Programming Guide 11/2002 Edition

sinumerik

Fundamentals

SINUMERIK 840D/840Di/810D

SIEMENS

SIEMENS

SINUMERIK 840D/840Di/810D

Fundamentals

Programming Guide

Valid for

Control	*Software Version*
SINUMERIK 840D	6
SINUMERIK 840DE (export version)	6
SINUMERIK 840D powerline	6
SINUMERIK 840DE powerline	6
SINUMERIK 840Di	2
SINUMERIK 840DiE (export version)	2
SINUMERIK 810D	3
SINUMERIK 810DE (export version)	3
SINUMERIK 810D powerline	6
SINUMERIK 810D powerline	6

11.02 Edition

SINUMERIK® Documentation

Printing history

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

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

Status code in the "Remarks" column:

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

Edition	Order No.	Remarks
02.95	6FC5298-2AB00-0BP0	A
08.97	6FC5298-4AB00-0BP0	A
12.95	6FC5298-3AB00-0BP0	C
03.96	6FC5298-3AB00-0BP1	C
08.97	6FC5298-4AB00-0BP0	C
12.97	6FC5298-4AB00-0BP1	C
12.98	6FC5298-5AB00-0BP0	C
08.99	6FC5298-5AB00-0BP1	C
04.00	6FC5298-5AB00-0BP2	C
10.00	6FC5298-6AB00-0BP0	C
09.01	6FC5298-6AB00-0BP1	C
11.02	6FC5298-6AB00-0BP2	C

This manual is included in the documentation available on CD ROM (**DOCONCD**)

Edition	Order No.	Remarks
11.02	6FC5298-6CA00-0BG3	C

Trademarks

SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIROTEC®, SINUMERIK® and SIMODRIVE® are registered trademarks of Siemens AG. Other names in this publication might be trademarks whose use by a third party for his own purposes may violate the rights of the registered holder.

Further information is available on the Internet under:
http://www.ad.siemens.de/sinumerik

This publication was produced with WinWord V8.0 and Designer V4.0.

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

We have checked that the contents of this document correspond to the hardware and software described. Nonetheless, differences might exist and therefore we cannot guarantee that they are completely identical. The information given in this publication is reviewed at regular intervals and any corrections that might be necessary are made in the subsequent printings. We welcome suggestions for improvement.

Subject to change without prior notice

Order No. 6FC5298-6AB00-0BP2
Printed in Germany

Siemens Aktiengesellschaft

Contents

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Preface

Organization of documentation

SINUMERIK documentation is organized on three different levels:

- General Documentation
- User documentation
- Manufacturer/Service Documentation

Target group

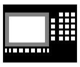
This Manual is intended for machine-tool users. It provides detailed information that the user requires to program the SINUMERIK 840D/840Di/810D control system.

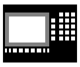
Standard scope

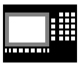
This Programming Guide describes the functionality afforded by standard functions. Differences and additions implemented by the machine-tool manufacturer are documented by the machine-tool manufacturer.

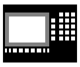
More detailed information about other publications relating to SINUMERIK 840D/840Di and publications that apply to all SINUMERIK controls (e.g. Universal Interface, Measuring Cycles...) can be obtained from your local Siemens branch office.

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.


840D
NCU 571


840D
NCU 572
NCU 573


810D


840Di

Applicability

This Programming Guide applies to the following controls:

SINUMERIK 840D	6
SINUMERIK 840DE (export version)	6
SINUMERIK 840D powerline	6
SINUMERIK 840DE powerline	6
SINUMERIK 840Di	2
SINUMERIK 840DiE (export version)	2
SINUMERIK 810D	3
SINUMERIK 810DE (export version)	3
SINUMERIK 810D powerline	6
SINUMERIK 810D powerline	6

with operator panels OP 010, OP 010C, OP 010S, OP 12 or OP 15 (PCU 20 or PCU 50)

SINUMERIK 840D powerline

From 09.2001 onwards, improved performance versions of

- SINUMERIK 840D powerline and
- SINUMERIK 840DE powerline

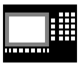
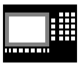
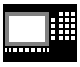
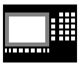
will be available. For a list of available **powerline** modules, please refer to Section 1.1 /PHD/ of the hardware description /PHD/.

SINUMERIK 810D powerline

From 12.2001 onwards, improved performance versions of

- SINUMERIK 810D powerline and
- SINUMERIK 810DE powerline

will be available. For a list of available **powerline** modules, please refer to Section 1.1 of the hardware description /PHC/.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Hotline

If you have any queries, please contact the following hotline:

A&D Technical Support Phone: ++49-(0)180-5050-222
Fax: ++49-(0)180-5050-223
Email: adsupport@siemens.com

Please send any queries about the documentation (suggestions or corrections) to the following fax number or email address:

Fax: ++49-(0)0131-98-2176
Email: motioncontrol.docu@erlf.siemens.de

Fax form: see feedback sheet and the end of the publication.

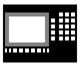
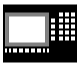
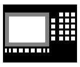
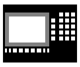
Internet address http://www.ad.siemens.de/sinumerik

Export version

The following functions are not available in the export version:

Function	810DE	840DE
Five axis machining package	–	–
Handling transformation package (five axes)	–	–
Multi-axis interpolation (> four axes)	–	–
Helical interpolation 2D+6	–	–
Synchronized actions, stage 2	–	O[1)]
Measurements, stage 2	–	O[1)]
Adaptive control	–	O[1)]
Continuous dressing	–	O[1)]
Utilization of compile cycles (OEM)	–	–
Sag compensation, multi-dimensional	–	O[1)]

– Function not available
1) Restricted functionality

840D NCU 571	840D NCU 572 NCU 573	810D	840Di

Fundamentals

This Programming Guide Fundamentals is intended for use by skilled machine operators with the appropriate expertise in drilling, milling and turning operations. Simple programming examples are used to explain the commands and statements which are also defined according to DIN 66025.

Advanced

The Programming Guide "Advanced" is intended for use by technicians with in-depth, comprehensive programming knowledge. By virtue of a special programming language, the SINUMERIK 840D/810D control enables the user to program complex workpiece programs (e.g. for sculptured surfaces, channel coordination, ...) and greatly facilitates the programming of complicated operations.
The commands and statements described in this Guide are not specific to one particular technology.
They can be applied for a variety of technologies, such as

- Grinding
- Cyclical machines (packaging, woodworking)
- Laser power controls.

840D
NCU 571

840D
NCU 572
NCU 573

810D

840Di

Structure of descriptions

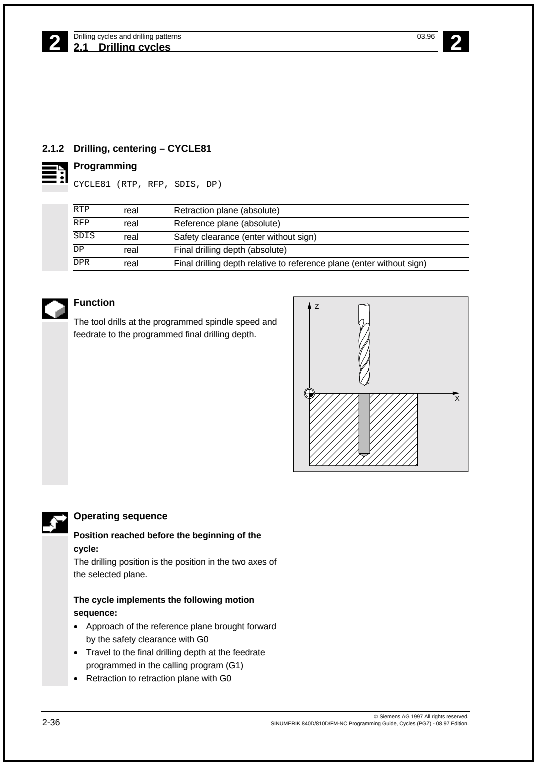
All cycles and programming options have been described according to the same internal structure as far as this is meaningful and practicable. The various levels of information have been organized such that you can selectively access the information you need for the task in hand.

1. A quick overview

If you look up a rarely used command or the meaning of a parameter, you can see at a glance how the function is programmed and find helpful explanations of the commands and parameters.

This information is always displayed at the top of the page.

Note:
Due to lack of space, it has not been possible to show all the modes of representation afforded by the programming language for individual commands and parameters. For this reason, we have illustrated those command programming schemes that are used most frequently in practice in a workshop situation.


2 Drilling cycles and drilling patterns 03.96 2
2.1 Drilling cycles

2.1.2 Drilling, centering – CYCLE81

Programming

```
CYCLE81 (RTP, RFP, SDIS, DP)
```

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)

Function

The tool drills at the programmed spindle speed and feedrate to the programmed final drilling depth.

Operating sequence

Position reached before the beginning of the cycle:
The drilling position is the position in the two axes of the selected plane.

The cycle implements the following motion sequence:
- Approach of the reference plane brought forward by the safety clearance with G0
- Travel to the final drilling depth at the feedrate programmed in the calling program (G1)
- Retraction to retraction plane with G0

2-36 © Siemens AG 1997 All rights reserved.
SINUMERIK 840D/810D/FM-NC Programming Guide, Cycles (PGZ) - 08.97 Edition.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

2. Detailed explanations

You will find detailed answers to the following questions in the theory section:

Why is the command needed?

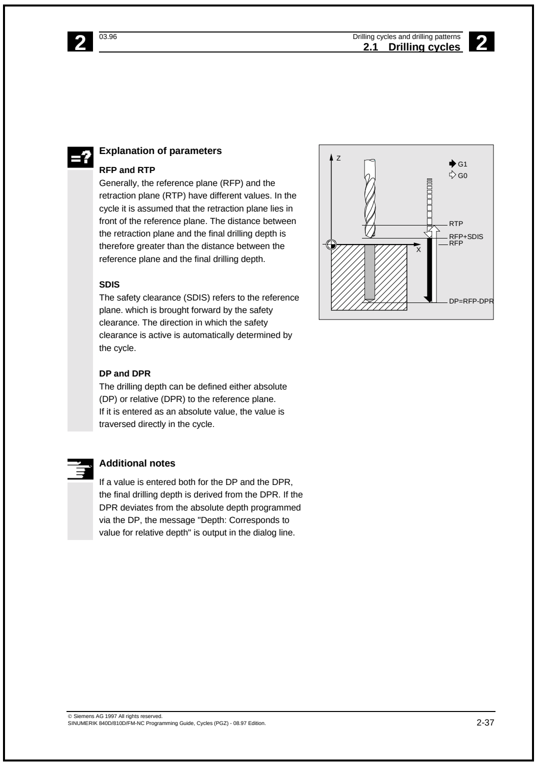
What does the command do?

How is it programmed and executed?

What do the parameters do?

What else do I need to know?

The theoretical sections are primarily intended as learning material for the NC entry-level user. You should work through the manual at least once to get an idea of the functional scope and capability of your SINUMERIK control.



2 03.96 Drilling cycles and drilling patterns 2
2.1 Drilling cycles

Explanation of parameters

RFP and RTP
Generally, the reference plane (RFP) and the retraction plane (RTP) have different values. In the cycle it is assumed that the retraction plane lies in front of the reference plane. The distance between the retraction plane and the final drilling depth is therefore greater than the distance between the reference plane and the final drilling depth.

SDIS
The safety clearance (SDIS) refers to the reference plane. which is brought forward by the safety clearance. The direction in which the safety clearance is active is automatically determined by the cycle.

DP and DPR
The drilling depth can be defined either absolute (DP) or relative (DPR) to the reference plane.
If it is entered as an absolute value, the value is traversed directly in the cycle.

Additional notes

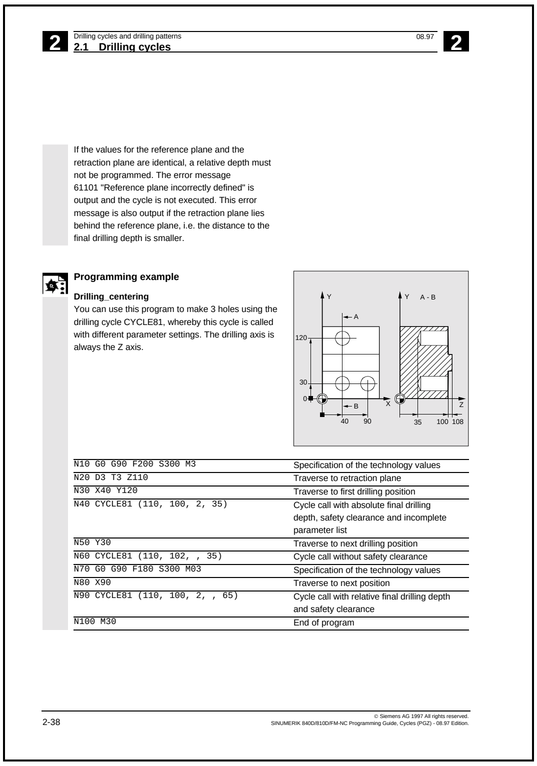
If a value is entered both for the DP and the DPR, the final drilling depth is derived from the DPR. If the DPR deviates from the absolute depth programmed via the DP, the message "Depth: Corresponds to value for relative depth" is output in the dialog line.

© Siemens AG 1997 All rights reserved.
SINUMERIK 840D/810D/FM-NC Programming Guide, Cycles (PGZ) - 08.97 Edition. 2-37

3. From theory to practice

The programming examples illustrate how commands can be applied in practice.

You will find an application example for virtually every command after the theoretical section.



2 Drilling cycles and drilling patterns 08.97 2
2.1 Drilling cycles

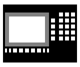
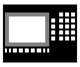
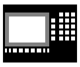
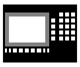
If the values for the reference plane and the retraction plane are identical, a relative depth must not be programmed. The error message 61101 "Reference plane incorrectly defined" is output and the cycle is not executed. This error message is also output if the retraction plane lies behind the reference plane, i.e. the distance to the final drilling depth is smaller.

Programming example

Drilling_centering
You can use this program to make 3 holes using the drilling cycle CYCLE81, whereby this cycle is called with different parameter settings. The drilling axis is always the Z axis.

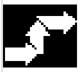
`N10 G0 G90 F200 S300 M3`	Specification of the technology values
`N20 D3 T3 Z110`	Traverse to retraction plane
`N30 X40 Y120`	Traverse to first drilling position
`N40 CYCLE81 (110, 100, 2, 35)`	Cycle call with absolute final drilling depth, safety clearance and incomplete parameter list
`N50 Y30`	Traverse to next drilling position
`N60 CYCLE81 (110, 102, , 35)`	Cycle call without safety clearance
`N70 G0 G90 F180 S300 M03`	Specification of the technology values
`N80 X90`	Traverse to next position
`N90 CYCLE81 (110, 100, 2, , 65)`	Cycle call with relative final drilling depth and safety clearance
`N100 M30`	End of program

2-38 © Siemens AG 1997 All rights reserved.
SINUMERIK 840D/810D/FM-NC Programming Guide, Cycles (PGZ) - 08.97 Edition.

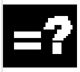
   

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Explanation of symbols



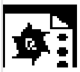
Operating sequence



Explanation

Function

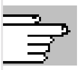
Parameters



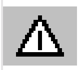
Programming example

Programming

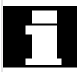
Additional notes



Cross-references to other documentation or sections

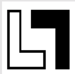


Notes and warnings



Machine manufacturer (MH n)

n= number of the note per section to which the machine manufacturer can refer.



Ordering data option

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Principle

Your SIEMENS 840D/840Di/810D has been designed and constructed according to state-of-the-art technology and approved safety regulations and standards.

Additional equipment

The applications of SIEMENS controls can be expanded for specific purposes through the addition of special add-on devices, equipment and expansions supplied by SIEMENS.

Personnel

Only **appropriately trained, authorized and reliable personnel** may be allowed to operate this equipment. The control must never be operated, even temporarily, by anyone who is not appropriately skilled or trained.

The relevant **responsibilities** of personnel who set up, operate and maintain the equipment must be clearly **defined**; the proper fulfillment of these responsibilities must be **monitored**.

Behavior

Before the control is started up, it must be ensured that the Operator's Guides have been read and understood by the personnel responsible. The operating company is also responsible for **constantly monitoring** the overall technical state of the control (visible faults and damage, altered service performance).

Servicing

Repairs must be carried out according to the information supplied in the service and maintenance guide by personnel who are **specially trained and qualified** in the relevant technical subject. All relevant safety regulations must be followed.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Note

The following is deemed to be **improper usage** and **exempts the manufacturer from any liability**:

Any application which does not comply with the rules for proper usage described above.

If the control is **not in technically perfect condition** or is operated without due regard for safety regulations and accident prevention instructions given in the Instruction Manual.

If faults that might affect the safety of the equipment are not rectified **before** the control is started up.

Any **modification, bypassing** or **disabling** of items of equipment on the control that are required to ensure fault-free operation, unlimited use and active and passive safety.

Improper usage gives rise to **unforeseen dangers** to:

- Life and limb of personnel,
- The control, machine or other assets of the owner and the user.

The following special symbols and keywords have been used in this documentation:

Notes
This symbol appears in this documentation whenever it is necessary to draw your attention to an important item of information.

In this documentation, you will find this symbol with a reference to an ordering option. The function described is executable only if the control contains the designated option.

Warnings
The following warnings with varying degrees of severity appear in this document.

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

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

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

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

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

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

Fundamental Geometrical Principles

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

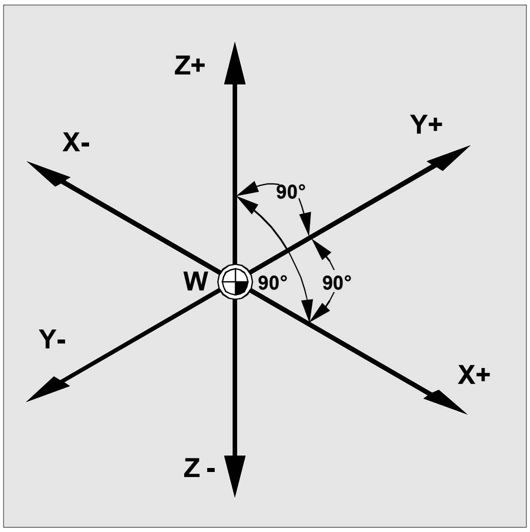
1.1 Description of workpiece points

1.1.1 Workpiece coordinate systems

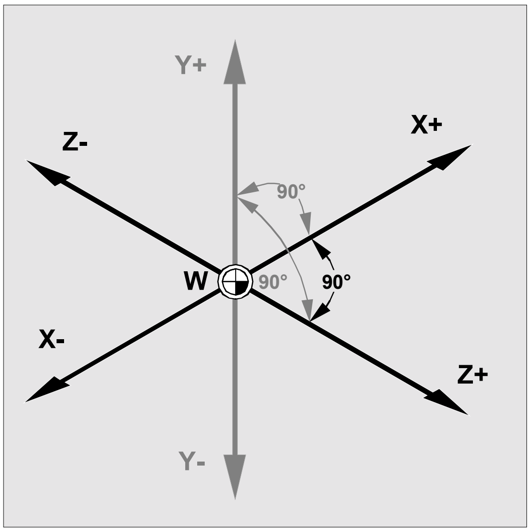
In order for the machine or control to operate with the specified positions, these data must be made in a reference system that corresponds to the direction of motion of the axis slides. A coordinate system with the axes X, Y and Z is used for this purpose.
DIN 66217 stipulates that machine tools must use right-handed, rectangular (cartesian) coordinate systems.

The workpiece zero (W) is the origin of the workpiece coordinate system. Sometimes it is advisable or even necessary to work with negative positional data. Positions to the left of the origin are prefixed by a negative sign (–).

Milling:



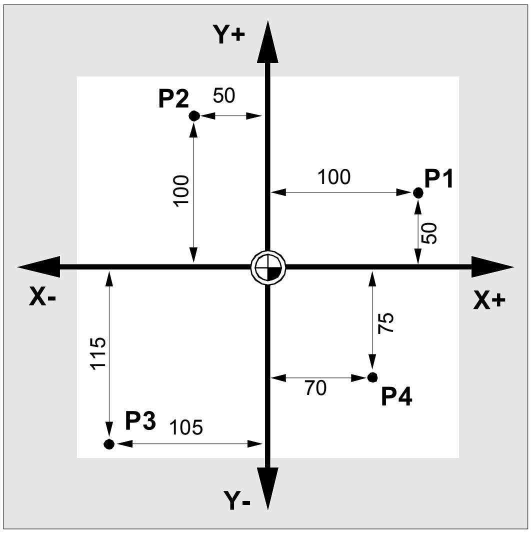
Turning:



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.1.2 Definition of workpiece positions

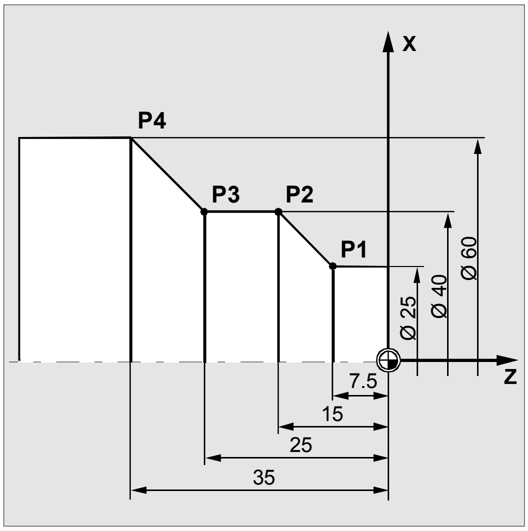
To specify a position, imagine that a ruler is placed along the coordinate axes. You can now describe every point in the coordinate system by specifying the direction (X, Y and Z) and three numerical values. The workpiece zero always has the coordinates X0, Y0 and Z0.



Example:
For the sake of simplicity, we will only use one plane of the coordinate system in this example, i.e. the X/Y plane. Points P1 to P4 then have the following coordinates:

P1	corresponds to	`X100 Y50`
P2	corresponds to	`X-50 Y100`
P3	corresponds to	`X-105 Y-115`
P4	corresponds to	`X70 Y-75`

One plane is sufficient to describe the contour on a **turning machine**.



