languages: English, German, French, Italian and Spanish. The user can always select two of the languages listed in the control system.

Leadscrew error compensation

Compensation for the mechanical inaccuracies of a spindle involved in the feed movement. Errors are compensated by the control system on the basis of the measured deviations that are stored in the control system.

Linear axis

In contrast to a rotary axis, a linear axis describes a straight line.

Linear interpolation

In the linear interpolation, the tool moves along a straight line to the end point during machining of the workpiece.

LookAhead

The "LookAhead" function is a means of optimizing the machining speed through looking ahead using a parameterizable number of traversing blocks.

LookAhead for contour violations

The control system recognizes and reports the following types of collision:
The travel path is shorter than the tool radius.
The width of the internal corner is smaller than the tool diameter.

Machine control panel

A control panel in the machine tool having operator controls such as keys, rotary switches, etc., as well as simple displays such as LEDs. The machine control panel is used for direct control of the machine tool via the PLC.

Machine coordinate system

Coordinate system based on the axes of the machine tool.

Machine zero

A fixed point on the machine tool to which all measuring systems (derived from it) refer.

Machining channel

The channel structure can be used to reduce downtime by executing motional sequences simultaneously. Thus, for instance, the gantry of a loader can execute its movements during the machining. In this case, the CNC functions as an autonomous control system which performs operations like decoding, block preparation and interpolation independently.

Macros

Several instructions in different programming languages can be combined with each other in one instruction. This abbreviated sequence of instructions is called in the CNC program under a user-defined name. The instructions are executed one after the other with the macro.

Main block

A block that is preceded by a ":" sign and that contains all the parameters necessary for starting the processing of a -> part program.

Main program

A -> part program that is denoted by a number or a name in which other main programs, subprograms or -> cycles can be called.

Main run

Part program blocks that have been decoded and prepared via block preparation, are processed in the "main run".

MDA

Control operating mode: Manual Data Automatic = manual date input in automatic mode. Individual program blocks or block sequences with no reference to a main program or subprogram can be input in MDA mode; these are processed immediately on actuating the NC start key.

Memory reset

The following -> CPU memories are erased by a memory reset operation:

- -> Work memory
- Read/write area of the -> load memory
- -> System memory
- -> Backup memory

Metric or inch dimensions

Position and thread lead values can be programmed in the machining program in inches. The control is always set to the basic system independently of the programmed unit of measurement (G70/G71).

Metric units system

Standardized system of length units in millimeters, meters, etc.

Mirroring

Mirroring changes the sign of the coordinate value of a contour with reference to an axis. Mirroring can be executed simultaneously for several axes.

Mode group

All axes/spindles are assigned at a randomly specified time to a single channel. Each channel is assigned to a mode group (BAG). The same -> mode is always assigned to the channels in a mode group.

Motion synchronization

This function can be used to trigger actions that run simultaneously (synchronously) with the machining. The starting point of the actions is defined by a condition (e.g. the status of a PLC input, the time that has elapsed since the start of a block). The start of motion-synchronized actions is not linked to block limits.

Examples of typical motion-synchronized actions are: Transfer of M- and H-(auxiliary) functions at the PLC or distance-to-go delete for certain axes.

NC

"Numerical Control" = numerical control system; contains all components of the control system of the machine tool: -> NCK, -> PLC, -> HMI, -> COM.

NCK

Numerical Control Kernel: Component of the NC control system, which executes -> part programs and essentially coordinates the movements at the machine tool.

Net, Network

A net or network is the connection of several S7-300 and other automation or HMI devices such as programming devices via -> connecting cables. The interconnected devices exchange data across the network.

NURBS

Motion control and path interpolation are executed in the control system internally on the basis of NURBS (non-uniform rational B splines). Thus, there is a standard procedure (SINUMERIK 840D sl) for all operating modes as an internal control function.

OEM

The scope of the implementation of individual solutions (OEM applications) for the SINUMERIK 840D sl was developed for machine manufacturers, who want to create their own user interfaces or who want to integrate process-oriented functions in the control system.

Online tool offset

This function can be used for grinding tools only.

The reduction in the size of the grinding wheel through dressing is transferred to the relevant active tool as tool offset and is immediately effective.

Operating mode

Operator control concept for SINUMERIK control systems. The following modes exist: -> JOG, -> MDA and -> AUTOMATIC.

Oriented spindle stop

Stops the spindle at a defined orientation angle, e.g. for the execution of additional machining operation at the specified position.

Oriented tool retraction

RETTOOL: If the machining is interrupted (e.g. in case of tool breakage), the tool can be retracted with a program command along a fixed path in an orientation defined by the user.