Example:

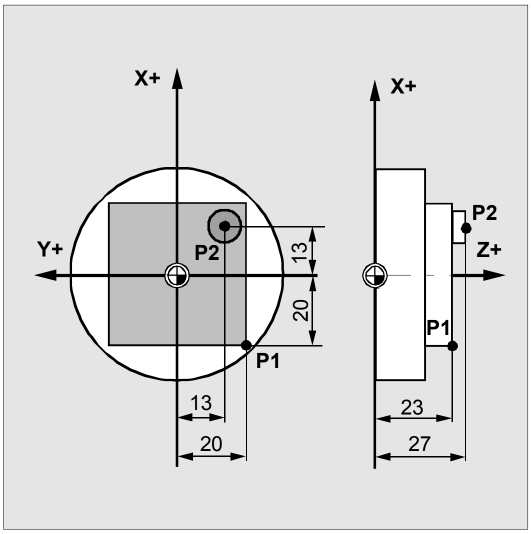
Points P1 to P4 are defined by the following coordinates:

P1	corresponds to	`X25 Z-7.5`
P2	corresponds to	`X40 Z-15`
P3	corresponds to	`X40 Z-25`
P4	corresponds to	`X60 Z-35`

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Example:
Points P1 and P2 are defined by the following coordinates:

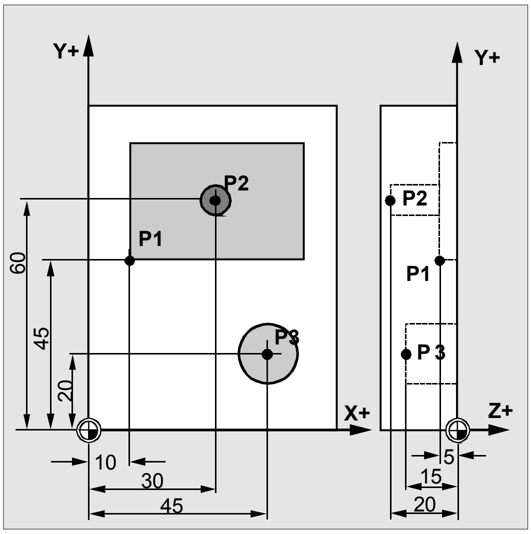
P1 corresponds to `X-20 Y-20 Z23`
P2 corresponds to `X13 Y-13 Z27`



The infeed depth must also be described in **milling operations**. To do this, we need to specify a numerical value for the third coordinate (Z in this case).

Example:
Points P1 to P3 are defined by the following coordinates:

P1 corresponds to `X10 Y45 Z-5`
P2 corresponds to `X30 Y60 Z-20`
P3 corresponds to `X45 Y20 Z-15`



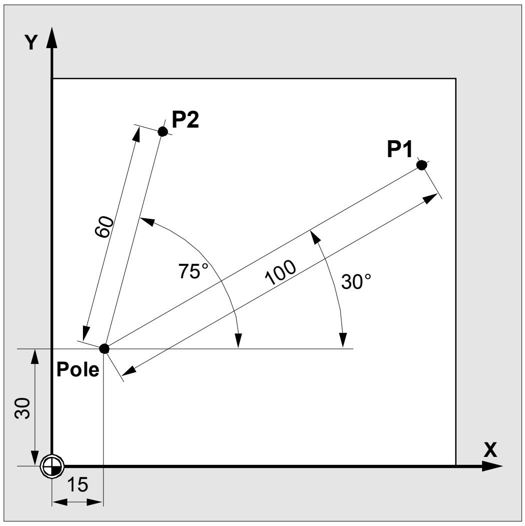
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.1.3 Polar coordinates

The coordinates used up to this point to specify points in the coordinate system are called "Cartesian coordinates".

However, there is another way to specify coordinates, namely as "polar coordinates".

It is useful to use polar coordinates in cases where a workpiece or part of a workpiece is dimensioned by radius and angle. The origin of the dimensional measurements is referred to as the "pole".



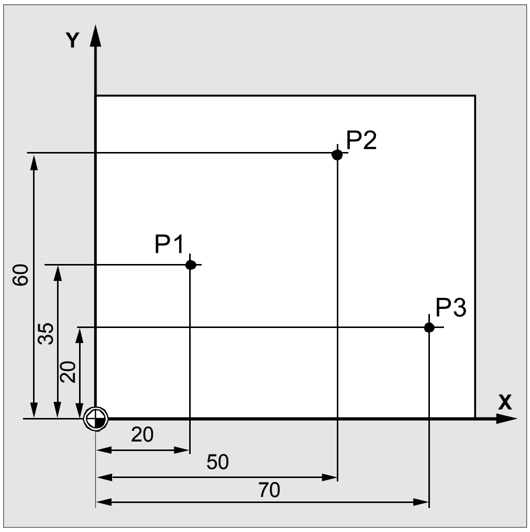
Example:
The points P1 and P2 can then be described – with reference to the **pole** – as follows:
P1 corresponds to radius =100 plus angle =30°
P2 corresponds to radius =60 plus angle =75°

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.1.4 Absolute dimension

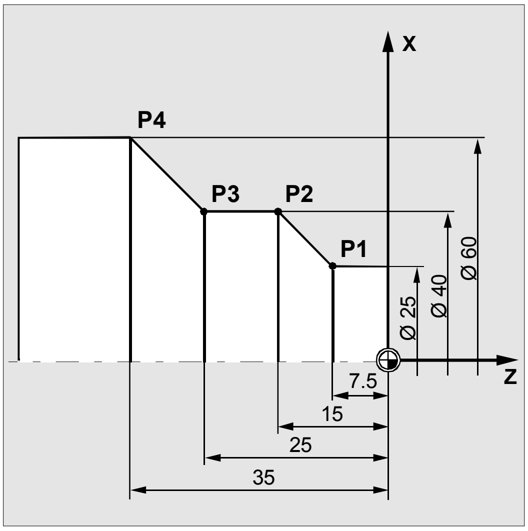
With absolute dimensions, all the positional parameters refer to the currently valid zero point. Applied to tool movement this means:

The absolute dimensions describe the position to which the tool is to travel.



Example for milling:
The positional parameters for points P1 to P3 in absolute dimensions **referring to the zero point** are the following:

P1 corresponds to `X20 Y35`
P2 corresponds to `X50 Y60`
P3 corresponds to `X70 Y20`



Example for turning:
The positional parameters for points P1 to P4 in absolute dimensions **referring to the zero point** are the following:

P1 corresponds to `X25 Z-7.5`
P2 corresponds to `X40 Z-15`
P3 corresponds to `X40 Z-25`
P4 corresponds to `X60 Z-35`

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.1.5 Incremental dimension

Production drawings are frequently encountered, however, where the dimensions refer not to the origin, but to another point on the workpiece.

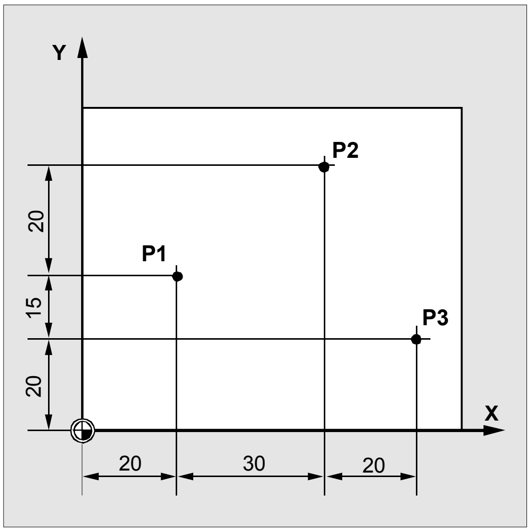
In order to avoid having to convert such dimensions, it is possible to specify them in incremental dimensions.

Incremental dimensions refer to the positional data for the previous point. Applied to tool movement this means:

The incremental dimensions describe the distance the tool is to travel.

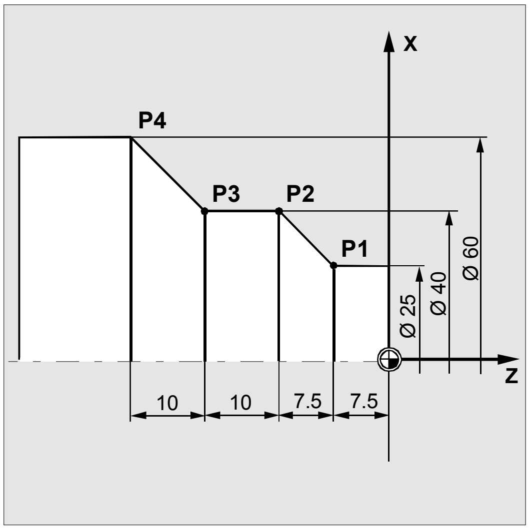
Example for milling:
The positional data for points P1 to P3 in incremental dimensions are:

P1 corresponds to `X20 Y35` ;(with reference to the zero point)
P2 corresponds to `X30 Y20` ;(with reference to P1)
P3 corresponds to `X20 Y-35` ;(with reference to P2)

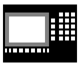


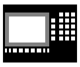
Example for turning:
The positional data for points P1 to P4 in incremental dimensions are:

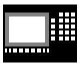
`G90` P1 corresponds to `X25 Z-7.5` ;(with reference to the zero point)
`G91` P2 corresponds to `X15 Z-7.5` ;(with reference to P1)
`G91` P3 corresponds to `Z-10` ;(with reference to P2)
`G91` P4 corresponds to `X20 Z-10` ;(with reference to P3)

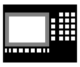


When DIAMOF or DIAM90 is active, the path setpoint is programmed as a radius dimension with G91.


840D
NCU 571


840D
NCU 572
NCU 573


810D

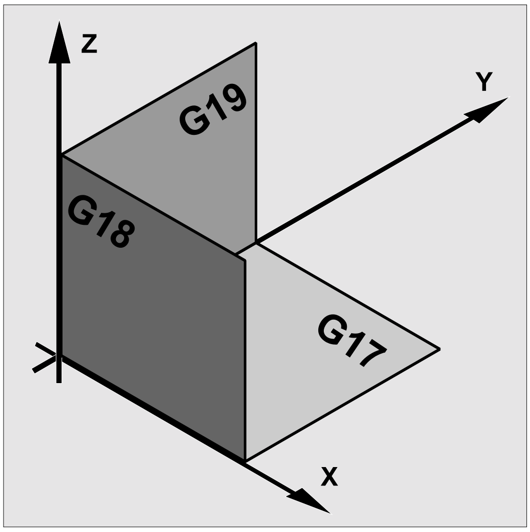

840Di

1.1.6 Plane designations

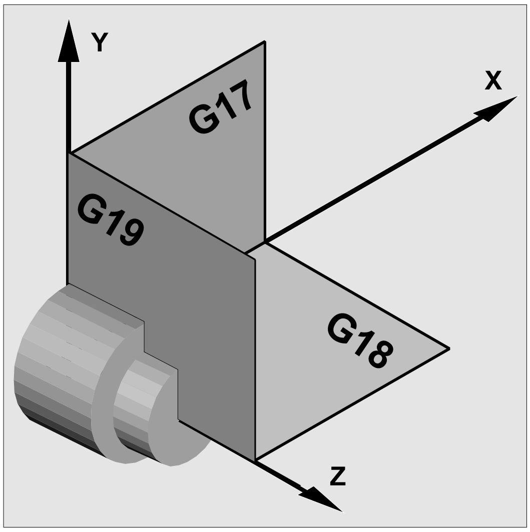
A plane is defined by means of two coordinate axes. The third coordinate axis is perpendicular to this plane and determines the infeed direction of the tool (e.g. for 2½D machining).

When programming, it is necessary to specify the working plane in order that the control can calculate the tool offset values correctly. The plane is also relevant to certain types of circular programming and polar coordinates.

Milling:



Turning:



The working planes are specified as follows in the NC program with G17, G18 and G19:

Plane	*Identifier*	*Infeed direction*
X/Y	G17	Z
Z/X	G18	Y
Y/Z	G19	X

1.2 Position of zero points

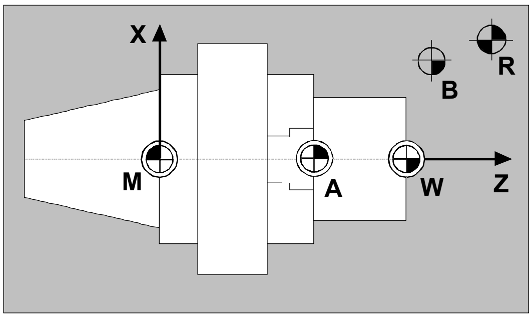
The various origins and reference positions are defined on the NC machine. They are reference points

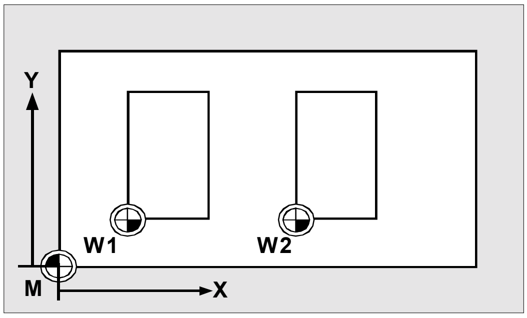
- for the machine to approach and
- refer to programming the workpiece dimensions.

They are:

M = Machine zero
A = Blocking point. Can coincide with the workpiece zero (turning machines only)
W = Workpiece zero = Program zero
B = Start point. Can be defined for each program. Start point of the first tool for machining.
R = Reference point. Position determined by cam and measuring system. The distance to the machine zero M must be known, so that the axis position can be **set** to exactly this value at this position.

The diagrams show the zero points and reference points for turning machines and drilling/milling machines.





1.3 Position of coordinate systems

1.3.1 Overview of various coordinate systems

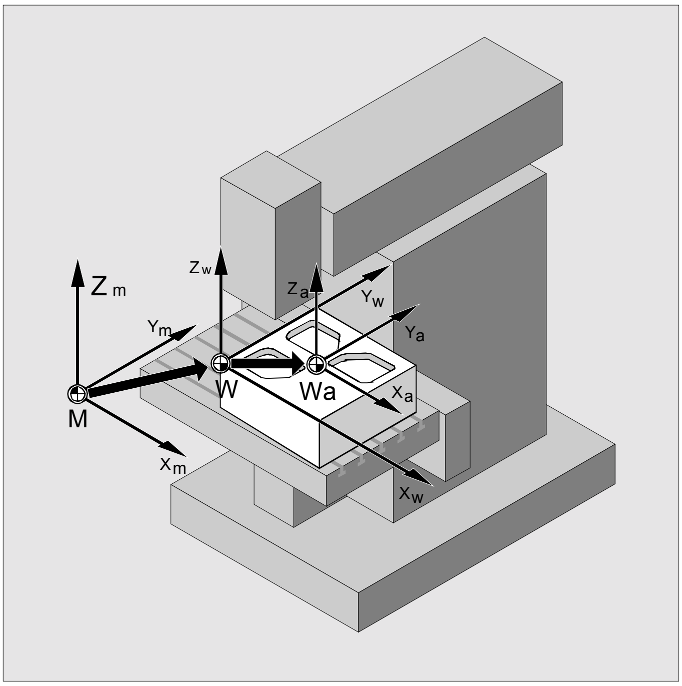
We distinguish between the following coordinate systems:

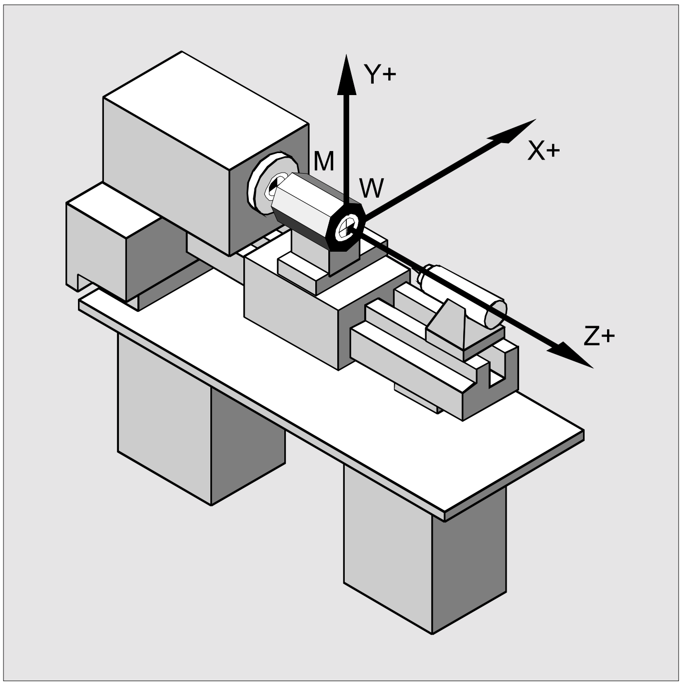
- The machine coordinate system with the machine zero **M**
- The basic coordinate system (this can also be the workpiece coordinate system W)
- The workpiece coordinate system with the workpiece zero **W**
- The current workpiece coordinate system with the current offset workpiece zero **Wa**

In cases where various different machine coordinate systems are in use (e.g. 5-axis transformation), an internal transformation function mirrors the machine kinematics on the coordinate system currently selected for programming.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

The individual axis identifiers are explained in the subsection headed "Axis types" in this section.





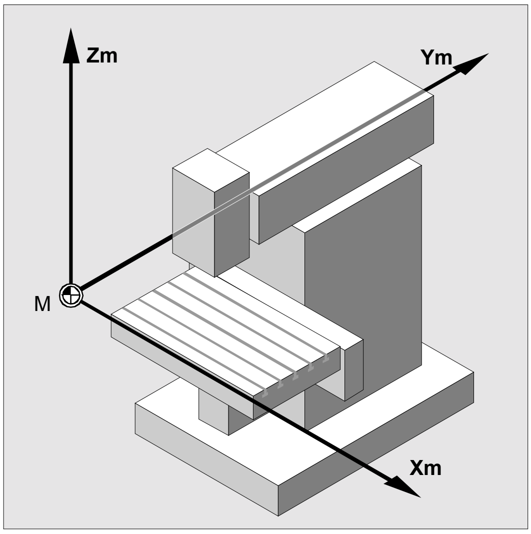
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.3.2 Machine coordinate system

The machine coordinate system comprises all the physically existing machine axes.

Reference points and tool and pallet changing points (fixed machine points) are defined in the machine coordinate system.

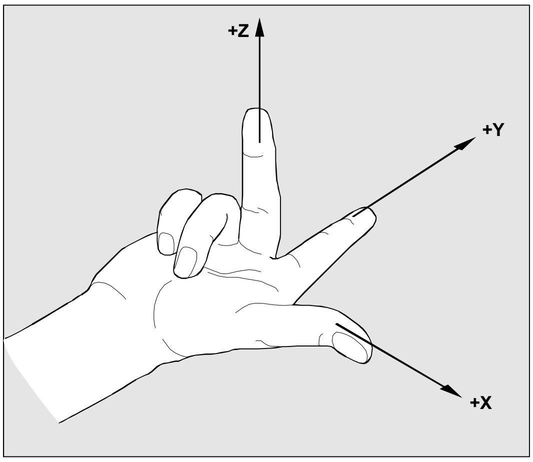
Where the machine coordinate system is used for programming (this is possible with some of the G functions), the physical axes of the machine are addressed directly. No allowance is made for workpiece clamping.



The location of the coordinate system relative to the machine depends on the machine type. The axis directions follow the so-called "three-finger rule" of the right hand (in accordance with DIN 66217).

Standing in front of the machine, the middle finger of the right hand points away from the infeed direction of the main spindle. The following then applies:

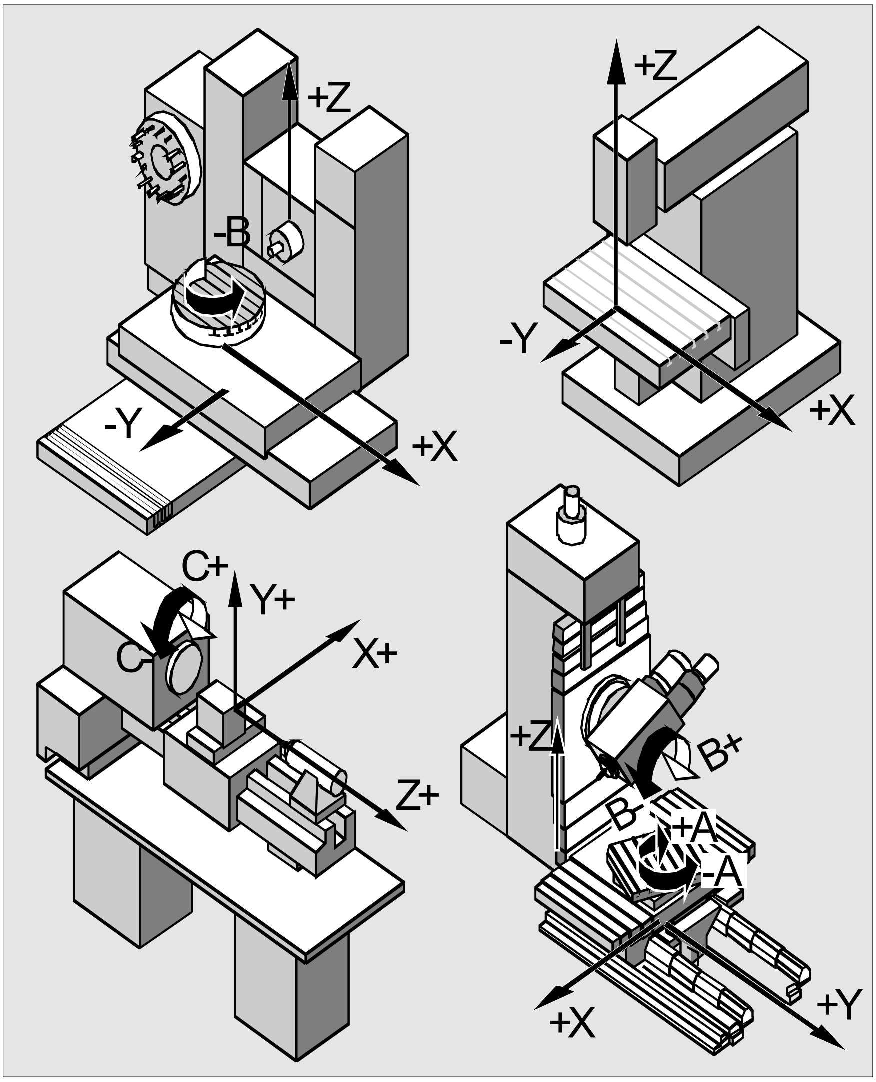
- The thumb points in the +X direction
- The index finger points in the +Y direction
- The middle finger points in the +Z direction



In practice, this can look quite different on different types of machine. The following are examples of machine coordinate systems for various machines.

1.3 Position of coordinate systems

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di



840D
NCU 571

840D
NCU 572
NCU 573

810D

840Di

1.3.3 Basic coordinate system

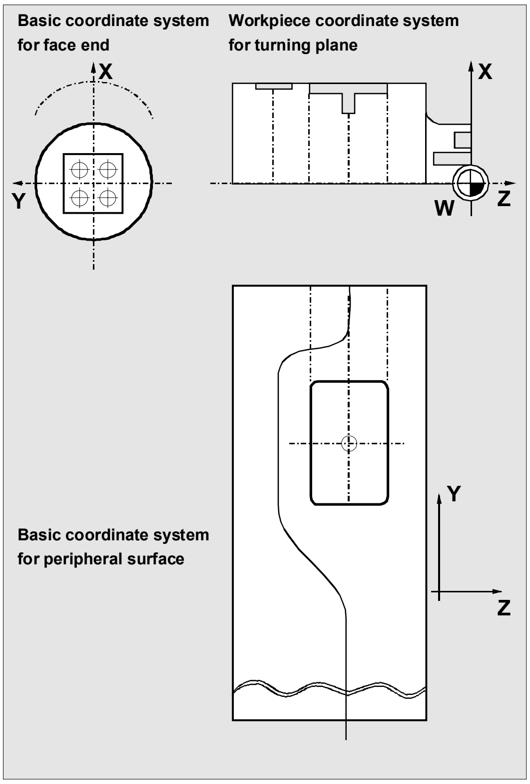
The basic coordinate system is a Cartesian coordinate system, which is mirrored by kinematic transformation (for example, 5-axis transformation or by using Transmit with peripheral surfaces) onto the machine coordinate system.

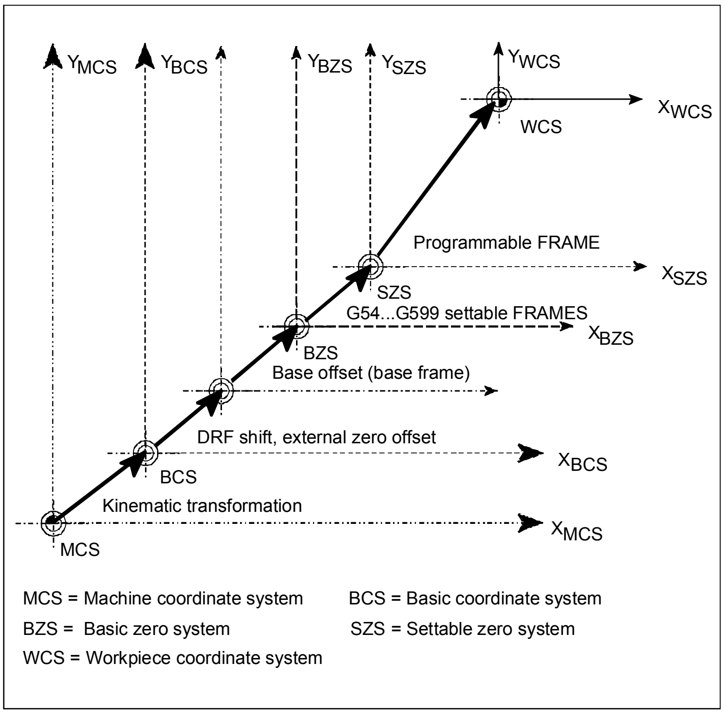
If there is no kinematic transformation, the basic coordinate system differs from the machine coordinate system only in terms of the axes designations.

The activation of a transformation can produce deviations in the parallel orientation of the axes. The coordinate system does not have to be at a right angle.

Zero offset, scaling, etc. are always executed in the basic coordinate system.

The coordinates also refer to the basic coordinate system when specifying the working field limitation.



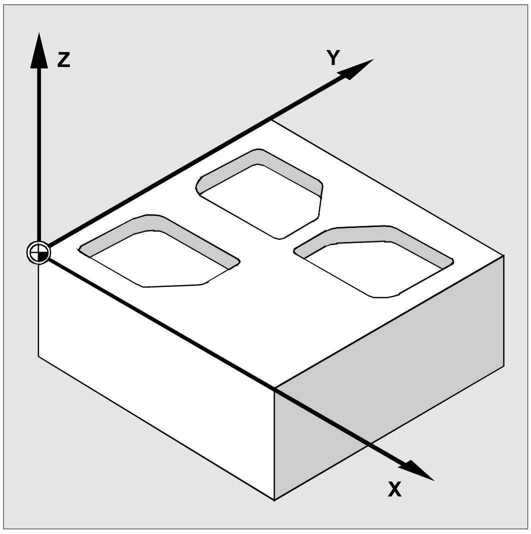


840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.3.4 Workpiece coordinate system

in the workpiece coordinate system. In other words, the data in the NC program refer to the workpiece coordinate system.

The workpiece coordinate system is always a Cartesian coordinate system and assigned to a specific workpiece.



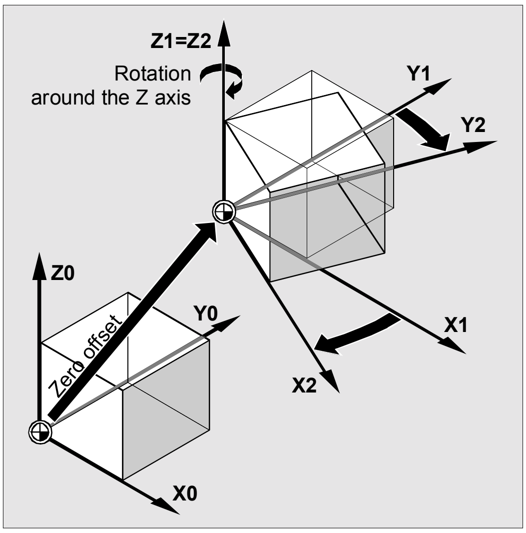
1.3.5 Frame system

The frame is a self-contained arithmetic rule that transforms one Cartesian coordinate system into another Cartesian coordinate system.
It is a:
Spatial description of the workpiece coordinate system

The following components are available within a frame:

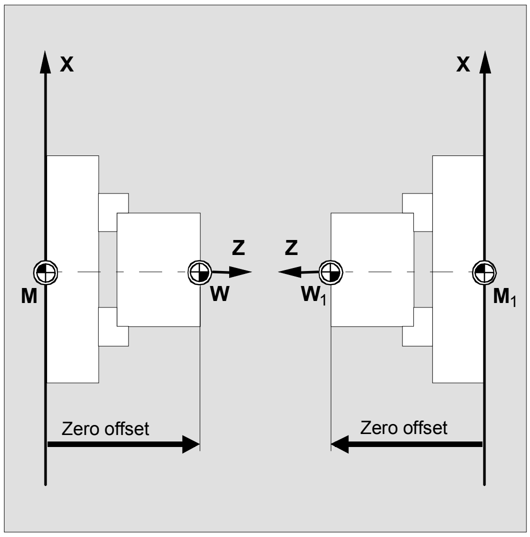
- Zero offset
- Rotate
- Mirror
- Scale

These components can be used individually or in any combination.

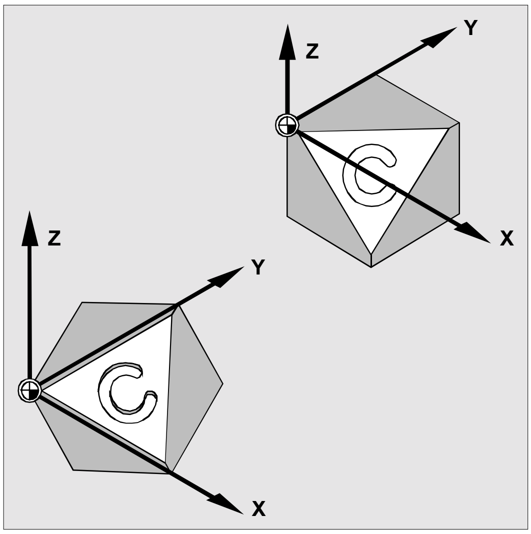


840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Mirroring of the Z axis



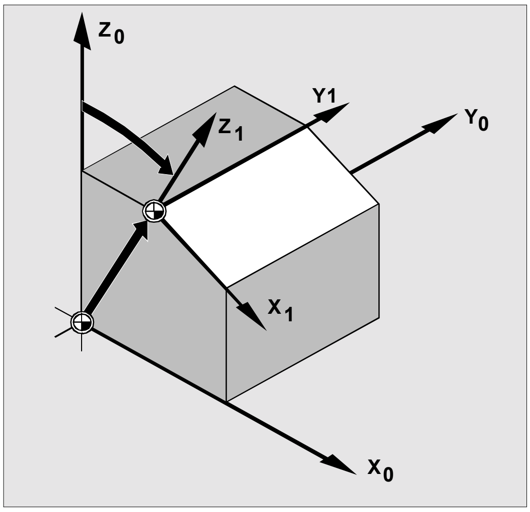
One way of machining inclined contours is to use appropriate fixtures to align the workpiece parallel to the machine axes.



Another way is to generate a coordinate system which is oriented to the workpiece. The coordinate system can be moved and/or rotated with programmable frames.

This enables you to

- move the zero point to any position on the workpiece
- align the coordinate axes parallel to the desired working plane by rotation
- and thus machine surface clamped in inclined positions, produce drill holes at different angles.
- perform multiside machining operations.



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

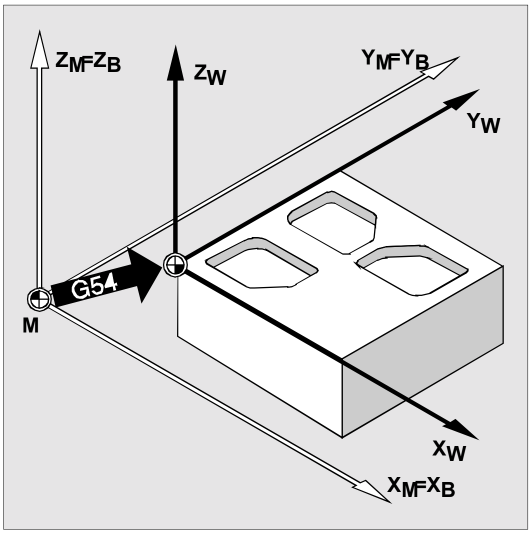
Working plane, tool offsets

The conventions for the working plane and the tool offsets must be observed – in accordance with the machine kinematics – for machining operations in inclined working planes. For further information, please see Section 3.6 "Selection of working plane, G17 to G19".

1.3.6 Assignment of workpiece coordinate system to machine axes

The location of the workpiece coordinate system in relation to the basic coordinate system (or machine coordinate system) is determined by settable frames.

The settable frames are activated in the NC program by means of commands such as G54.

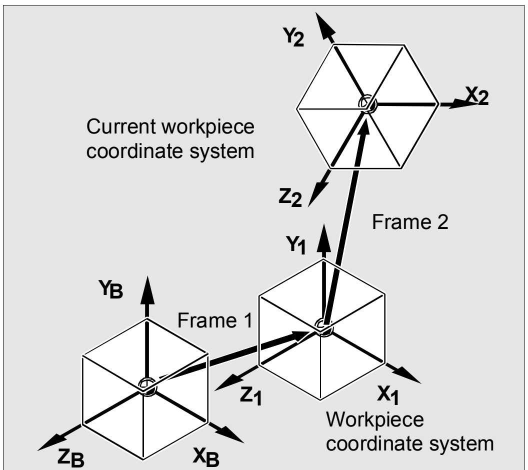


1.3.7 Current workpiece coordinate system

Sometimes it is advisable or necessary to reposition and to rotate, mirror and/or scale the originally selected workpiece zero within a program.

The programmable frames can be used to reposition (rotate, mirror and/or scale) the current zero point at a suitable point in the workpiece coordinate system. You will thus obtain the current workpiece coordinate system.

Several zero offsets are possible in the same program.



Frame 1...settable offset and rotation
Frame 2...programmable offset and rotation

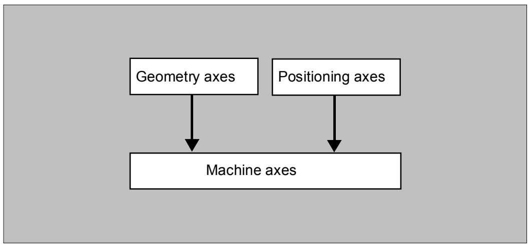
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.4 Axes

A distinction is made between the following types of axis when programming:

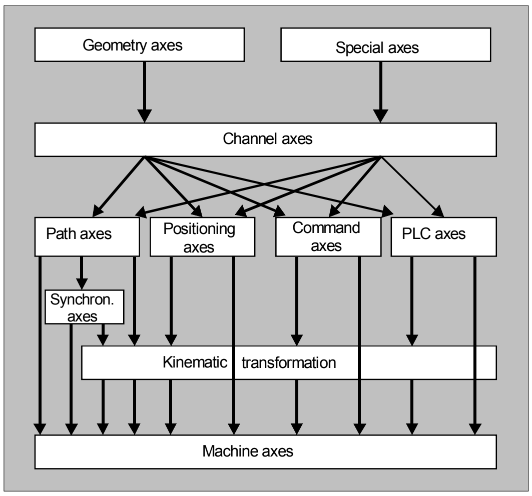
- Machine axes
- Channel axes
- Geometry axes
- Special axes
- Path axes
- Synchronized axes
- Positioning axes
- Command axes
  (motion-synchronized axes)
- PLC axes
- Link axes
- Leading link axes.

Geometry, synchronized and positioning axes are programmed.



Path axes traverse with feedrate F in accordance with the programmed travel commands.

Synchronized axes traverse synchronously to path axes and take the same time to traverse as all path axes.



Positioning axes traverse asynchronously to all other axes. These traversing movements take place independently of path and synchronized movements.

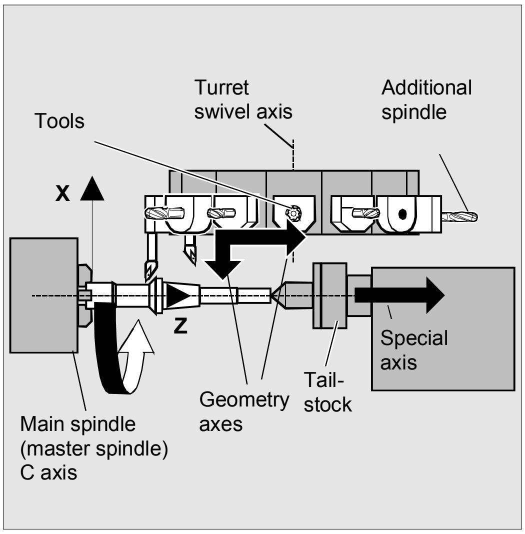
Command axes traverse asynchronously to all other axes. These traversing movements take place independently of path and synchronized movements.

PLC axes are controlled by the PLC and can traverse asynchronously to all other axes. The traversing movements take place independently of path and synchronized movements.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.4.1 Main axes/Geometry axes

The main axes define a right-angled, right-handed coordinate system. Tool movements are programmed in this coordinate system.



In NC technology, the main axes are called geometry axes. This is the term used in this Programming Guide.

For turning machines:
Geometry axes X and Z are used, and sometimes Y.

For milling machines:
Geometry axes X, Y and Z are used.

A maximum of three geometry axes are used for programming frames and the workpiece geometry (contour).

Identifiers: X, Y, Z

The identifiers for geometry and channel axes can be the same, as long as mirroring is possible.

Geometry axis and channel axis names can be the same in any channel in order that the same programs can be executed.
The "**Switchable geometry axes**" function (see Advanced) can be used to alter the geometry axes grouping configured by machine data. Here any geometry axis can be replaced by a channel axis defined as a synchronous special axis.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.4.2 Special axes

In contrast to the geometry axes, no geometrical relationship is defined between the special axes.
Example:
Turret position U, tailstock V

1.4.3 Main spindle, master spindle

The machine kinematics determine which spindle is the main spindle. This spindle is declared the master spindle in the machine data. As a rule, the main spindle is declared the master spindle.

This assignment can be changed with the program command SETMS (spindle number) (see Chapter 7). Special functions such as thread cutting apply to the master spindle.
Identifiers: S or S0

1.4.4 Machine axes

The axis identifiers can be set in the machine data.

Standard identifiers:
X1, Y1, Z1, A1, B1, C1, U1, V1

There are also standard axis identifiers that can always be used:
AX1, AX2, …, AXn

1.4.5 Channel axes

All axes which traverse in a channel.

Identifiers: X, Y, Z, A, B, C, U, V

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.4.6 Path axes

Path axes define the path and therefore the movement of the tool in space.

The programmed feedrate is active for this path.

The axes involved in this path reach their position at the same time. As a rule, these are the geometry axes.

However, default settings define which axes are the path axes and therefore determine the velocity. Path axes can be specified in the NC program with FGROUP (see Chapter 5).

1.4.7 Positioning axes

Positioning axes are interpolated separately, i.e. each positioning axis has its own axis interpolator and its own feedrate.

A distinction is made between positioning axes with synchronization at the block end or over several blocks.

POS axes: Block change occurs at the end of the block when all the path and positioning axes programmed in this block have reached their programmed end point.

POSA axes: The movement of these positioning axes can extend over several blocks.

POSP axes: The movement of these positioning axes for approaching the end position takes place in sections.

You will find further information on POS, POSA and POSP in the section on "Traversing positioning axes, POS, POSA, POSP".

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Additional notes

Positioning axes become synchronized axes if they are traversed without the special POS/POSA identifier.

Continuous-path mode (G64) for path axes is only possible if the positioning axes (POS) reach their final position before the path axes.

Path axes that are programmed with POS/POSA are removed from the path axis grouping for the duration of this block.

Positioning axes are traversed by the NC program or the PLC.

If an axis is to be traversed simultaneously by the NC program and the PLC, an error message appears.

Typical positioning axes are:

- Loaders for workpiece loading
- Loaders for workpiece unloading
- Tool magazine/turret.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1.4.8 Synchronized axes

Synchronized axes traverse synchronously to the path from the start position to the programmed end position.

The feedrate programmed in F applies to all the path axes programmed in the block, but does not apply to synchronized axes. Synchronized axes take the same time as the path axes to traverse.

A synchronized axis can be a rotary axis which is traversed synchronously to the path interpolation.

1.4.9 Command axes

Command axes are started from synchronized actions in response to an event (command). They can be positioned, started and stopped fully asynchronous to the parts program. An axis cannot be moved from the parts program and from synchronized actions simultaneously.
Command axes are interpolated separately, i.e. each command axis has its own axis interpolator and its own feedrate.
References: /FBSY/, Synchronized Actions

1.4.10 PLC axes

PLC axes are traversed by the PLC via special function blocks in the basic program; their movements can be asynchronous to all other axes. The traversing movements take place independently of path and synchronized movements.

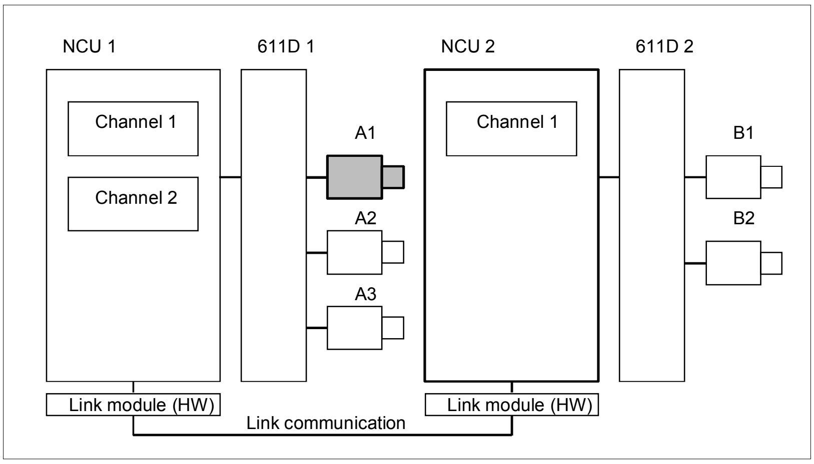
840D
NCU 573

1.4.11 Link axes (SW 5 and higher)

Link axes are axes which are physically connected to another NCU and whose position is controlled from this NCU. Link axes can be dynamically assigned to channels of a **different** NCU. Link axes are not local axes from the perspective of a particular NCU.
The dynamic changing of an assignment to an NCU is based on the **axis container** concept. Axis substitution with GET and RELEASE is not available for link axes from the parts program.

Preconditions:

- The participating NCUs, NCU1 and NCU2, must be connected by means of high-speed communication via the link module.
  References:
  /PHD/, Configuring Manual NCU 571-573.2,
  Link Module
- The axis must be configured appropriately by machine data.
- The link axis option must be installed.



840D
NCU 573

Functionality

The position control is implemented on the NCU on which the axis is physically connected to the drive. This NCU also contains the associated axis VDI interface. The position setpoints for link axes are generated on another NCU and communicated via the NCU link.
The link communication must provide the means of interaction between the interpolators and the position controller or PLC interface. The setpoints calculated by the interpolators must be transported to the position control loop on the home NCU and, vice versa, the actual values must be returned from there back to the interpolators.

For further information about link axes, please refer to
References: /FB/ B3, Multiple Operator Panels and NCUs

Axis container (SW 5 and higher)
An axis container is a circular buffer data structure in which local axes and/or link axes are assigned to channels. The entries in the circular buffer can be **shifted cyclically**.
In addition to the direct reference to local axes or link axes, the link axis configuration in the logical machine axis image also allows references to axis containers.
Such a reference consists of:

- a container number and
- a slot (circular buffer location within the container)

The entry in a circular buffer location contains:

- a local axis **or**
- a link axis

Axis container entries contain local machine axes or link axes from the perspective of an individual NCU.
The entries in the logical machine axis image MN_AXCONF_LOGIC_MACHAX_TAB of an individual NCU are fixed.
The axis container function is described in
References: /FB/ B3, Multiple Operator Panels and NCUs

840D
NCU 573

1.4.12 Leading link axes (SW 6 and higher)

A leading link axis is one that is interpolated by one NCU and utilized by one or several other NCUs as the master axis for controlling slave axes.
An axial position controller alarm is sent to all other NCUs which are connected to the affected axis via a leading link axis.