Override

Manually adjustable or programmable feature of the control system, with which the users can superimpose the programmed feedrates and speeds to adjust them to their special tool or material.

Part program

A sequence of instructions to the NC control whose combination is supposed to create a certain -> workpiece by executing certain DO operations on a specified -> raw part.

Part program management

The "part program management" function can be organized according to the -> workpieces. The number of programs and data to be managed depends on the capacity of the control memory and can also be configured via machine data settings. A name consisting of a maximum of 16 alphanumeric characters can be assigned to each file (programs and data).

Path axis

Path axes are all the machining axes of a -> channel that is controlled by the -> interpolator in such a way that they start together, accelerate at the same time and reach their end position at the same time.

Path feedrate

The path feedrate acts on the -> path axes. It represents the geometrical sum of the feedrates of the participating -> path axes.

Path velocity

The maximum programmable path velocity depends on the input resolution. The maximum programmable path velocity at a resolution of, say, 0.1 mm is 1.000 m/min.

PCIN data transfer program

PCIN is a routine for transfer and reception of CNC user data, such as part programs, tool offsets etc. over a serial interface. The PCIN program runs on standard commercial PCs under MS-DOS.

PLC

Programmable Logic Control -> programmable logic controller. Component of -> NC: Programmable control system for executing the control logic of the machine tool.

PLC program memory

The PLC user program, the user data and the PLC main program are stored together in the PLC user memory of the PLC. The PLC user memory can be expanded to 128 KB.

PLC programming

The PLC is programmed using the STEP 7 software. The STEP 7 programming software is based on the WINDOWS standard operating system and contains the functionality of STEP 5 programming with innovative enhancements and developments.

Polar coordinates

A coordinate system, in which the position of a point in the plane is defined with a defined axis from the perspective of its distance from the coordinate origin and the angle formed by the radius vector.

Polynomial interpolation

Polynomial interpolation is a means with which a very large range of curve characteristics including straight line, parabola and exponential functions can be created.

Positioning axis

An axis that executes auxiliary movement in the machine (e.g. tool magazine, palette transport). Positioning axes are axes that do not interpolate with -> path axes.

Preprocessing memory, dynamic

The traversing blocks are prepared before their execution (pre-processed) and stored in a "preprocessing" memory. Block sequences can be executed from this memory at a very high speed. Blocks can be loaded in the preprocessing memory continuously during the machining.

Preprocessing stop

Program command The next block in a part program is executed only if all the earlier prepared blocks stored in the preprocessing memory have been processed.

Preset

The machine zero point can be redefined in the machine coordinate system by means of the preset function. No axis is traveled during preset; instead, a new position value is input for the current axis position.

Programmable frames

New starting points of a coordinate system can be defined dynamically with programmable -> frames when the program is running. One differentiates between absolute definitions in which new frames are used, and additive definitions in which the definition is undertaken with reference to an existing starting point.

Programmable logic controller

Programmable logic controllers (PLC) are electronic control systems whose functions are stored in the control system as a program. Hence, the structure and wiring do not depend on the control functions. Programmable logic controllers are designed exactly like a computer, i.e. they consist of a CPU with memory, I/O modules and an internal bus system. The selection of the I/O modules and the programming language depends on the technology used.

Programmable working area limitation

Limiting the traversing range of the tool to defined, programmable limits.

Programming key

Characters and character strings with an accurately defined meaning within the programming language for -> part programs (refer to programming manual).

Protection zone

Three-dimensional zone inside a -> working area which the tool cannot reach (can be programmed via MD).

Quadrant error compensation

Contour errors in quadrant transitions that are caused by frictional losses on the guide rails can be corrected to a large extent with the quadrant error compensation. A circularity test is used to define the parameters of the quadrant error compensation.

Rapid traverse

The maximum rapid traverse rate of an axis is, for instance, used to traverse a tool from the idle position to the -> workpiece contour or to retract the tool from the workpiece contour.

Reference point

Point on the machine to which the measuring system of the -> machine axis refers.

Reference point approach

If the position measuring system used is not an absolute encoder, then reference point approach must be started so that the actual values delivered by the measuring system match the machine coordinate system values.

REPOS

1. Repositioning to the contour, triggered by the operator

   REPOS can be used to reset the tool to the interruption point with direction keys.

2. Programmed repositioning to the contour

   A selection of approach strategies is available in the form of program commands: Approaching the interruption point, approaching the starting block, approaching the last block, approaching a point on the path between block start and interruption point.

Revolutional feedrate

The axis feedrate is set in the channel as a function of the main spindle speed (programming with G95).