NCUs that are dependent on the leading link axis can utilize the following coupling relationships with it:

- Master value (setpoint, actual value, simulated master value)

- Coupled motion

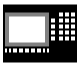
- Tangential follow-up

- Electronic gear (ELG)

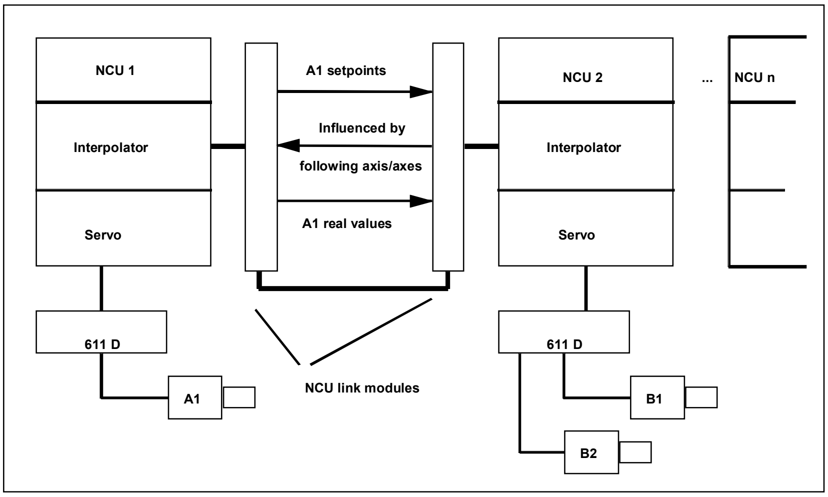
- Synchronous spindle

Preconditions:

- The dependent NCUs, i.e. NCU1 to NCUn (n equals, max. of 8), must be interconnected via the link module for high-speed communication.
  References:
  /PHD/, Configuring Manual NCU 571-573.2, Link Module
- The axis must be configured appropriately by machine data.
- The link axis option must be installed.
- The same interpolation cycle must be configured for all NCUs connected to the leading link axis.



840D
NCU 573



Restrictions:

- A master axis which is leading link axis cannot be a link axis, i.e. it cannot be operated by other NCUs as their home NCU.
- A master axis which is leading link axis cannot be a container axis, i.e. it cannot be addressed alternately by different NCUs.
- A leading link axis cannot be the programmed leading axis in a gantry grouping.
- Couplings with leading link axes cannot be cascaded.
- Axis replacement can only be implemented within the home NCU of the leading link axis.

Programming:

Master NCU:

Only the NCU which is physically assigned to the master value axis can program travel motions for this axis. The travel program must not contain any special functions or operations.

NCUs of slave axes:

The travel program on the NCUs of the slave axes must not contain any travel commands for the leading link axis (master value axis). If it does, an appropriate alarm will be generated.

The leading link axis is addressed in the usual way via channel axis identifiers. The states of the leading link axis can be accessed by means of selected system variables.

840D
NCU 573

System variables:
The following system variables can be used in conjunction with the channel axis identifiers of the leading link axis:

$AA_LEAD_SP ; Simulated master value position
SAA_LEAD_SV ; Simulated master value velocity

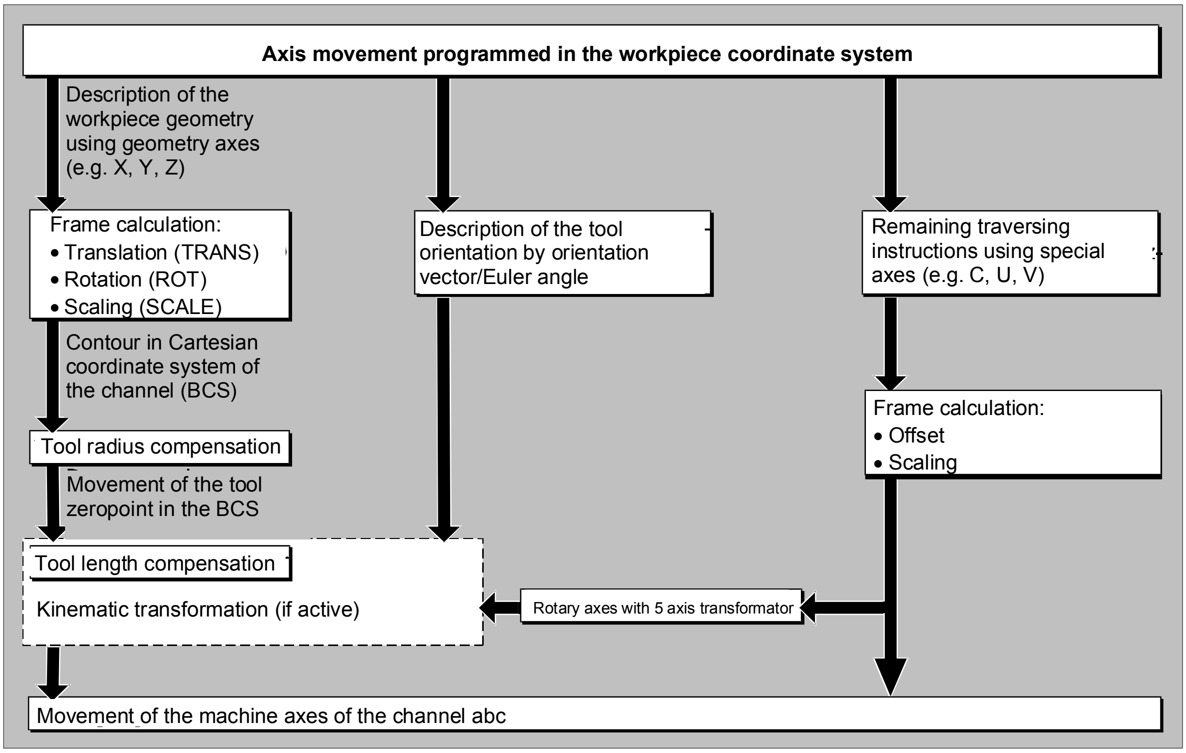
If these system variables are updated by the home NCU of the master axis, the new values are also transferred to any other NCUs who wish to control slave axes as a function of this master axis.

References: /FB/ B3, Multiple Operator Panels and NCUs

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

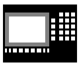
1.5 Coordinate systems and workpiece machining

Relationship between the travel commands from workpiece coordinates and the resulting machine movements

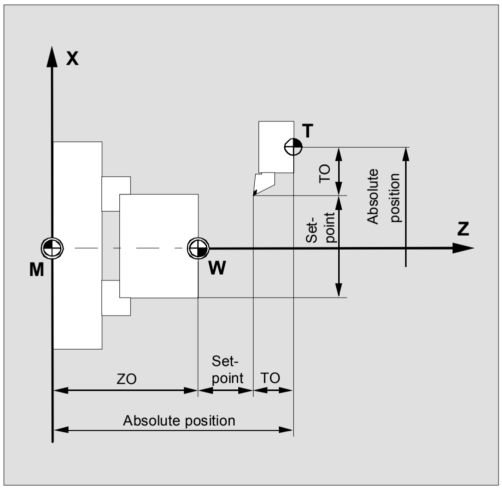


Path calculations

The path calculation determines the distance to be traversed in a block, taking into account all offsets and compensations.

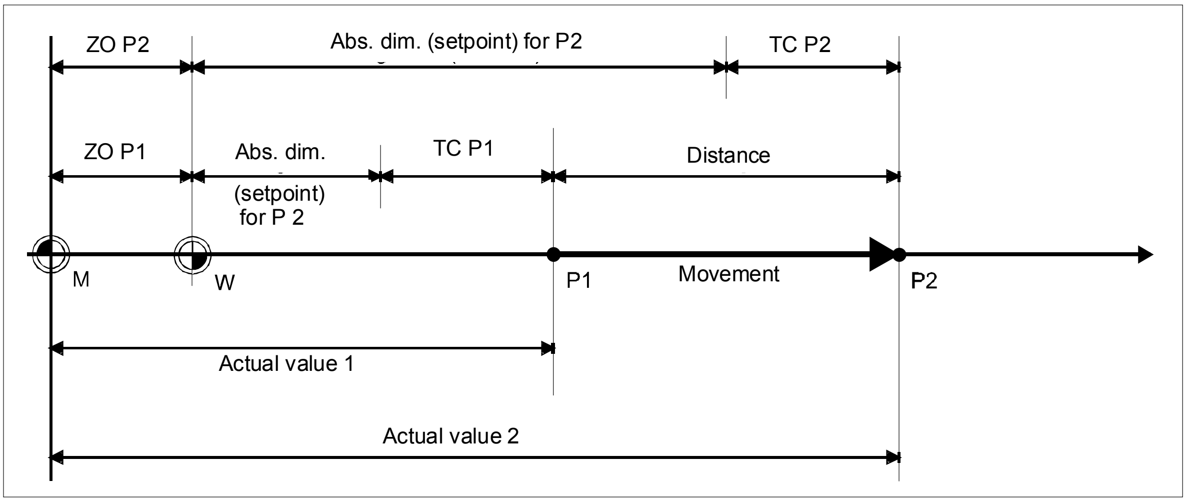

840D
NCU 573

In general:
Distance =
setpoint - actual value + zero offset (ZO) + tool offset (TO)



If a new zero offset and a new tool offset are programmed in a new data block, the following applies:

- With absolute dimensioning:
  Distance = (absolute dimension P2 - absolute dimension P1) + (ZO P2 - ZO P1) + (TO P2 - TO P1).
- With incremental dimensioning:
  Distance = incremental dimension + (ZO P2 - ZO P1) + (TO P2 - TO P1).



■

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Notes

Fundamental Principles of NC Programming

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

2.1 Structure and contents of an NC program

DIN 66025 is the guideline for designing a parts program.

An (NC/part) program consists of a sequence of NC **blocks** (see table below). Each data block represents one machining step. Instructions are written in the blocks in the form of **words**. The last block in the execution sequence contains a special word for the **end of program: M2**, **M17** or **M30.**

Block	Word	Word	Word	...	;Comment
Block	N10	G0	X20	...	;1st block
Block	N20	G2	Z37	...	;2nd block
Block	N30	G91	...	...	;...
Block	N40	...	...	...	
Block	N50	M30	...	...	;End of program (last block)

Program names

Each program has a different name; the name can be chosen freely during program creation (except for punch tape format), taking the following conditions into account:

- The first two characters must be letters (or a letter with an underscore character)
- Otherwise: letters or numerals

Example:
```
_MPF100 or
SHAFT or
SHAFT_2
```

Only the first **24** characters of a program identifier are displayed on the NC.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Punch tape format

File names:

1. File names can contain the characters `0...9, A...Z, a...z` or _and may be up to 24 characters in length.
2. File names must have a 3-digit identifier (_xxx).
3. Data in punch tape format can be created externally or modified using an editor. The name of a file which is stored internally in the NC memory begins with "_N_".
   A file in punch tape format begins with **%**<name>, "**%**" must appear in the first column of the first line.

Examples:

`%_N_SHAFT123_MPF` = parts program `SHAFT123`

or

`%flange3_MPF` = parts program `flange3`

For further information on downloading, creating and storing parts programs, please refer to:
/BA/, Operator's Guide, in the sections on the "Program" and "Services" user areas.

2.2 Language elements of the programming language

Character set

The following characters are available for writing NC programs:

Upper case

`A, B, C, D, E, F, G, H, I, J, K, L, M,`
`N, (O), P, Q, R, S, T, U, V, W, X, Y, Z`

Please note:
Take care to differentiate between the letter "O" and the numeral "0".

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Lower case

a, b, c, d, e, f, g, h, i, j, k, l, m,
n, o, p, q, r, s, t, u, v, w, x, y, z

Numerals

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

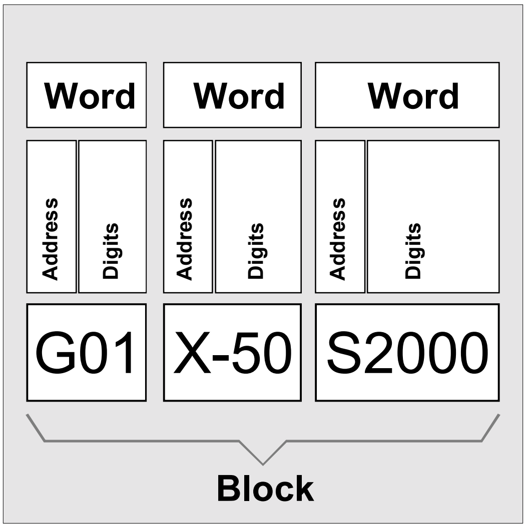
No distinction is made between upper and lower case letters.

Special characters

%	Program start character (used only for writing programs on an external PC)
(	For bracketing parameters or expressions
)	For bracketing parameters or expressions
[	For bracketing addresses or indexes
]	For bracketing addresses or indexes
<	Less than
>	Greater than
:	Main block, label suffix, chain operator
=	Assignment, part of equation
/	Division, block suppression
*	Multiplication
+	Addition
-	Subtraction, minus sign
"	Double quotation marks, identifier for character string
´	Single quotation marks, identifier for special numerical values: hexadecimal, binary
$	System variable identifiers
_	Underscore, belonging to letters
?	Reserved
!	Reserved
.	Decimal point
,	Comma, parameter separator
;	Comment start
&	Format character, same effect as space character
L_F	Block end
Tab character	Separator
Space character	Separator (blank)

Non-printable special characters are treated like blanks.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di



Words

NC programs are made up of blocks and each block is made up of words.

A word in the NC language consists of an address character and a digit or sequence of digits representing an arithmetic value.

The address character of the word is usually a letter. The sequence of digits can contain a leading sign and decimal point. The leading sign always appears between the address letter and the sequence of digits. The positive leading sign (+) does not have to be specified.

Blocks and block format

An NC program consists of individual blocks. A block generally consists of (several) words.
A block should contain all the data required for performing an operation step and is terminated with the character "L_F" (LINE FEED = new line).

⚠ *The characters "L_F" character does not have to be inserted manually, it is generated automatically when you change lines.*

Block length

A block may contain

- up to SW 3.x a maximum of **242** characters
- **SW 4 and higher** a maximum of **512 characters** (including the comment and end-of-block character "L_F").

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Three blocks of up to 66 characters each are normally displayed in the current block display on the screen. Comments are also displayed. Messages are displayed in a separate message window.

Word sequence in blocks
In order to keep the block structure as clear as possible, the words in a block should be arranged as follows:

Example:
`N10 G… X… Y… Z… F… S… T… D… M… H…`

Address	*Definition*
`N`	Address of block number
`10`	Block number
`G`	Preparatory function
`X,Y,Z`	Positional data
`F`	Feed
`S`	Speed
`T`	Tool
`D`	Tool offset number
`M`	Miscellaneous function
`H`	Auxiliary function

Some addresses can be used repeatedly within a block (e.g.: `G…, M…, H…`).

Main block/subblock
There are two types of blocks:

- **Main blocks** and
- **Subblocks.**

The main block must contain all the words necessary to start the operation sequence in the program section beginning with the main block.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Main blocks can be contained in both main programs and subprograms. The control does not check whether a main block contains all the necessary information.

Block number
Main blocks are identified by a main block number. A main block number consists of the character ":" and a positive integer (block number). The block number always appears at the start of a block.

Main block numbers must be unique within a program to achieve an unambiguous result when searching.

Example: `:10 D2 F200 S900 M3`

Subblocks are identified by a subblock number. A subblock number consists of the character "N" and a positive integer (block number). The block number always appears at the start of a block.

Example:
```
N20 G1 X14 Y35
N30 X20 Y40
```

Subblock numbers must be unique within a program in order to achieve an unambiguous result when searching.

The order of the block numbers is arbitrary, however increasing block numbers are recommended.

You can also program NC blocks without block numbers.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Addresses

Addresses are fixed or variable identifiers for axes (X, Y, ...) spindle speed (S), feed (F), circle radius (CR), etc.

Example:

`N10 X100`

Important addresses

Address	*Meaning (default setting)*	*Remarks*

`A=DC(...)`
`A=ACP(...)`
`A=ACN(...)`
	Rotary axis	variable

`ADIS`	Rounding clearance for path functions	fixed

`B=DC(...)`
`B=ACP(...)`
`B=ACN(...)`
	Rotary axis	variable

`C=DC(...)`
`C=ACP(...)`
`C=ACN(...)`
	Rotary axis	variable

`CHR=...`	Chamfer the contour corner	fixed
`D...`	Tool number	fixed
`F...`	Feed	fixed
`FA[axis]=...` or `FA[spindle]=...` or `[SPI(spindle)]=...`	Axial feed (only if spindle no. defined by variable)	fixed
`G...`	Preparatory function	fixed
`H...` `H=QU(...)`	Auxiliary function Auxiliary function without read stop	fixed
`I...`	Interpolation parameters	variable
`J...`	Interpolation parameters	variable
`K...`	Interpolation parameters	variable
`L...`	Subprogram call	fixed
`M...` `M=QU(...)`	Miscellaneous function Miscellaneous fct. w/o read stop	fixed
`N...`	Subblock	fixed
`OVR=...`	Path override	fixed
`P...`	Number of program passes	fixed

840D NCU 571	840D NCU 572 NCU 573	810D	840Di

Address	*Meaning (default setting)*	*Remarks*
`POS[axis]=...`	Positioning axis	fixed
`POSA[axis]=...`	Positioning axis across block boundary	fixed
`SPOS=...` `SPOS[n]=...`	Spindle position	fixed
`SPOSA=...` `SPOSA[n]=...`	Spindle position across block boundary	fixed
`Q...`	Axis	variable

`R0=... to Rn=...`
 	
`R...`
	• Arithmetic parameter, n can be set via MD (default 0-99)
	• Axis (SW 5.1 and higher)
		fixed
		
		variable

`RND`	Round contour corner	fixed
`RNDM`	Round contour corner (modal)	fixed
`S...`	Spindle speed	fixed
`T...`	Tool number	fixed
`U...`	Axis	variable
`V...`	Axis	variable
`W...`	Axis	variable

`X...`
`X=AC(...)`
`X=IC(...)`
	Axis
	" absolute
	" incremental
		variable

`Y...`
`Y=AC(...)`
`Y=IC(...)`
	Axis	variable

`Z...`
`Z=AC(...)`
`Z=IC(...)`
	Axis	variable

`AR+=...`	Aperture angle	variable
`AP=...`	Polar angle	variable
`CR=...`	Circle radius	variable
`RP=...`	Polar radius	variable
`:...:`	Main block	fixed

"fixed"
These address names are available for a specific function.

Machine manufacturer (MH2.1)

"variable"
These addresses can be assigned another name via the machine data.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Modal/non-modal addresses

Modal addresses remain valid with the programmed value (in all subsequent blocks) until a new value is programmed at the same address.

Non-modal addresses only apply in the block in which they were programmed.

Example:

```
N10 G01 F500 X10
N20 X10              ;Feed is effective until
                      a new value is entered.
```

Addresses with axial extension

In addresses with axial extension, an axis name is inserted in square brackets after the address. The axis name assigns the axis.

Example: `FA[U]=400;`
Axis-specific feed for U axis

Extended addresses

Extended address notation enables a larger number of axes and spindles to be organized in a system. An extended address is composed of a numeric extension or a variable identifier enclosed in square brackets and an arithmetic expression assigned with an "=" sign.

Example:

```
X7          ;No "=" required, 7 is a value, but the "=" ;character is also possible here
X4=20       ;Axis X4 ("=" required)
CR=7.3      ;2 letters ("=" required)
S1=470      ;Speed for 1st spindle 470rpm
M3=5        ;Spindle stop for 3rd spindle
```

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

The extended address notation is only permitted for the following direct addresses:

X, Y, Z, …	Axis addresses
I, J, K	Interpolation parameters
S	Spindle speed
SPOS, SPOSA	Spindle position
M	Miscellaneous functions
H	Auxiliary functions
T	Tool number
F	Feed

The number (index) in extended address notation can be substituted by a variable for M, H and S addresses and SPOS and SPOSA. The variable identifier is enclosed in square brackets.

Example:

```
S[SPINU]=470   ;Speed for the spindle whose number is stored in the variable SPINU
M[SPINU]=3     ;Clockwise rotation for the spindle whose number is stored in the variable SPINU
T[SPINU]=7     ;Selection of the tool for the spindle whose number is stored in the variable SPINU
```

Fixed addresses

The following addresses are set permanently:

Address	*Meaning (default setting)*
D	Tool edge number
F	Feed
G	Preparatory function
H	Auxiliary function
L	Subprogram call
M	Special function
N	Subblock
P	Number of program passes
R	Arithmetic parameter
S	Spindle speed
T	Tool number
:	Main block

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Example for programming:
`N10 G54 T9 D2`

Fixed addresses with axis extension

Address	*Meaning (default setting)*
AX	Axis value (variable axis programming)
ACC	Axial acceleration
FA	Axial feed
FDA	Axis feedrate for handwheel override
FL	Axial feed limit
IP	Interpolation parameter (variable axis programming)
OVRA	Axial override
PO	Polynomial coefficient
POS	Positioning axis
POSA	Positioning axis across block boundary

Example: `N10 POS[X]=100`
When programming with the axis extension, the axis to be traversed is enclosed in square brackets.

You will find a complete list of all fixed addresses in the Appendix.

Settable addresses
Addresses can be defined either as an address letter (with numerical extension if necessary) or as freely selected identifiers.

Variable addresses must be unique within the control, i.e. the same identifier name may not be used for different address types.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

A distinction is made between the following address types:

- Axis values and end points
- Interpolation parameters
- Feeds
- Approximate positioning criteria
- Measurements
- Axis and spindle response
- …

Variable address letters are:
`A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z`

The user can change the names of the variable addresses in the machine data.

Example:
`X1, Y30, U2, I25, E25, E1=90, …`

The numeric extension has one or two digits and is always positive.

Address identifier
The address notation can be expanded by adding further letters.

Example:

`CR`	e.g. for circle radius
`XPOS`	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Operators/mathematical functions

+	Addition
-	Subtraction
*	Multiplication
/	Division **NOTICE**: (Type INT)**/**(Type INT)**=**(Type REAL); e.g. 3/4 = 0.75
DIV	Division, for variable type INT and REAL **NOTICE**: (Type INT)DIV(Type INT)**=**(Type INT); e.g. 3 DIV 4 = 0
MOD	Modulo division (only type INT) produces remainder of INT division, e.g. 3 MOD 4=3
:	Chain operator (for FRAME variables)

Sin()	Sine
COS()	Cosine
TAN()	Tangent
ASIN()	Arcsine
ACOS()	Arccosine
ATAN2()	Arctangent2
SQRT()	Square root
ABS()	Absolute number
POT()	2nd power (square)
TRUNC()	Truncate to integer
ROUND()	Round to integer
LN()	Natural logarithm
EXP()	Exponential function

Comparison and logic operators

==	Equal to
<>	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

AND	AND
OR	OR
NOT	Negation
XOR	Exclusive OR

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

In arithmetic expressions, the execution order of all the operators can be specified by parentheses, in order to override the normal priority rules.

Value assignments

Values can be assigned to the addresses. The method of value assignment depends on the type of address identifier.

An "=" sign must be inserted between the address identifier and the value if

- the address identifier consists of more than one letter, or
- the value consists of more than one constant.

The "="-sign can be omitted if the address identifier is a single letter and the value consists of only one constant. Leading signs are allowed and separators are permitted after the address letter.

Examples:

```
X10                 ;Value assignment (10) to address X, "="  not required
X1=10               ;Value assignment (10) to address (X) with numeric extension (1), "="
                    required
FGROUP(X1, Y2)      ;Axis names from passed parameters
AXDATA[X1]          ;Axis name as an index when accessing axis data
AX[X1]=10           ;Indirect axis programming
X=10*(5+SIN(37.5))  ;Value assignment by means of a numeric expression, "=" required
```

A numeric extension must always be followed by one of the special characters "=", "(", "[")"]", "," or an operator in order to distinguish an address name with numeric extension from an address letter with a value.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Identifiers

Identifiers can also be used to describe words (in compliance with DIN 66025). The identifiers have the same meaning as the words within an NC block. Identifiers must be unique. Identical identifiers must not be used for different objects.

Identifiers can stand for:

- Variables
  - System variables
  - User variables
- Subprograms
- Vocabulary words
- DIN addresses with several letters
- Jump labels

Structure

The identifiers are composed of up to 32 characters. The following characters may be used:

- Letters
- Underscores
- Numerals

The first two characters must be letters or underscores, separators must not be programmed between the individual characters (see the following pages).

Example: `CMIRROR, CDON`

Reserved vocabulary words must not be used as identifiers. Separators are not permitted between the individual characters.

Number of characters for each identifier:

- Program names: 24 characters
- Axis identifiers: 8 characters
- Variable identifiers: 31 characters

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Rules for allocating identifiers

The following rules are provided in order to avoid identifier collisions:

- All identifiers which begin with "CYCLE" or "_" are reserved for SIEMENS cycles.
- All identifiers which begin with "CCS" are reserved for SIEMENS compile cycles.
- User compile cycles begin with "CC".
- We recommend that users select identifier names which either begin with "U" (User) or contain the underscore symbol, because these are not used by the system or compile cycles or SIEMENS cycles.

Further reserved identifiers

- The identifier "RL" is reserved for conventional turning machines.
- All identifiers beginning with "E_ " are reserved for EASYSTEP programming.

Variable identifiers

In variables used by the system, the first letter is replaced by the "$" character. This character may not be used for user-defined variables.

Examples (see Programming Guide "Advanced"):
`$P_IFRAME, $P_F`

Leading zeroes are ignored in variables with numeric extensions (i.e. R01 is interpreted as R1). Separators are allowed before a numeric extension.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Array identifiers

The rules for elementary variables also apply to array identifiers. It is possible to address arithmetic variables as arrays.

Example: `R[10]=…`

Data types

A variable can contain a numeric value (or several) or a character (or several), e.g. an address letter.

The data type permitted for the variable is determined when the variable is defined. The data type for system variables and predefined variables is fixed.

Elementary variable types/data types are:

Type	*Definition*	*Value range*
`INT`	Integers with leading sign	$\pm(2^{31} - 1)$
`REAL`	Real numbers (fractions with decimal point, LONG REAL according to IEEE)	$\pm(10^{-300} \ldots 10^{+300})$
`BOOL`	Boolean values: TRUE (1) and FALSE (0)	1, 0
`CHAR`	1 ASCII character specified by the code	0 … 255
`STRING`	Character string, number of characters in […], maximum of 200 characters	Sequence of values with 0 … 255
`AXIS`	Axis names (axis addresses) only	Any axis identifiers in the channel
`FRAME`	Geometrical parameters for translation, rotation, scale and mirror	

Identical elementary types can be combined in arrays. Up to two-dimensional arrays are possible.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Constants

Integer constants:
Integer with or without leading sign, e.g. for assigning a value to an address

Examples:

```
X100        ;Assignment of the value +100 to address X
X-100       ;Assignment of the value –100 to address X
```

Real constants:
Real number, e.g. with decimal point, with or without leading sign, e.g. for assigning a value to an address

Example:

```
X10.25      ;Assignment of the value +10.25 to address X
X-10.25     ;Assignment of the value –10.25 to address X
X0.25       ;Assignment of the value +0.25 to address X
X.25        ;Assignment of the value +0.25 to address X without leading "0"
X=-.1EX-3   ;Assignment of the value –0.1*10-3 to address X
```

If, in an address which permits decimal point input, more decimal places are specified than actually provided for the address, then they are rounded to fit the number of places provided.

X0 cannot be replaced with X.

Example: Do not replace G01 X0 with G01 X!

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Hexadecimal constants

Constants can also be interpreted in hexadecimal format. The letters "A" to "F" stand for the digits 10 to 15.

Hexadecimal constants are enclosed in single quotation marks and start with the letter "H", followed by the value in hexadecimal notation. Separators are allowed between the letters and digits.

Example for machine data (see also "Advanced"):

```
$MC_TOOL_MANAGEMENT_MASK='H3C7F'      ;Assignment of hexadecimal values to machine data
```

The maximum number of characters is limited by the value range of the integer data type.

Binary constants

Constants can also be interpreted in binary format. In this case, only the digits "0" and "1" are used.

Binary constants are enclosed in single quotation marks and start with the letter "B", followed by the binary value. Separators are allowed between the digits.

Example for machine data (see also "Advanced"):

```
$MN_AUXFU_GROUP_SPEC='B10000001'      ;Assignment of binary constants to machine data
                                      Bits 0 and 7 are set
```

The maximum number of characters is limited by the value range of the integer data type.

840D NCU 571	840D NCU 572 NCU 573	810D	840Di

Program section

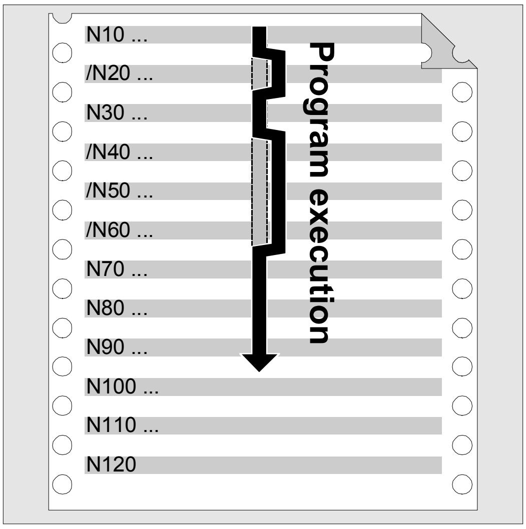
A program section consists of a main block and several subblocks.

Examples:

```
:10 D2 F200 S900 M3
N20 G1 X14 Y35
N30 X20 Y40
N40 Y-10
...
N100 M30
```

Skipping blocks

Blocks which are not to be executed on every program pass can be skipped (e.g. positioning program). Blocks which are to be skipped are marked with an oblique "/" in front of the block number. Several consecutive blocks can also be skipped. The instructions in the skipped blocks are not executed; the program continues with the next block which is not skipped.



Examples:

`N10 …`	;is executed
`/N20 …`	;skipped
`N30 …`	;is executed
`/N40 …`	;skipped
`N70 …`	;is executed

SW 5 and higher

Up to eight skip levels can be programmed. Only one skip level can be specified per NC block:

/ ...	;Block is skipped (1st skip level)
/0 ...	;Block is skipped (1st skip level)
/1 N010...	;Block is skipped (2nd skip level)
/2 N020...	;Block is skipped (3rd skip level)
...	
/7 N100...	;Block is skipped (8th skip level)

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

SW 6.3 and higher

Up to ten skip levels can be programmed. Only one skip level can be specified per NC block:

/8 N080...	;Block is skipped (9th skip level)
/9 N090...	;Block is skipped (10th skip level)

Machine manufacturer (MH2.2)

The number of skip levels that can be used depends on a display MD.

Block skipping of levels /0 to /9 is activated by an operator action (see /BA/ Operator's Guide, program control menu in Machine operating area) or by the programmable controller.

System and user variables can also be used in conditional jumps in order to control program execution.

Jump destinations (labels)

Labels can be defined to jump within a program.

You will find further information in the Programming Guide Advanced.

Label names are allocated with at least two and up to 32 characters (letters, digits, underscore). The first two characters must be letters or underscores. The label name is followed by a colon (":").

Labels must be unique within a program.

Labels always appear at the start of a block. If a program number exists, the label appears immediately after the block number.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Comments

To make NC programs easier to understand for other users and programmers, it is advisable to insert meaningful comments in the program.

Comments are appended to the end of a block and are separated from the program section of the NC block by a semicolon (";").

Examples:

```
N10 G1 F100 X10 Y20      ; Comments to explain the NC block
```

or

```
N10                      ; G&S Co., order no. 12A71
N20                      ; Program written by Bob Miller, Dept. TV 4, on 21.11.94
N50                      ; Part no. 12, housing for pump type TP23A
```

Comments are stored and appear in the current block display when the program is running.

Programming messages

Messages can be programmed to provide the user with information about the current machining situation during program execution.

A message in an NC program is generated when the message text is typed after vocabulary word "MSG" in round parentheses "()" and quotation marks.

A message can be cleared by programming "MSG ()".

Example:

```
N10 MSG ("Roughing the contour")          ;Activate message
N20 X… Y…
N …
N90 MSG ()                                ;Clear message from N10
```

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

A message text can be **up to 124** characters long and is displayed in two lines (2*62 characters).
Contents of variables can also be displayed in message texts.

Examples:

```
N10 R12=$AA_IW [X]                              ;Current position of the X axis in R12
N20 MSG (″check position of X axis″<<R12<<″″)
N …
N90 MSG ()                                      ;Clear message from N20
```

or

```
N20 MSG (″check position of X axis″<<$AA_IW[X]<<″″)
```

Setting alarms

You can also set alarms in addition to messages in an NC program. Alarms are displayed in a separate field on the screen display. An alarm is associated with a reaction on the control which depends on the alarm category.

Alarms are programmed by writing the vocabulary word "SETAL" followed by the alarm number enclosed in brackets.

The valid range for alarm numbers lies between 60,000 and 69,999, whereby 60,000 to 64,999 are reserved for SIEMENS cycles and 65,000 to 69,999 are available to the user.

Alarms are always programmed in a separate block.

Example:

```
N100 SETAL (65000)                              ;Set alarm no. 65,000
```

You will find a list of reactions associated with specific alarms in the Installation and Start-up Guide.

The alarm text must be configured in the MMC.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

2.3 Programming a sample workpiece

Planning the machining sequence

The actual programming of the individual operation steps in the NC language generally only represents a small proportion of the work in the development of an NC program.

Programming of the actual instructions should be preceded by the planning and preparation of the operation steps. And the more accurately you consider how the NC program is to be structured and organized, the faster and easier it will be to produce a complete program which is clear and free of errors.

Clearly structured programs are a particular advantage if you need to make changes at a later date.

Since different parts do not all look the same, it is naturally inadvisable to develop every program according to the same method. However, certain procedures apply in most cases and these are presented on the following pages in the form of a checklist.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

1. Prepare the workpiece drawing

- Define the workpiece zero
- Sketch in the coordinate system
- Calculate any missing coordinates

2. Define machining sequence

- What tools are used when and to machine which contour?
- In what order are the individual elements of the workpiece manufactured?
- Which individual elements are repeated (or also rotated) and should therefore be stored in a subprogram?
- Do these or similar contour sections already exist in other parts programs or subprograms which could be used here?
  Where is it advisable or necessary to perform zero offset, rotation, mirroring or scaling (frame concept)?

3. Create a machining plan

Define all the machining processes in steps, e.g.:

- Rapid traverse movements for positioning
- Tool change
- Tool retraction for recalibration
- Switching the spindle and coolant on and off
- Calling tool data
- Infeed
- Path correction
- Approach contour
- Retract from contour
- etc.

4. Translate the work steps into the programming language

Enter each individual step in an NC block or blocks

5. Combine all the individual steps in a program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

2.4 First programming example for milling application

Please proceed on the NC as described below to verify the following programming example (see Operator's Guide):

- Create a new parts program (name)
- Edit parts program
- Select parts program
- Activate single block
- Start parts program

Alarms can occur during program verification. These alarms have to be reset first.

Machine manufacturer (MH2.3)

The MD settings must be defined correctly before the program can run on the machine.
References: /FB/ K2, "Axes, Coordinate Systems,.."

Programming example

`_MILL1_MPF`	;Sample program
`N10  MSG("THIS IS MY NC PROGRAM")`	;MSG = Message output in an alarm line
`:10  F200 S900 T1 D2 M3`	;Feed, spindle, tool, ;tool offset, spindle clockwise
`N20  G0 X100 Y100`	;Rapid traverse to position
`N30  G1 X150`	;Rectangle with feed, straight line in X
`N40  Y120`	;Straight line in Y
`N50  X100`	;Straight line in X
`N60  Y100`	;Straight line in Y
`N70  G0 X0 Y0`	;Return rapid traverse movement
`N100 M30`	;End of block

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

2.5 Second programming example for milling application

This programming example contains surface and side milling, as well as drilling.

- The workpiece is intended for machining on a **vertical milling machine**.
- The dimensions are in inches.

Machine manufacturer (MH2.4)

The MD settings must be defined correctly before the program can run on the machine.
References: /FB/ K2, "Axes, Coordinate Systems,.."

Programming example

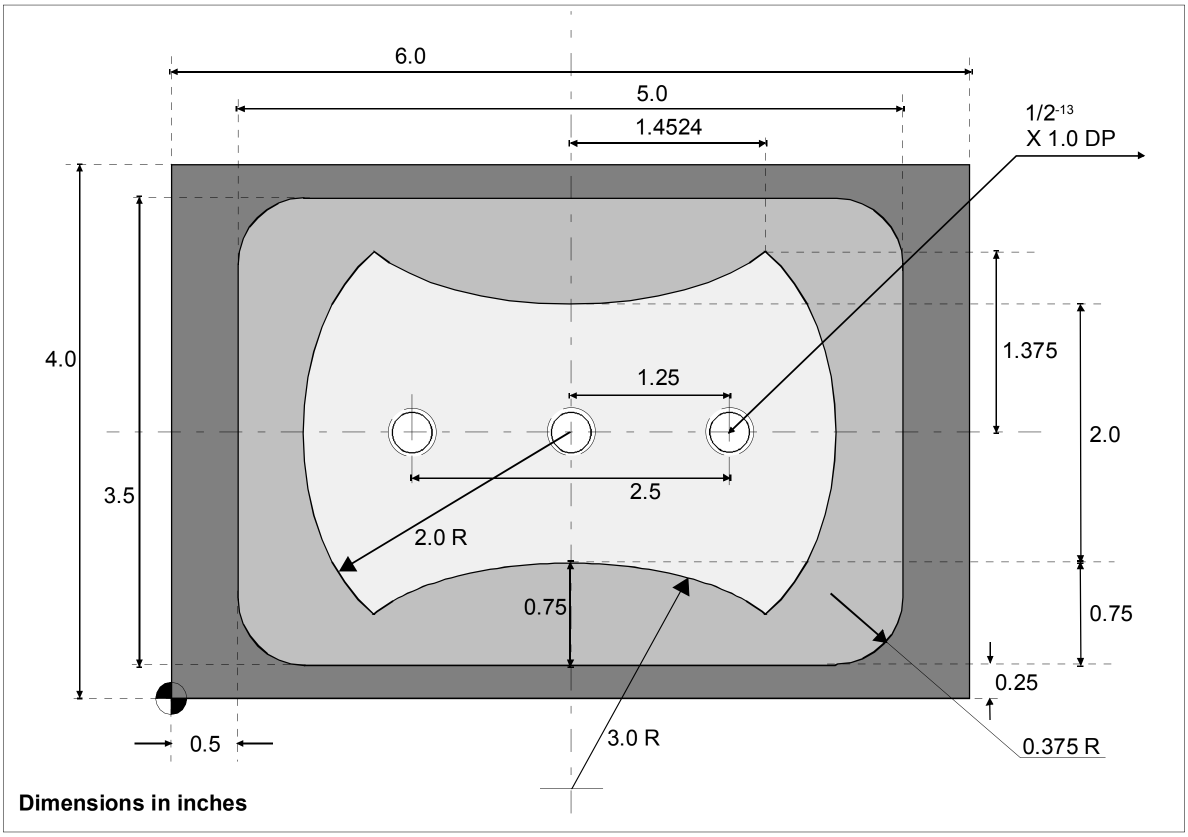
```
%_N_RAISED_BOSS_MPF
N005  MSG ("Traverse axes to tool change location")
N010  START01:SUPA G0 G70 Z0 D0
N015  SUPA X0 Y0
;********************Tool change********************
N020  MSG ("Tool change active")
N025  T1 M6                    ; d = 3 inch face cutter
N030  MSG ()                   ; clears the message from block N020
N035  MSG ("Face milling Z=0 workpiece surface")
N040  G0 G54 X-2 Y.6 S800 M3 M8
N045  Z1 D1
N050  G1 Z0 F50
N055  X8 F25
N060  G0 Y3.5
N065  G1 X-2
N070  SUPA G0 Z0 D0 M5 M9
;********************Tool change********************
N075  T2 M6                    ; d = 1 inch face cutter
MSG ("Side machining")
N080  G0 X-1 Y.25 S1200 M3 M8
N085  Z1 D1
N090  G1 Z-.5 F50
N095  G42 X.5 F30
N100  X5.5 RNDM=-.375          ; modal rounding. Radius=0.375
N105  Y3.625
```

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

```
N110  X.5
N115  Y.25
N120  X=IC(.375) RNDM=0                ; required for edge rounding
N125  G40 G0 Y-1 M5 M9                 ; rapid traverse to delete position
N130  Z1
N135  X-1 Y0
N140  Z-.25
,********************Continue to use 1-inch mill****************
MSG ("Side Cut Top Boss")
N145  G01 G41 X1 Y2
N150  G2 X1.5476 Y3.375 CR=2
N155  G3 X4.4524 CR=3
N160  G2 Y.625 CR=2
N165  G3 X1.5476 CR=3
N170  G2 X1 Y2 CR=2
N175  G0 G40 X0
N180  SUPA G0 Z0 D0 M5 M9              ; Z approaches tool change location
N185  SUPA X0 Y0                       ; X and Y to the tool change location
;********************Tool change*******************
N190  T3 M6                            ; 27/64 drill
MSG ("Drill 3 holes")
N195  G0 X1.75 Y2 S1500 M3 M8          ; approach first drill hole
N200  Z1 D1
N205  MCALL CYCLE81 (1,0,.1,-.5,)
N207  X1.75                            ; drill first hole
N210  X3                               ; drill second hole
N215  X4.25                            ; drill third hole
N220  MCALL
N221  SUPA Z0 D0 M5 M9  ; Delete modal call. Z axis traverses to machine zero
N225  SUPA X0 Y0
MSG ()
N230  M30                              ; end of program
```

2.5 Second programming example for milling application

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di



Dimension drawing of workpiece "The Raised Boss" (not to size).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

2.6 Programming example for turning application

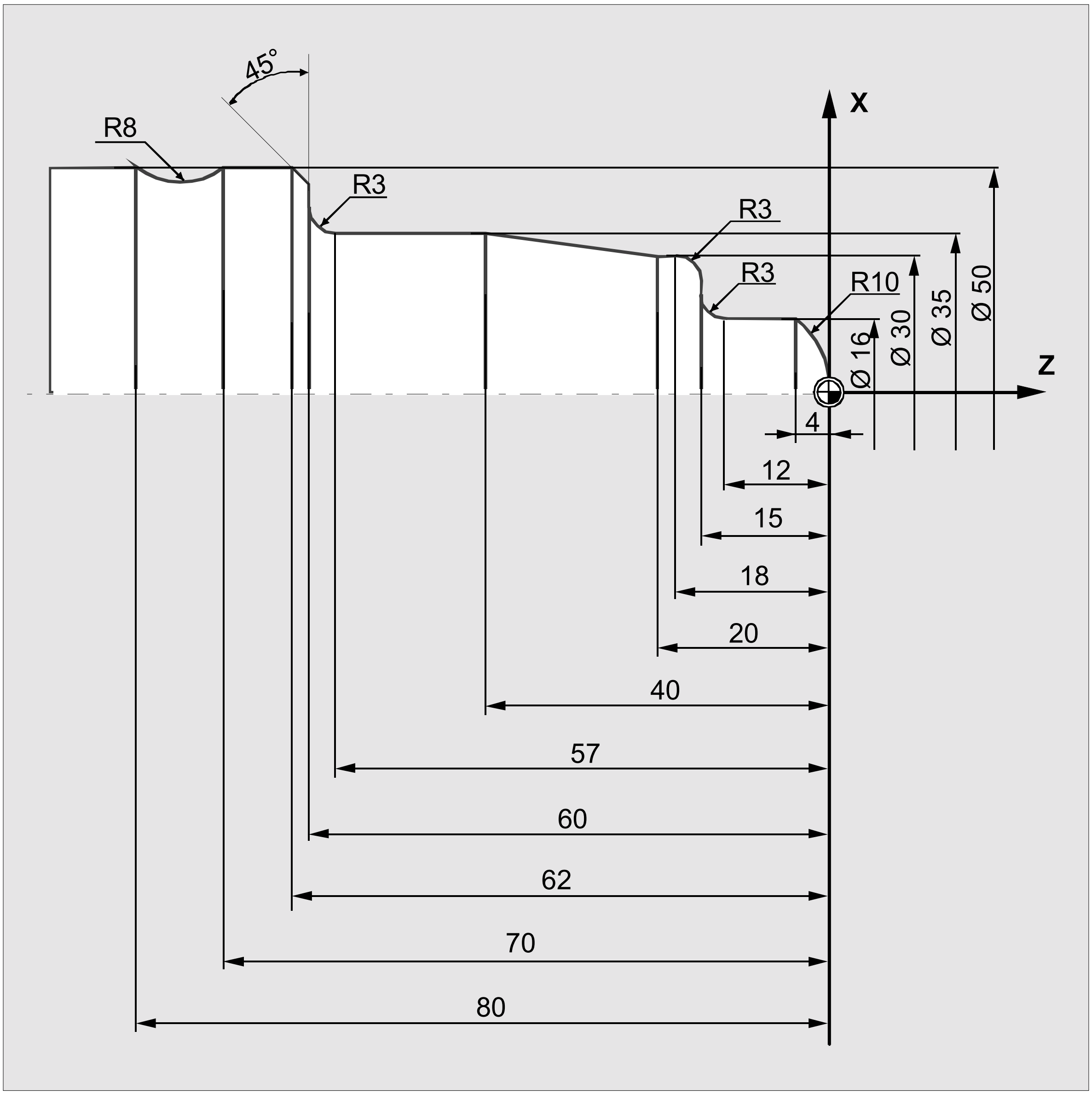
The sample program contains radius programming and tool radius compensation.