Rigid tapping

This function is used for tapping without compensating chuck. In this case, the spindle is controlled as an interpolatory rotary axis with the effect that the threads are drilled up to the final drilling depth, e.g. in tapping of blind holes (precondition: Spindle is operated as axis).

Rotary axis

Rotary axes are used to rotate the tool or workpiece by a certain angle.

Rotary axis, continuously turning

The traversing range of a rotary axis can be set either to a modulo value (can be adjusted via machine data) or as endlessly turning in both directions, depending on the relevant application. Endlessly turning rotary axes are used, e.g. for non-circular machining, grinding and winding jobs.

Rotation

Component of a -> frame which defines a rotation of the coordinate system through a specific angle.

Rounding axis

Rounding axes enable the workpiece or tool to turn by a certain angle that is stored in the indexing grid. The rounding axis is "in position" when the latching is reached.

R-parameters

The programmer can assign or query the values of the R-parameters, if necessary in the -> part program.

S7 configuration

"S7 configuration" is a tool for defining the parameters of modules. "S7 configuration" can be used to set different -> parameter sets of the -> CPU and the I/O modules in the -> programming device. These parameters are loaded to the CPU.

S7-300 bus

The S7-300 bus is a serial data bus that supplies the appropriate voltage to the modules, and that is used by these modules for data interchange. The modules are interconnected via plug-in bus connectors.

Safety functions

The control system has constantly active monitoring functions with which errors in the -> CNC, the programmable control system (-> PLC) and the machine can be detected early enough to avoid damages to the workpiece, tool or machine to a large extent. The machining is interrupted and the drives are stopped when a fault or error occurs. The cause of the error is recorded and an alarm is output. At the same time, the PLC is notified that a CNC alarm has been triggered.

Safety Integrated

Effective protection according to the EU Guideline >>89/392/EEC<<, >>Safety Integrity Level 3<< according to EN-954-1 (the B 1-4 classes are defined in this standard) for the safety of operators and machines, integrated in the control system for safe installation and testing.

Failure safety is guaranteed. This safety function is also effective during individual faults.

Scaling

Component of a -> frame through which axis-specific changes are made.

Set

All files that are required for the programming and execution of a program are denoted as sets.
A section of a -> part program which ends with "LineFeed" (line break). A distinction is made between -> main blocks and -> subblocks.

Setting data

Data that supplies information on the features of the machine to the control system; the method is defined in the system software. Unlike -> machine data, setting data can be modified by the user.

Softkey

A key whose name is displayed in one part of the screen. The selection of the softkeys that are displayed is adjusted automatically to the relevant operating state. The freely programmable function keys (softkeys) are assigned to certain functions that are defined via the software.

Software limit switch

The limits of the traversing range of an axis is defined with software limit switches and this prevents the carriage from coming in contact with the hardware limit switches. Two pairs of values can be assigned per axis, and these can be activated separately via the -> PLC.

Speed limitation

Maximum/minimum (spindle) speed: The maximum spindle speed is limited by the values that are specified either in the machine data by the-> PLC or the -> setting data.

Spindles

The spindle functionality is a construct with two levels:

Spindles: Speed or position-controlled spindle drives, analog digital (SINUMERIK 840D sl)

Auxiliary spindles: Speed-controlled spindle drives without encoder, e.g. for power tools.

Spline interpolation

The control system can generate a smooth curve with spline interpolation, for which a small number of interpolation points are necessary along a programmed contour.

Standard cycles

Standard cycles are used to program DO operations that are repeated frequently:

- For drilling/milling
- For measuring tools and workpiece

The available cycles are listed in the "Cycle support" menu in the "Program" operating area. The required parameters for the assignment of the values are displayed in plain text after the desired execution cycle has been selected.

Station number

The station number represents the "contact address" of a -> CPU or a -> programming device or any other intelligent I/O module if these communicate with each other via a -> network. The station number is assigned to the CPU or the programming device with the S7 tool -> "S7 configuration".

Subblock

Block preceded by "N" containing information about a machining step, e.g. a position specification.

Subprogram

A sequence of instructions of a -> part program which can be called multiple times with various output parameters. Subprograms are always called from the main programs. Subprograms can be locked to prevent unauthorized export and impermissible viewing. -> Cycles are subprograms from the perspective of type.

Synchronization

Instructions in -> part programs for coordinating working cycles in different -> channels at certain machining points.

Synchronized actions

- Auxiliary function output

  Technological functions (-> auxiliary functions) can be output by the CNC program to the PLC during the machining of a workpiece. These auxiliary functions can, for instance, be used to control ancillary equipment on the machine (quill, gripper, chuck etc.).