Programming example

%_N_1001_MPF	;Program name
N5 G0 G53 X280 Z380 D0	;Start point
N10 TRANS X0 Z250	;Zero offset
N15 LIMS=4000	;Speed limitation (G96)
N20 G96 S250 M3	;Select constant feed
N25 G90 T1 D1 M8	;Select tool and offset
N30 G0 G42 X-1.5 Z1	;Activate tool with tool radius compensation
N35 G1 X0 Z0 F0.25	
N40 G3 X16 Z-4 I0 K-10	;Rotate radius 10
N45 G1 Z-12	
N50 G2 X22 Z-15 CR=3	;Rotate radius 3
N55 G1 X24	
N60 G3 X30 Z-18 I0 K-3	;Rotate radius 3
N65 G1 Z-20	
N70 X35 Z-40	
N75 Z-57	
N80 G2 X41 Z-60 CR=3	;Rotate radius 3
N85 G1 X46	
N90 X52 Z-63	
N95 G0 G40 G97 X100 Z50 M9	;Deselect tool radius compensation and approach tool change location
N100 T2 D2	;Call up tool and select offset
N105 G96 S210 M3	;Select constant cutting speed
N110 G0 G42 X50 Z-60 M8	;Activate tool with tool radius compensation
N115 G1 Z-70 F0.12	;Rotate diameter 50
N120 G2 X50 Z-80 I6.245 K-5	;Rotate radius 8
N125 G0 G40 X100 Z50 M9	;Retract tool and deselect tool radius compensation
N130 G0 G53 X280 Z380 D0 M5	;Move to tool change location
N135 M30	;End of program

2.6 Programming example for turning application

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di



Machine manufacturer (MH2.5)

The MD settings must be defined correctly before the program can run on the machine.
References: /FB/ K2, "Axes, Coordinate Systems,.."

■

Positional Data

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

3.1 General information

In this section you will find a description of the commands that normally appear at the beginning of a NC program.

The way in which these functions are combined is not intended to be a patent remedy. For example, the choice of working plane may be made at another point in the NC program.

The real purpose of this and all the following sections is to illustrate the conventional structure of an NC program.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

3.2 Absolute/incremental dimensions, G90/G91

Programming

Absolute dimensioning

```
G90
X=AC(…) Y=AC(…) Z=AC(…)
```

Incremental dimensioning

```
G91 or
X=IC(…) Y=IC(…) Z=IC(…)
```

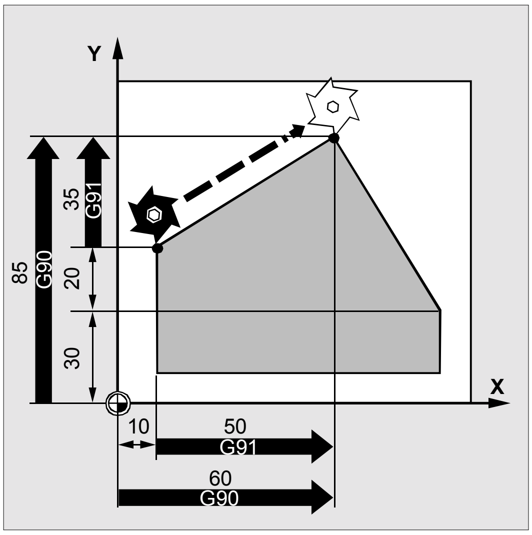
Explanation of the parameters

`X Y Z`	Axis identifiers of the axes to be traversed
`=AC`	Absolute dimensions (non-modal)
`=IC`	Incremental dimensions (non-modal)

Function

The G90/91 commands and the non-modal dimensions AC/IC are used to define the system for describing the approach to setpoints.

Sequence



Absolute dimensions, G90
The dimensions refer to the origin of the active coordinate system. You program the point to which the tool is to travel, e.g. in the workpiece coordinate system.

Incremental dimensions, G91
The dimensions refer to the last point approached. You program how far the tool is to travel.

Non-modal absolute or incremental dimensioning AC, IC
When G91 is active, AC can be used to allow entry of absolute dimensions for individual axes in a specific block. When G90 is active, IC can be used to allow entry of incremental dimensions for individual axes in a specific block.

840D NCU 571

840D NCU 572 NCU 573

810D

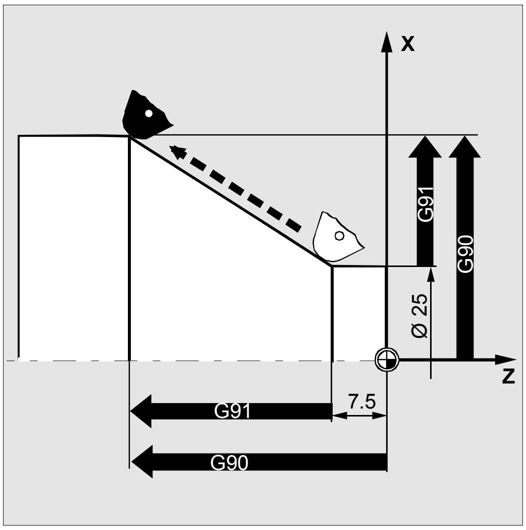
840Di

Additional notes

The commands G90 and G91 generally apply to all axes programmed in subsequent NC blocks.
Both commands are modal.

On conventional turning machines it is standard practice to interpret incremental NC blocks in the transverse axis as radius values, while diameter dimensions are valid for absolute coordinates. This conversion for G90/G91 is performed using the commands DIAMON, DIAMOF or DIAM90.

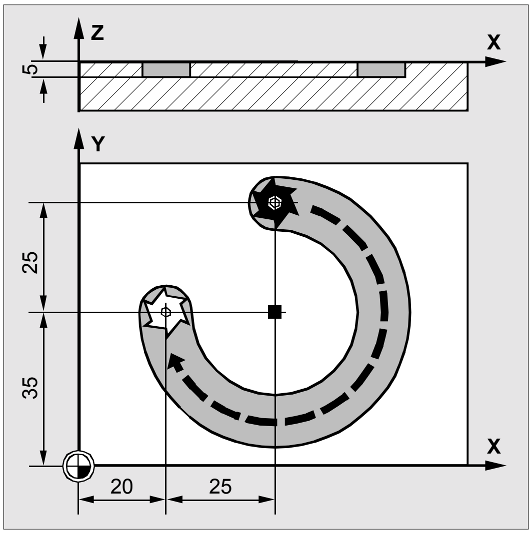
You will find further information in "Special turning functions" (Section 4.13) in this Programming Guide.



Programming example

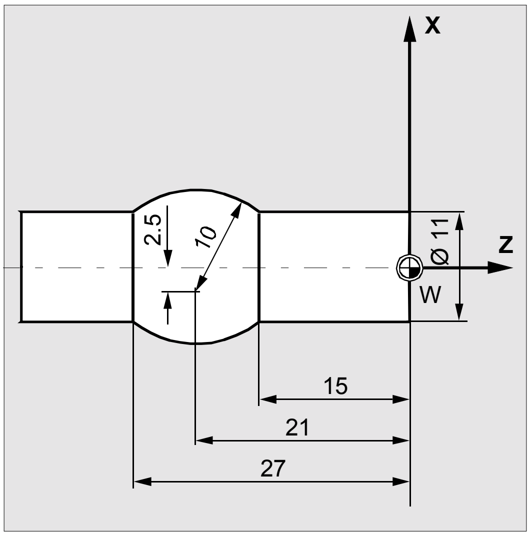
The traverse paths are entered in absolute coordinates with reference to the workpiece zero.

The center point coordinates I and J for circular interpolation are specified blockwise in absolute coordinates, since the arc center is programmed – independent of G90/G91 – in incremental coordinates as standard.



`N10`	`G90 G0 X45 Y60 Z2 T1 S2000 M3`	Absolute dimensioning, rapid traverse to XYZ, tool, spindle on clockwise
`N20`	`G1 Z-5 F500`	Tool infeed
`N30`	`G2 X20 Y35 I=AC(45) J=AC(35)`	Circle center point in absolute dimensions
or		
`N30`	`G2 X20 Y35 I0 J-25`	Circle center point in incremental dimensions
`N40`	`G0 Z2`	Retract
`N50`	`M30`	End of block

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di



N5	T1 D1 S2000 M3	Tool, spindle on clockwise
N10	G0 G90 X11 Z1	Absolute dimensioning, rapid traverse to XYZ
N20	G1 Z-15 F0.2	Tool infeed
N30	G3 X11 Z-27 I=AC(-5) K=AC(-21)	Circle center point in absolute dimensions
or		
N30	G3 X11 Z-27 I-8 K-6	Circle center point in incremental dimensions
N40	G1 Z-40	Retract
N50	M30	End of block

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

3.2.1 G91 extension (SW 4.3 and higher)

Programming

Incremental dimension data input `G91` or
`X=IC(…) Y=IC(…) Z=IC(…)`

- **without** traversing through the active **tool offset**

`SD 42442 TOOL_OFFSET_INCR_PROG = 0`

- **without** traversing through the active **zero offset**

`SD 42440 FRAME_OFFSET_INCR_PROG = 0`

Explanation of the parameters

`SD 42440 FRAME_OFFSET_INCR_PROG = 0`	The active zero offset is not traversed.
`SD 42442 TOOL_OFFSET_INCR_PROG = 0`	The active tool offset is not traversed.

Function

For applications such as scratching, it is necessary only to traverse the path programmed in the incremental coordinates. The active zero offset or tool offset is not traversed.
This can be set separately via SDs FRAME_OFFSET_INCR_PROG (zero point) and TOOL_OFFSET_INCR_PROG (tool offset).

Programming example

- G54 contains an offset of 25 in X
- SD 42440 FRAME_OFFSET_INCR_PROG = 0
  (no retraction of the active zero offset)

N10	G90 G0 G54 X100	
N20	G1 G91 X10	Traverse X by 10mm, the offset is not traversed
N30	G90 X50	Traverse to position X75, the offset is traversed

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

3.3 Absolute dimensions for rotary axes, DC, ACP, ACN

Programming

`A=DC(…) B=DC(…) C=DC(…)`
or
`A=ACP(…) B=ACP(…) C=ACP(…)`
or
`A=ACN(…) B=ACN(…) C=ACN(…)`

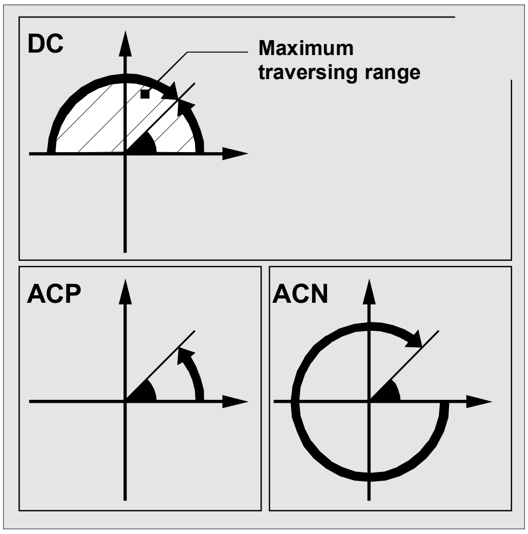
Explanation of the parameters

`A B C`	Axis identifier for rotary axis to be traversed
`DC`	Absolute dimensions, approach position directly
`ACP`	Absolute dimensions, approach position in positive direction
`ACN`	Absolute dimensions, approach position in negative direction

Function

With the above parameters you can define the desired approach strategy for positioning rotary axes.

Sequence



Absolute dimensioning with
The rotary axis travels to the position programmed in absolute coordinates along the shortest direct path. The rotary axis traverses across an area of up to 180°.

Absolute dimensioning with ACP
The rotary axis travels to the positions programmed in absolute coordinates in the positive direction of axis rotation (counterclockwise).

Absolute dimensioning with ACN
The rotary axis travels to the positions programmed in absolute coordinates in the negative direction of axis rotation (clockwise).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

The traversing range between 0° and 360° must be set in the machine data (modulo method) for positioning with directional data (ACP, ACN). G91 or IC must be programmed to traverse modulo rotary axes by more than 360° in a block. You will find more information on the previous pages.

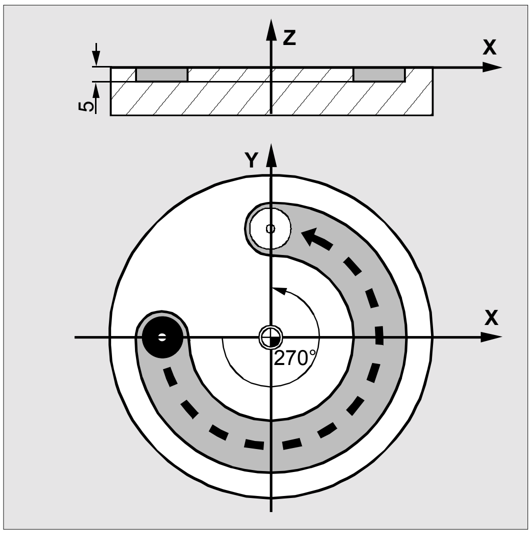
The positive direction of rotation (clockwise or counterclockwise) is set in the machine data.

Additional notes

All of the commands are modal.
You can also use DC, ACP and ACN for spindle positioning from zero speed.
Example:
`SPOS=DC(45)`

Programming example

Machining on a rotary table: The tool is stationary, the table rotates through 270° **in clockwise direction** to produce a circular groove.



`N10 SPOS=0`	Spindle in position control
`N20 G90 G0 X-20 Y0 Z2 T1`	Absolute, infeed in rapid traverse
`N30 G1 Z-5 F500`	Reduce feed
`N40 C=ACP(270)`	The table rotates through 270° in clockwise direction, the tool mills a circular groove
`N50 G0 Z2 M30`	Lift, end of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

3.4 Metric/imperial dimensions, G70/G71/G700/G710

Programming

Call

`G70` or `G71`

`G700` or `G710` SW5 and higher

Explanation of the commands

`G70`	Imperial measure (length [inches])
`G71`	Metric measure (length [mm])
`G700`	Imperial measure (length [inch]; feedrate [inch/min])
`G710`	Metric measure (length [mm]; feed [mm/min])

Function

Depending on the dimensions in the production drawing, you can program workpiece geometries alternately in metric measures and inches.

In SW 5 and higher, the functionality of G70/G71 has been extended with G700/G710. In addition to the geometrical parameters, the technological parameters, such as feed F, are interpreted during parts program execution in the system of units set in G700/G710.

Sequence

G70 or G71

You can instruct the control to convert the following geometrical dimensions (with necessary deviations) into the system of units not set and then enter them directly (see examples):

- Positional data X, Y, Z, …
- Intermediate point coordinates I1, J1, K1
  Interpolation parameters I, J, K and circle radius CR in circle programming
- Thread lead
- Programmable zero offset (TRANS)
- Polar radius RP

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

All other parameters such as feedrates, tool offsets or settable zero offsets are interpreted (when using G70/G71) in the default system of units (MD 10240: SCALING_SYSTEM_IS_METRIC).

The representation of system variables and machine data is also independent of the G70/G71 context.

G700 or G710

In SW 5 and higher, the controller interprets all feedrates used with G700/G710 in the programmed system of units, unlike G70/G71.

The G700/G710 codes are contained in the same group as G70/G71.

For information about the action of G70/G71 and G700/G710 on NC addresses, please refer to Section 12.2. "List of addresses".

The programmed feedrate value is modal and thus does not change automatically on subsequent G70/G71/G700/G710 selections.

If the feedrate in the G70/G71/G700/G710 context is to be activated, a new F value must be programmed explicitly.

All length-related NC data, machine data and setting data for G700/G710 are always read and written in the programmed context of G700/G710.

References: /FB, G2/, Sect. 2.2 "Metric/Inch System of Units"

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

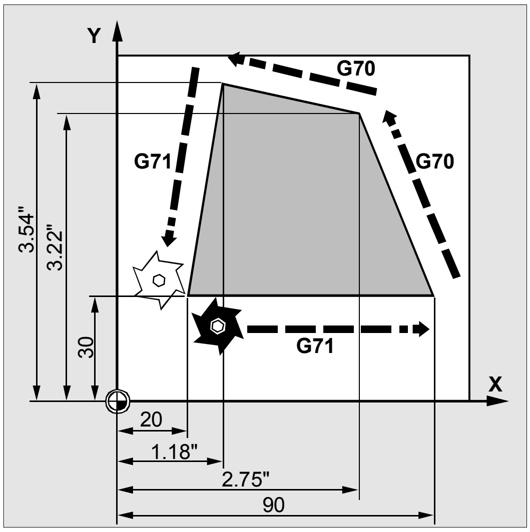
Synchronized actions

If positioning tasks are performed in synchronized actions and no G70/G71/G700/G710 command is programmed in the synchronized action itself, the G70/G71/G700/G710 context active at the time of execution determines which system of units is used.

References: /PGA/ Chapter 10, Motion-synchronous actions
/FBSY/ Synchronized Actions

Programming example

Change between metric and imperial input with basic setting metric (G70/G71).



N10	G0 G90 X20 Y30 Z2 S2000 M3 T1	Basic setting metric
N20	G1 Z-5 F500	At feed in Z [mm/min]
N30	X90	
N40	G70 X2.75 Y3.22	Enter destination positions in inches, G70 is active until deselected by G71 or end of program
N50	X1.18 Y3.54	
N60	G71 X 20 Y30	Enter positions in mm
N70	G0 Z2 M30	Retract in rapid traverse, end of program

840D NCU 571	840D NCU 572 NCU 573	810D	840Di

3.5 Zero offset (frame), G54 to G599

Programming

Call

`G54` or `G55` or `G56` or `G57` or `G505 … G599`

Deactivate

`G53` or `G500` or `SUPA` or `G153`

Explanation of the commands

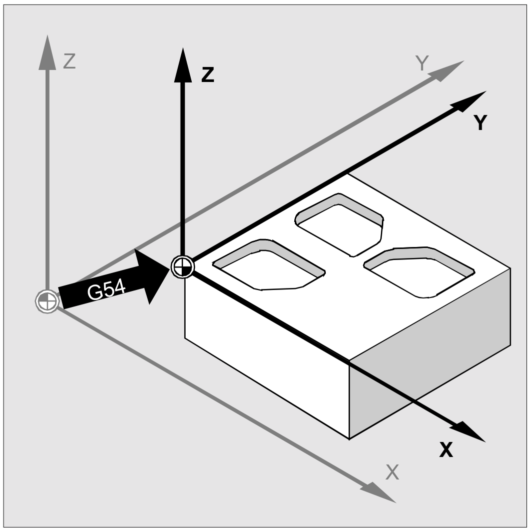
`G53`	Non-modal deactivation of current settable zero offset and programmable zero offset
`G54 to G57`	Call the second to fifth settable zero offset/frame
`G153`	Non-modal suppression of settable, programmable and total basic frame
`G500`	• G500=zero frame, default setting, (contains no offset, rotation, mirroring or scaling) • Deactivation of settable zero offsets / frames (G54 to G599) until the next call. • Activation of the total basic frame ($P_ACTBFRAME). • G500 is not 0 • Activation of first settable zero offset/frames ($P_UIFR[0]) and • Activation of total basic frame ($P_ACTBFRAME), or a modified basic frame is activated.
`SUPA`	Non-modal deactivation, including programmed offsets, handwheel offsets (DRF), external zero offset and PRESET offset.
`G505 ... G599`	Call the 6th to the 99th settable zero offset

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Function

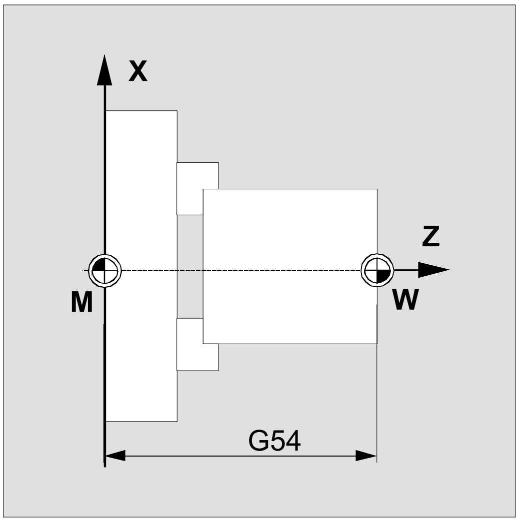
The settable zero offset relates the workpiece zero on all axes to the origin of the basic coordinate system.
It is therefore possible to call up cross-program zero points for different fixtures with a G command.

Milling:



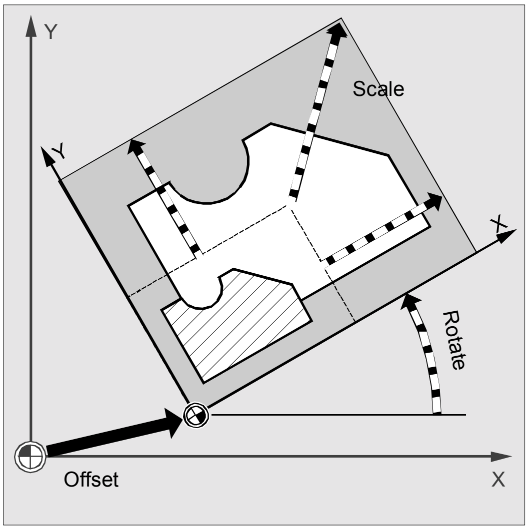
For turning, e.g. the offset value for tightening the chuck is entered in G54.

Turning:



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence



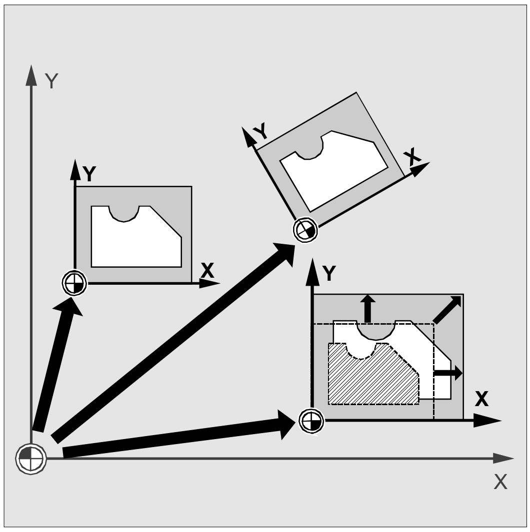
Setting the offset values

On the operator panel or universal interface, enter the following values in the internal control zero offset table:

- Coordinates for the offset
- Angle for rotated clamping and
- Scale factors if necessary

Please see the Operator's Guide for the procedure.

Activating the zero offset



In the NC program, the zero offset is moved from the machine coordinate system to the workpiece coordinate system by executing commands G54 to G57.

In the next NC block with a programmed movement, all of the positional parameters and thus the tool movements refer to the workpiece zero which is now valid.

The four available zero offsets can be used, e.g. for multiple machining operations, to describe four workpiece clamping positions simultaneously and execute them in the program.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Further settable zero offsets, G505 to G599
Command numbers G505 to G599 are available for this purpose. This enables you to create up to 100 settable zero offsets in total, in addition to the four default zero offsets G54 to G57, by using the machine data. They are stored in the zero point memory.
Please refer to Chapter 4 for further information.

Deactivating a zero offset
Command **G500** activates the first settable zero offset including basic offset, i.e. when zero frame is selected as the default, the current settable zero offset is deactivated.
G53 suppresses the programmable and settable offset non-modally.
G153 has the same effect as G53 and also suppresses the total basic frame.
SUPA has the same effect as G153 and also suppresses the DRF offset, overlaid motions and external ZOs.
You will find more information on programmable zero offsets in Chapter 6 (frames).

Additional notes

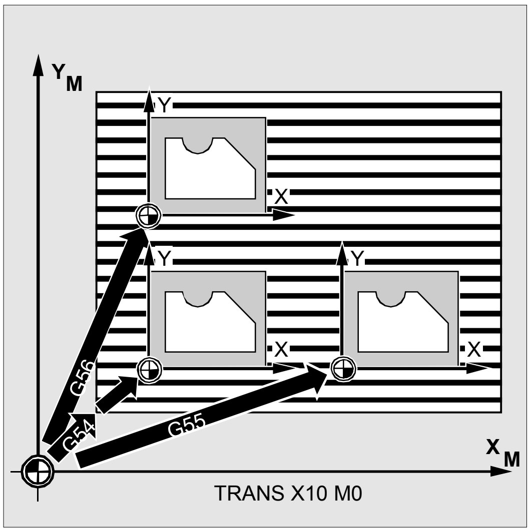
The basic setting at program start, e.g. G54 or G500, can be set with machine data.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

In this example, three workpieces, arranged on a pallet according to the zero offset values G54 to G56, are machined successively.

The machining sequence is programmed in subprogram L47.



N10	G0 G90 X10 Y10 F500 T1	Approach
N20	G54 S1000 M3	Call the first zero offset, spindle clockwise
N30	L47	Run program, in this case as a subprogram
N40	G55 G0 Z200	Call the second zero offset Z via obstacle
N50	L47	Run program as subprogram
N60	G56	Call third zero offset
N70	L47	Run program as subprogram
N80	G53 X200 Y300 M30	Suppress zero offset, end of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

3.6 Selecting the working plane, G17 to G19

Programming

Call

`G17` or `G18` or `G19`

Explanation of the commands

`G17`	Working plane X/Y	Infeed direction Z
`G18`	Working plane Z/X	Infeed direction Y
`G19`	Working plane Y/Z	Infeed direction X

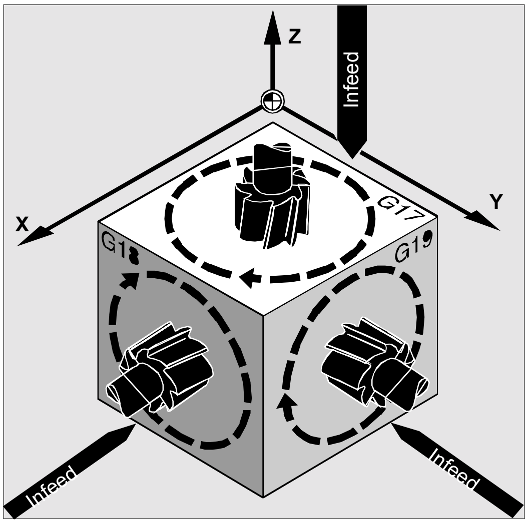
The axis assignment for G17, G18, G19 specified above is based on the supposition that X is assigned to the 1st geometry axis, Y to the second and Z to the third in the machine data.

Function

The specification of the working plane, in which the contour is to be machined also defines the following functions:

- The plane for tool radius compensation
- The infeed direction for tool length compensation depending on the tool type
- The plane for circular interpolation.

Milling:



3.6 Selecting the working plane, G17 to G19

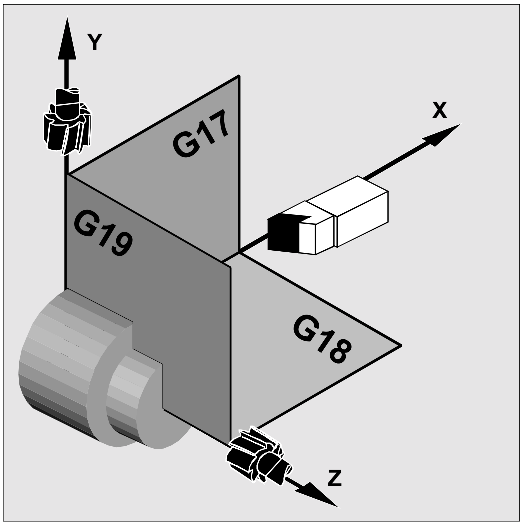
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

It is advisable to define the working plane at the beginning of the program.

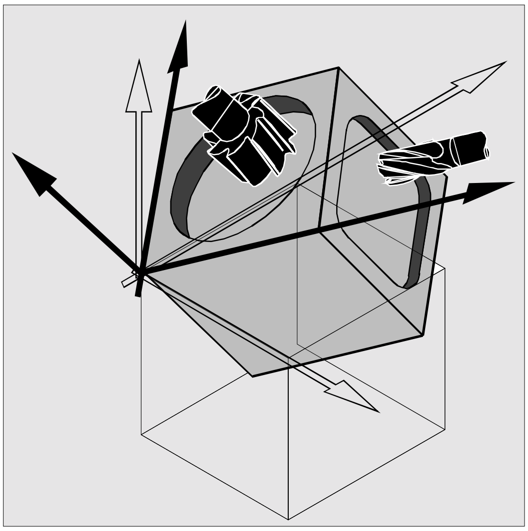
The working plane must be specified when the tool path compensation G41/G42 (see Section "Tool offsets") is called so that the control can correct the tool length and radius. In the initial setting, G17 (X/Y plane) is defined for milling and G18 (Z/X plane) is defined for turning.

Turning:



Machining on inclined planes
Rotate the coordinate system with ROT (see Section "Coordinate system offset") to position the coordinate axes on the inclined surface. The working planes rotate accordingly.

Tool length compensation in inclined planes
The tool length compensation generally always refers to the fixed, non-rotated working plane.



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Note

The tool length components can be calculated according to the rotated working planes with the functions for "Tool length compensation for orientable tools". For more information on this calculation type, please refer to Section "Tool Offsets".
The offset plane is selected with CUT2D, CUT2DF.
For more information see Section "Tool Offsets".

Additional notes

The control provides convenient coordinate transformation functions for the spatial definition of the working plane.
For further information, please refer to Section "Coordinate system offset".

Programming example

The "conventional" approach:
Define the working plane, call up the tool type and tool offset values, activate the path compensation, program the traversing movements.
Example for milling tool:

N10	G17 T5 D8	G17 Call the working plane, in this example X/Y T, D tool call. The length compensation is performed in the Z direction.
N20	G1 G41 X10 Y30 Z-5 F500	The radius compensation is performed in the X/Y plane.
N30	G2 X22.5 Y40 I50 J40	Circular interpolation / tool radius compensation in the X/Y plane.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

3.7 Programmable working area limitation, G25/G26

Programming

```
G25                 X…Y…Z…          (Programmed in a separate NC block)
G26                 X…Y…Z…          (Programmed in a separate NC block)
WALIMON, WALIMOF
```

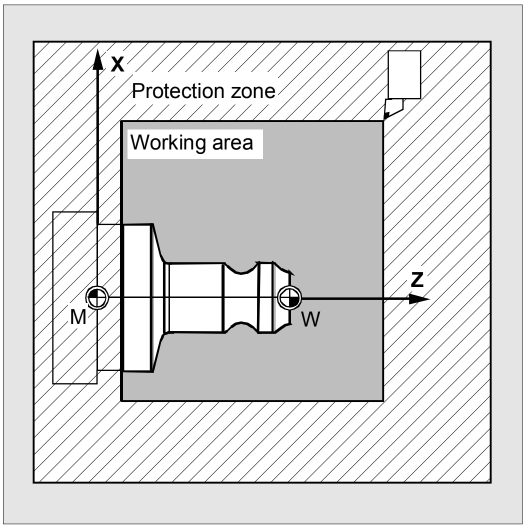
Explanation of the commands

`G25 X Y Z`	Lower working area limitation, value assignment in the channel axes*
`G26 X Y Z`	Upper working area limitation, value assignment in the channel axes*
`WALIMON`	Working area limitation activate
`WALIMOF`	Deactivate working area limitation

* Value assignments in the basic coordinate system

Function

G25/G26 limits the working area in which the tool can traverse in all channel axes.



These commands allow you to set up protection zones in the working area which are out of bounds for tool movements.

In addition to programming values using G25/G26, you can also parameterize them in setting data.

The axial setting data define the axes for which the working area limitation is valid.

The working area limitation for all validated axes must be programmed with the WALMON command. The WALIMOF command deactivates the working area limitation.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

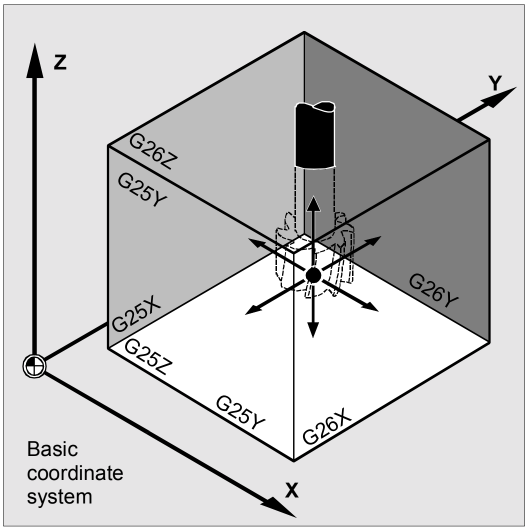
Reference points on the tool

When tool length compensation is active, the reference point is the tip of the tool, otherwise it is the toolholder reference point. If the tool is positioned outside the specified area or leaves this area, the program stops executing.

Programmable working area limitation, G25/G26

An upper (G26) and lower (G25) working area limit is defined for each axis. These values apply immediately and are not lost on Reset and when the control is switched on again.

The tool (milling tool) radius can be changed in the channel-specific machine data $MC_WORKAREA_WITH_TOOL_RADIUS (see "Advanced").



The coordinates for the individual axes apply in the basic coordinate system!

Activate/deactivate working area limitation

The command WALIMON activates working area limitation for all axes with the values programmed in G25/G26.

WALIMON is the default setting. It therefore only has to be programmed if working area limitation has been disabled.

The command WALIMOF is used to deactivate working area limitation for all of the axes.

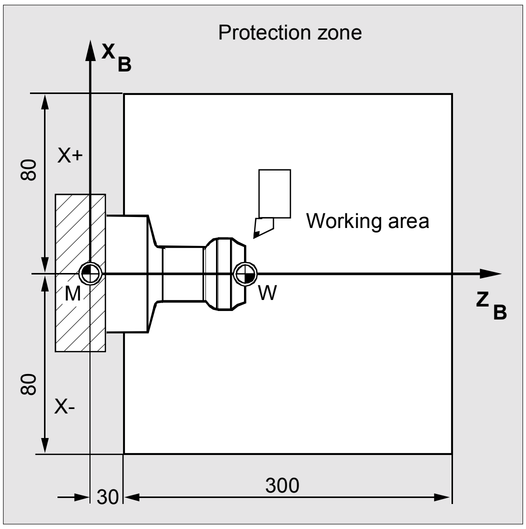
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Additional notes

G25/G26 can also be used to program limits for spindle speeds at the address S.
For further information, please refer to Section "Feed control and spindle motion".

Programming example

A protection zone is defined in the working area of a turning machine. This protects the surrounding equipment such as turrets, measuring stations, etc. against damage.
Default setting: WALIMON



N10	G0 G90 F0.5 T1	
N20	G25 X-80 Z30	Define the lower limit for the individual coordinate axes
N30	G26 X80 Z330	Define the upper limit
N40	L22	Cutting program
N50	G0 G90 Z102 T2	To tool change location
N60	X0	
N70	WALIMOF	Deactivate working area limitation
N80	G1 Z-2 F0.5	Boring
N90	G0 Z200	Back
N100	WALIMON	Activate working area limitation
N110	X70 M30	End of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

3.8 Reference point approach, G74

Programming

`G74` `X1=0 Y1=0 Z1=0 A1=0` … (programmed in a separate NC block)

Explanation of the commands

`G74`	Reference point approach
`X1=0 Y1=0…`	The specified machine axis address X1, Y1... approaches the reference point.

Function

When the machine has been powered up (where incremental position measurement systems are used), all of the axis slides must approach their reference point.
Only then can traversing movements be programmed.

The reference point can be approached in the NC program with G74.

Sequence

The speed at which the axis slide travels is defined in the machine data and cannot be programmed.

The control detects the traversing direction automatically.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

The machine axis addresses are programmed (X1, Y1, Z1, etc.)!

A transformation should not be programmed for an axis which is to approach the reference point with G74.

Deactivate transformation with the command TRAFOOF.

Programming example

When the measurement system is changed, the reference point is approached and the workpiece zero is initialized.

N10	SPOS=0	Position control
N20	G74 X1=0 Y1=0 Z1=0 C1=0	Reference point approach for linear axes and rotary axes
N30	G54	Zero offset
N40	L47	Cutting program
N50	M30	End of program

■

Programming Motion Commands

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

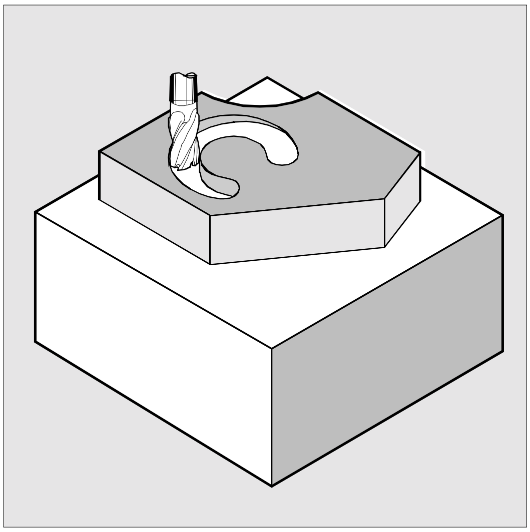
4.1 General information

Motion commands
In this section you will find a description of all the travel commands you can use to machine workpiece contours.

You can program straight lines and arcs of a circle. A helix can be produced by combining these two elements.

Executed in succession, these contour elements produce the workpiece contour.

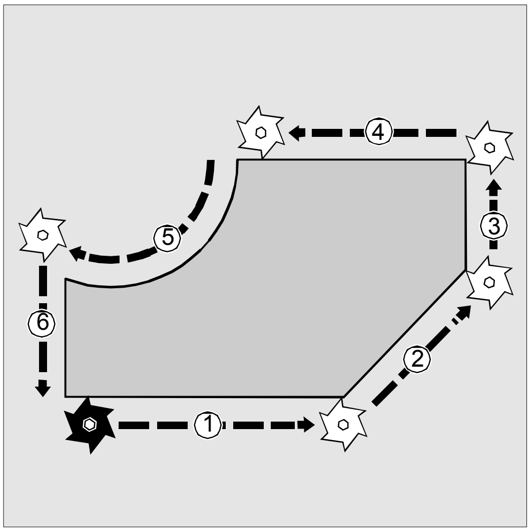
Before a machining process is started, you need to position the tool in such a way as to avoid any damage to the tool or workpiece.



Start point – destination point
The traversing movement always runs from the last approached position to the programmed destination position. This destination position is also the start position for the next travel command.

Number of axis values
Depending on the control configuration, you can program up to 8 axes per set of movements. These may include path axes, synchronized axes, positioning axes and spindle oscillation mode.

Milling:



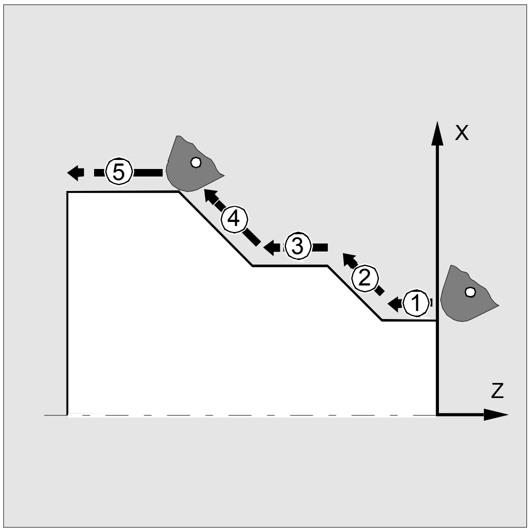
840D
NCU 571

840D
NCU 572
NCU 573

810D

840Di

Turning:



An axis address can only be programmed once in each block.

These commands can be programmed in Cartesian or polar coordinates.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.2 Traversing commands with polar coordinates, G110, G111, G112, AP, RP

Programming

Define pole:

```
G110, G111, G112 X… Y… Z…
G110, G111, G112 AP=… RP=…
```

Traversing commands with polar coordinates:

```
G0 AP=… RP=…
G1 AP=… RP=…
G2 AP=… RP=…
G3 AP=… RP=…
```

The new end point is defined in relation to a pole.

Explanation of the commands and parameters

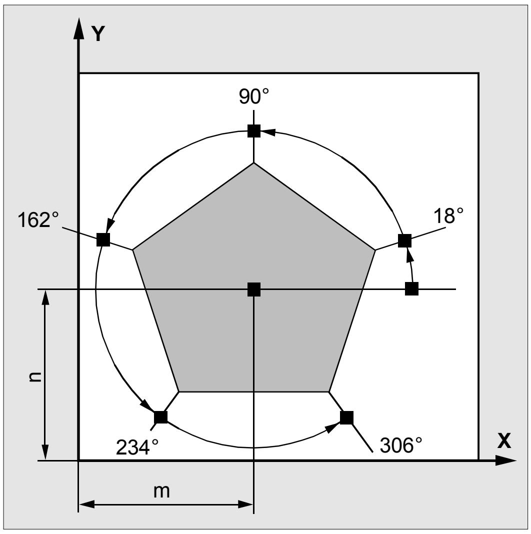
`G110`	Pole parameter, with reference to the last approached position
`G111`	Pole parameter, absolute in the workpiece coordinate system
`G112`	Pole parameter, with reference to the last valid pole
`AP=`	Polar angle, value range ±0…360°, angle refers to horizontal axis of the working plane
`RP=`	Polar radius in mm or inches

All the commands relating to pole input must be programmed in a separate NC block

Function

A workpiece is frequently dimensioned with a central point as origin; the dimensions are given in terms of angles and radii, e.g. in drilling patterns.

Polar coordinates can be used to program these dimensions directly in accordance with the drawing.



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

Traversing commands

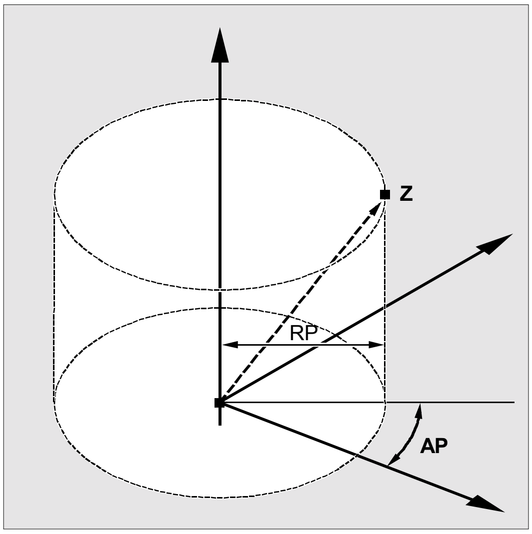
The positions specified by polar coordinates can be traversed with G0, G1, G2 and G3.

Working plane

The polar coordinates are valid in the working plane selected with G17 to G19.