- Fast auxiliary function output

  The acknowledgement times of -> auxiliary functions can be reduced and unnecessary machining process for executing less critical switching functions can be avoided.

Synchronized actions can be combined to build programs (technology cycles). Axis programs can be started in the same IPO cycle, e.g. by scanning digital inputs.

Synchronized axes

Synchronized axes require the same time to traverse the path as -> geometry axes.

Synchronous spindle

Accurate synchronism of an angle between a master spindle and one or more slave spindles. This way, the flying transfer of a workpiece from Spindle 1 to Spindle 2 is possible in turning machines.

In addition to speed synchronism, relative angle positions of the spindles can also be programmed, e.g. "floating" or the position-oriented transfer of inclined workpieces.

Several pairs of synchronous spindles can be implemented.

System variables

A variable that exists even though it was not programmed by -> part program programmers. It is defined by the data type and the variable name with the sign $. See also -> User-defined variable.

Teach-in

Teach-in is a means of creating and correcting part programs. The individual program blocks can be input via the keyboard and executed immediately. The positions approached with the direction keys or the handwheel can also be stored. Additional information such as G functions, feedrates or M functions can be entered in the same block.

Tool

A tool that is used to shape a workpiece. Tools are, e.g. turning tools, milling cutters, twist drills, laser beams, grinding wheels, etc.

Tool offset

A tool is selected by programming a T function (5 digits, integer) in the block. Every T number can be assigned up to 9 cutting edges (D addresses). The number of tools that are managed in the control system can be parameterized.

The tool length compensation is selected by programming D numbers.

Tool radius compensation

A contour is programmed on the basis of the assumption that a tool is used with a tool tip. As this is not always the case in practice, the radius of curvature of the used tool is specified so that an allowance is considered by the tool. The center of curvature is guided to the contour which is equidistant with an offset that corresponds to the radius of curvature.

Transformation

If programming is done in one Cartesian coordinate system and execution is done in another non-Cartesian coordinate system (e.g. with machine axis as rotary axis); then inclined axes and 5-axis transformation are used in connection with transmit.

Transmit

This function can be used to mill outside contours on rotating parts, e.g. four-sided parts (linear axis with rotary axis).

3D interpolation with two linear axes and one rotary axis is also possible. The advantages of Transmit simplify programming and improve the efficacy of the machine through complete machining: Turning and milling can be executed on the same machine without rechucking.

Traversing range

The maximum possible traversing range for linear axes is ± 9 decades. The absolute value depends on the selected resolution for the data input and position control, and on the used unit of measurement (inch or metric).

User interface

The user interface (UI) is the human machine interface (HMI) of a CNC. It appears on the screen and has eight horizontal and eight vertical softkeys.

User memory

All programs and data, such as part programs, subprograms, comments, tool offsets, work offsets/frames, as well as channel and program user data, can be stored in the shared CNC user memory.

User-defined variables

Users have the option of defining variables for their own purposes in the -> part program or in a data block (global user data). The variable definition contains the specification of the data type and the variable name. See also -> System variable.

Variable definition

A variable is defined by specifying a data type and a variable name. The variable name can be used to access the value of the variable.

Velocity control

The control system can execute a look-ahead evaluation over several blocks (-> LookAhead) to achieve an acceptable traversing speed during movements that require only very small adjustments of position in a block.

Vocabulary words

Words with a certain notation and a defined meaning in the programming language for -> part programs.

Work memory

The work memory is a memory offering free access (RAM or Random Access Memory) in the -> CPU which the processor accesses while executing the application program.

Work offset

Specification of a new reference point for a coordinate system by referring to an existing zero point and a -> Frame.

1. Settable
SINUMERIK 840D sl: There is a parameterizable number of adjustable work offsets for each CNC axis. Each work offset can be selected via G functions; the selection is exclusive.

2. External
All movements that define the position of the workpiece zero can be superimposed by an external work offset, which is either defined by a handwheel (DRF movement) or the PLC.

3. Programmable
The TRANS instruction can be used to program work offsets for all path and positioning axes.

Working area

Three-dimensional zone in which the tool tip can be moved due to the physical design of the machine. See also -> Protection zone.

Workpiece

The part that is produced/machined on the machine tool.

Workpiece contour

Programmed contour of the -> workpiece to be made/machined.

Workpiece coordinate system

The origin of the workpiece coordinate system is the -> workpiece zero. For working cycles that are programmed in the workpiece coordinate system, the dimensions and directions refer to this system.

Workpiece zero

The workpiece zero is the origin of the -> workpiece coordinate system. It is determined based on its distance from the machine zero point.

Index