Cylindrical coordinates

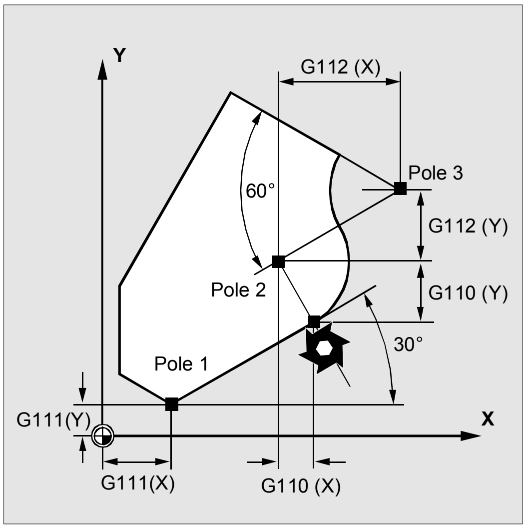
The 3rd geometry axis, which lies perpendicular to the working plane, can also be specified in Cartesian coordinates.
This enables spatial parameters to be programmed in cylindrical coordinates.
Example: `G17 G0 AP… RP… Z…`



Defining the pole G110, G111, G112

The pole can be specified in Cartesian or polar coordinates.

G commands G110 to G112 are used to provide a unique definition of the reference point for dimensions. Absolute or incremental dimensioning (AC/IC) therefore has no effect on the systematics specified in the G command.



If no pole is specified, the origin of the active workpiece coordinate system applies.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

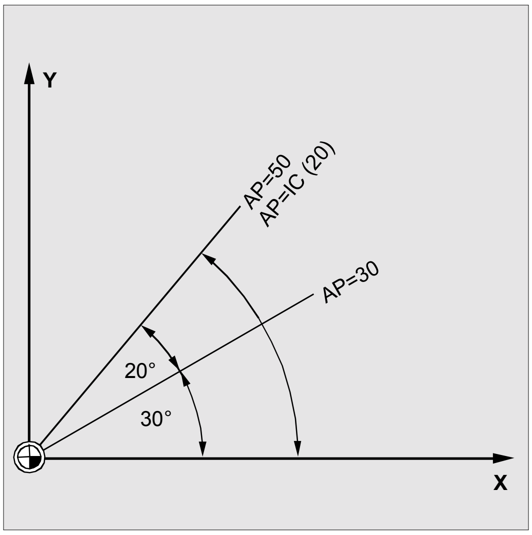
Polar angle AP

Value range 0…±360°.

With absolute input, the angle refers to the horizontal axis of the working plane, e.g. X axis with G17. The positive direction of rotation runs counterclockwise.

When incremental coordinates are entered (AP=IC…), the last angle programmed is taken as the reference.

The polar angle is stored until a new pole is defined or the working plane is changed.



Polar radius RP

The polar radius is specified in mm or inches in absolute positive values. RP is stored until a new value is input.

SW 4.1 and higher

If the modally active polar radius is RP = 0

The polar radius is calculated from the distance between the starting point vector in the polar plane and the active pole vector. The calculated polar radius is stored modally afterwards.

This applies irrespective of the selected pole definition G110, G111, G112. If both points are programmed identically, then this radius becomes 0 and alarm 14095 is generated.

If a pole angle AP is programmed with RP = 0

If the current block contains a polar angle AP rather than a polar radius RP and if there is a difference between the current position and pole in workpiece coordinates, then this difference is applied as the polar radius and stored modally.

If the difference = 0, the pole coordinates are specified again and the modal polar radius remains zero.

The following general rule applies:

You must not program Cartesian coordinates, such as interpolation parameters or axis addresses, for the selected working plane in NC blocks with polar end position coordinates.

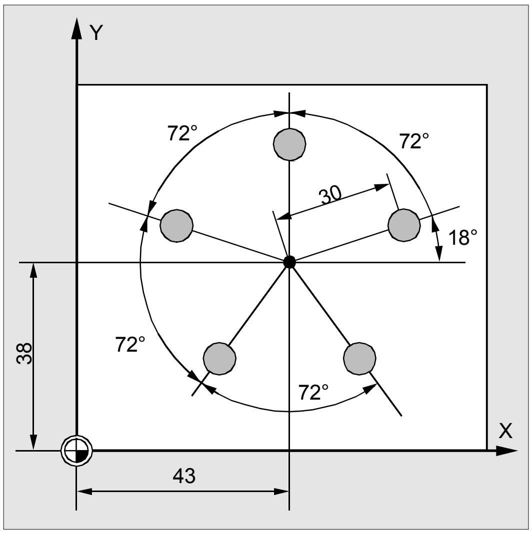
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Additional notes

In the NC program you can switch between polar and Cartesian coordinates, block by block.

Programming example

Making a hole pattern: The positions of the holes are specified in polar coordinates. Each hole is machined with the same production sequence: Predrill, drill to size, ream etc.
The machining sequence is stored in a subprogram.



N10	G17 G54	Working plane X/Y, workpiece zero
N20	G111 X43 Y38	Define pole
N30	G0 RP=30 AP=18 Z5	Approach starting point, position in cylindrical coordinates
N40	L10	Subprogram call
N50	G91 AP=72	Approach next position in rapid traverse, polar angle in incremental dimensions, polar radius from block N30 is still stored and does not need to be specified
N60	L10	Subprogram call
N70	AP=IC(72)	…
N80	L10	…
N90	AP=IC(72)	
N100	L10	…
N110	AP=IC(72)	
N120	L10	…
N130	G0 X300 Y200 Z100 M30	Retract tool, end of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.3 Rapid traverse movement, G0

Programming

```
G0    X… Y… Z …
G0    AP=… RP=…
RTLIOF, RTLION (SW 6.1 and higher)
```

Explanation of the parameters

`X Y Z`	End point in Cartesian coordinates
`AP=`	End point in polar coordinates, in this case the polar angle
`RP=`	End point in polar coordinates, in this case the polar radius
`RTLIOF with G0`	Nonlinear interpolation (each path axis interpolates as a single axis)
`RTLION with G0`	Linear interpolation (path axes are interpolated together)

Function

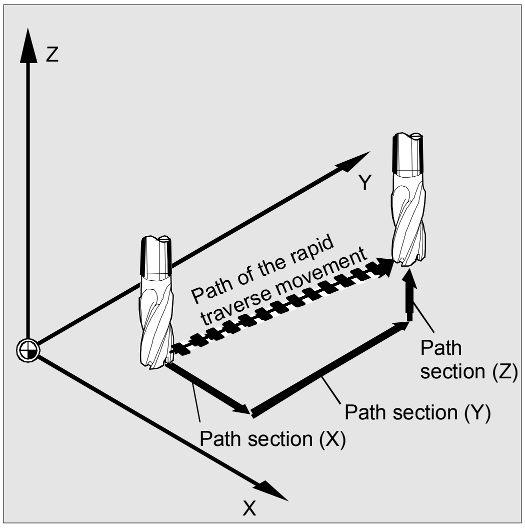
You can use the rapid traverse movements to position the tool rapidly, to travel round the workpiece or to approach tool change locations.

This function is not suitable for workpiece machining!

Sequence

The tool movement programmed with G0 is executed at the highest possible speed (rapid traverse). The rapid traverse speed is defined separately for each axis in machine data.

If the rapid traverse movement is executed simultaneously on several axes, the rapid traverse speed is determined by the axis which requires the greatest time for its section of the path.



Additional notes

G0 is modal.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Function

SW 6.1 and higher

Traversing path axes as positioning axes with G0

Path axes can travel in one of two different modes to execute movements in rapid traverse:

- **Linear interpolation:** (behavior in earlier SW version)
  The path axes are interpolated together.
- **Nonlinear interpolation**: (SW 6 and higher)
  Each path axis is interpolated as an individual (positioning) axis independently of the other axes involved in the rapid traverse movement.

Relevant parts program commands:

- RTLIOF activates nonlinear interpolation
- RTLION activates linear interpolation

Linear interpolation must always be selected in the following cases:

- With a G code combination including G0 which does not permit positioning movements (e.g. G40/41/42).
- With a combination of G0 and G64
- When the compressor is active
- When a transformation is active

With nonlinear interpolation, the setting for the relevant positioning axis BRISKA, SOFTA, DRIVEA applies with regard to axial jerk.

Since a different contour can be traversed in nonlinear interpolation mode, synchronized actions that refer to coordinates of the original path are not operative in some cases!

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

Traverse path axes as positioning axes with G0
Example:

```
G0 X0 Y10
G0 G40 X20 Y20
G0 G95 X100 Z100 m3 s100
```

Path POS[X]=0 POS[Y]=10 is traversed in path mode.
No revolutional feedrate is active if path POS[X]=100 POS[Z]=100 is traversed.

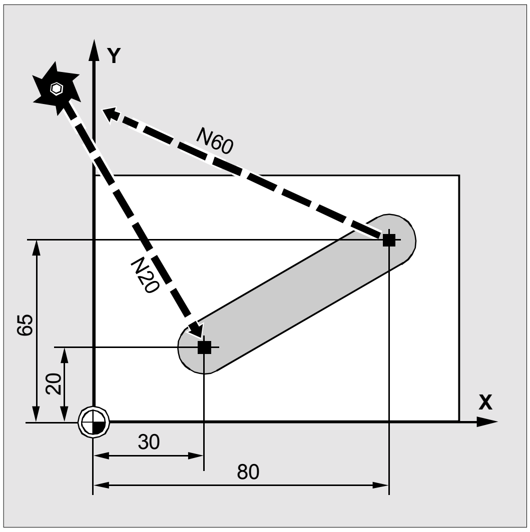
Additional notes

SW 6.2 and higher
Settable block change time with G0:
It is possible to set a new motion end criterion, i.e. FINEA, COARSEA or IPOENDA, in single axis interpolation mode for block changes within the braking ramp.
All axes can reach their end points independently of one another through a combination of "Block change settable in braking ramp of single axis interpolation" and "Traverse path axes as positioning axes with G0 rapid traverse".
In this way, two sequentially programmed X and Z axes are treated like positioning axes in conjunction with G0. The block change to axis Z can be initiated by axis X as a function of the braking ramp time setting (100-0%). Axis Z starts to move while axis X is still in motion. Both axes approach their end point independently of one another.
You will find more information in Chapter 7.

Programming example

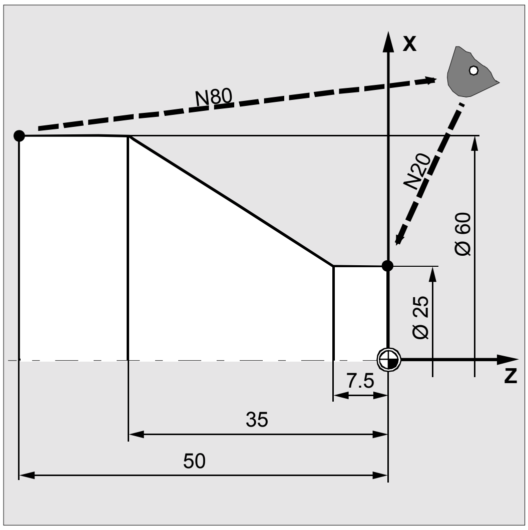
G0 is used for approaching starting positions or tool change locations, retracting the tool, etc.

Milling:



N10	G90 S400 M3	Absolute dimensioning, spindle clockwise
N20	G0 X30 Y20 Z2	Approach start position
N30	G1 Z-5 F1000	Tool infeed
N40	X80 Y65	Travel on straight line
N50	G0 Z2	
N60	G0 X-20 Y100 Z100 M30	Retract tool, end of program

Turning:



N10	G90 S400 M3	Absolute dimensioning, spindle clockwise
N20	G0 X25 Z5	Approach start position
N30	G1 G94 Z0 F1000	Tool infeed
N40	G95 Z-7.5 F0.2	
N50	X60 Z-35	Travel on straight line

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

N60	Z-50	
N70	G0 X62	
N80	G0 X80 Z20	Retract tool
N90	M30	End of program

G0 cannot be replaced with G.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.4 Linear interpolation, G1

Programming

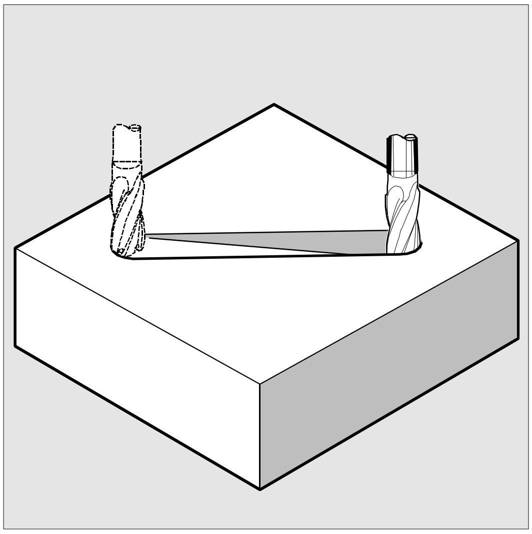
```
G1      X…  Y…  Z  …  F…
G1      AP=…  RP=…  F…
```

Explanation of the parameters

X Y Z	End point in Cartesian coordinates
AP=	End point in polar coordinates, in this case the polar angle
RP=	End point in polar coordinates, in this case the polar radius
F	Feedrate in mm/min

Function

With G1, the tool travels along straight lines that are parallel to the axis, inclined or in any orientation in space. The straight line interpolation enables machining of 3D surfaces, grooves, etc.



Sequence

The tool travels at feedrate F along a straight line from the current starting point to the programmed destination point. The workpiece is machined along this path. You can enter the destination point in Cartesian or polar coordinates.
Example:

```
G1 G94 X100 Y20 Z30 A40 F100
```

The end point on X, Y, Z is approached at a feedrate of 100 mm/min; the rotary axis A is traversed as a synchronized axis in order that all four movements are completed at the same time.

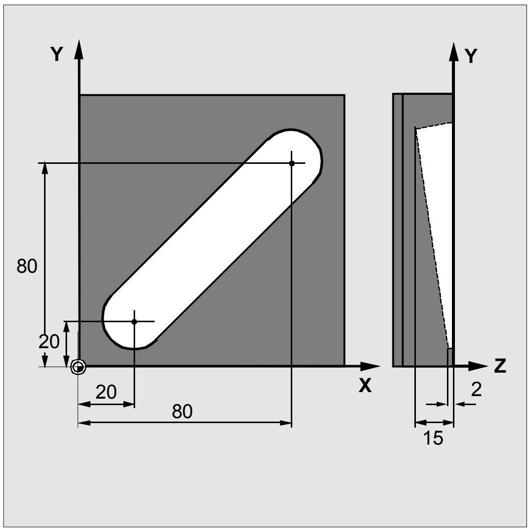
Additional notes

G1 is modal. The spindle speed S and the direction of spindle rotation M3/M4 must be specified for machining.
FGROUP can be used to define groups of axes to which the path feed F applies.
You will find more information in Chapter 5.

Programming example

Machining of a groove: The tool travels from the starting point to the end point in the X/Y direction. Infeed takes place simultaneously in the Z direction.

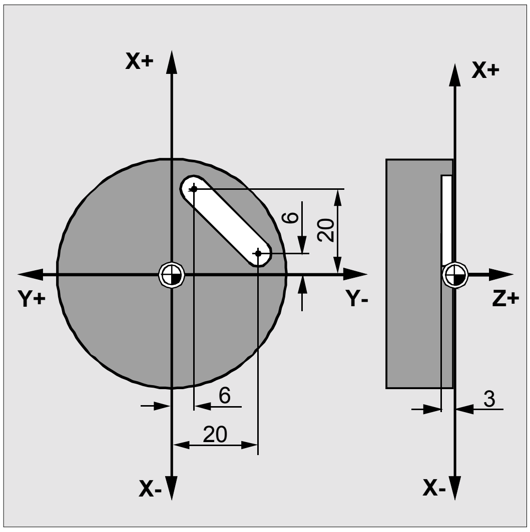
Milling:



N10	G17 S400 M3	Select working plane, spindle clockwise
N20	G0 X20 Y20 Z2	Approach start position
N30	G1 Z-2 F40	Tool infeed
N40	X80 Y80 Z-15	Travel along inclined straight line
N50	G0 Z100 M30	Retract to tool change location

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Turning:



`N10   G17 S400 M3`	Select working plane, spindle clockwise
`N20   G0 X40 Y-6 Z2`	Approach start position
`N30   G1 Z-3 F40`	Tool infeed
`N40   X12 Y-20`	Travel along inclined straight line
`N50   G0 Z100 M30`	Retract to tool change location

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.5 Circular interpolation, G2/G3, CIP

Programming

```
G2/G3 X… Y… Z… I… J… K…
G2/G3 AP=… RP=…
G2/G3 X… Y… Z… CR=…
G2/G3 AR=… I… J… K…
G2/G3 AR=… X… Y… Z…
CIP   X… Y… Z… I1=… J1=… K1=…
CT    X… Y… Z…
```

Explanation of the commands and parameters

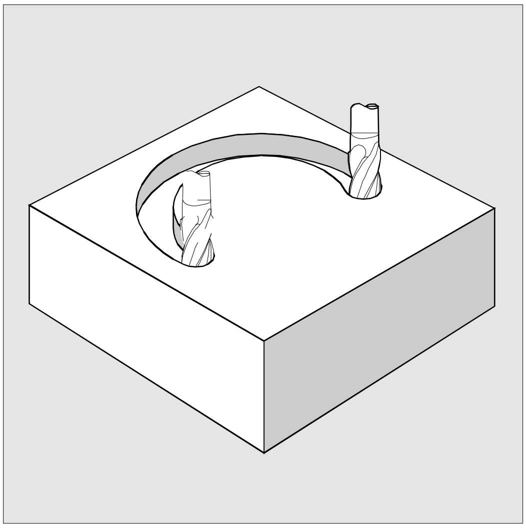
G2	Travel on a circular path in clockwise direction
G3	Travel on a circular path in counterclockwise direction
CIP	Circular interpolation through intermediate point
CT	Circle with tangential transition

X Y Z	End point in Cartesian coordinates
I J K	Circle center point in Cartesian coordinates (in X, Y, Z direction)
AP=	End point in polar coordinates, in this case the polar angle
RP=	End point in polar coordinates, in this case polar radius corresponding to circle radius
CR=	Circle radius
AR=	Arc angle
I1= J1= K1=	Intermediate points in Cartesian coordinates (in X, Y, Z direction)

There is no practical limitation on the maximum size of the programmable radius.

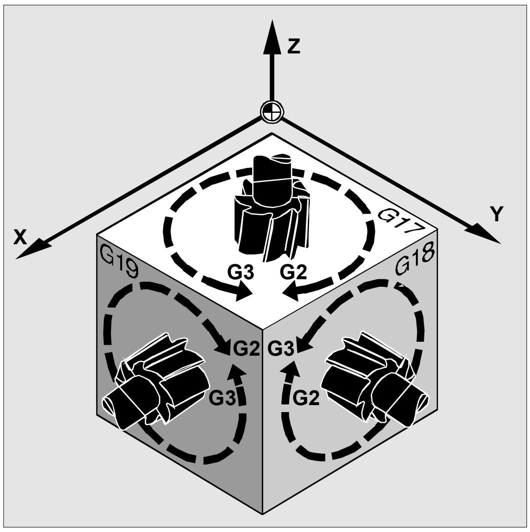
Function

Circular interpolation enables machining of full circles or arcs.



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

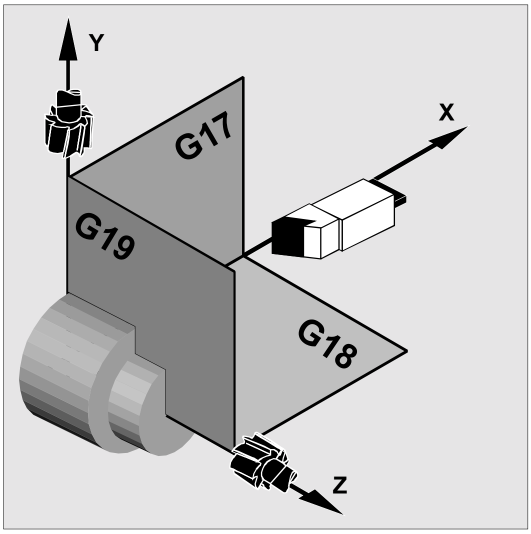
Sequence



Indication of working plane

The control needs the working plane parameter (G17 to G19) in order to calculate the direction of rotation for the circle – G2 is clockwise/G3 is counterclockwise. It is generally advisable to specify the working plane.
Exception:
You can also machine circles outside the selected working plane (not with arc angle and helix parameters). In this case, the axis addresses that you specify as an end point determine the circle plane.

Additional notes



G2/G3 are modal.

You can use FGROUP to specify which axes are to be traversed with a programmed feedrate.
You will find more information in Chapter 5.

The control provides a range of different ways to program circular movements. This allows you to implement almost any type of drawing dimension directly.

For detailed descriptions please refer to the following pages.

Programming a circle with center point and end point

The circular movement is described by:

- The end point in Cartesian coordinates X, Y, Z and
- the circle center point at addresses I, J, K.

The identifiers have the following meanings:

I: Coordinate of the circle center point in the X direction
J: Coordinate of the circle center point in the Y direction
K: Coordinate of the circle center point in the Z direction

If the circle is programmed with a center point but no end point, the result is a full circle.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Input in absolute and incremental dimensions

The defaults G90/G91 absolute or incremental coordinates are valid only for the circle end point.
The center point coordinates I, J, K are normally entered in incremental dimensions with reference to the circle starting point.

You program the absolute center point non-modally with reference to the workpiece zero with: I=AC(…), J=AC(…), K=AC(…)

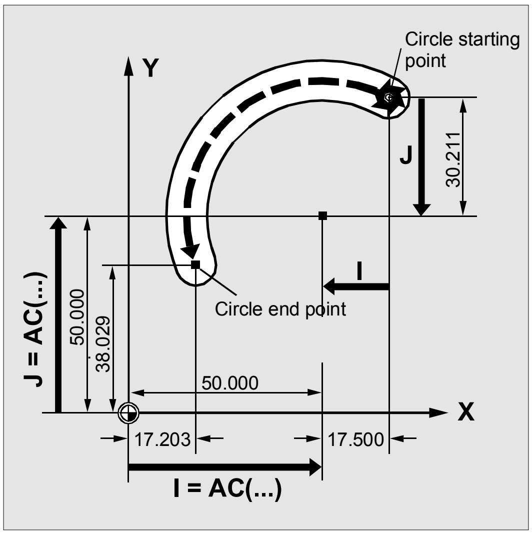
Example for incremental dimensions:

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I–17.5 J–30.211
    F500
```

Example for absolute dimensions:

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I=AC(50)
    J=AC(50)
```

Milling:



An interpolation parameter I, J, K with value 0 can be omitted but the second associated parameter must always be specified.

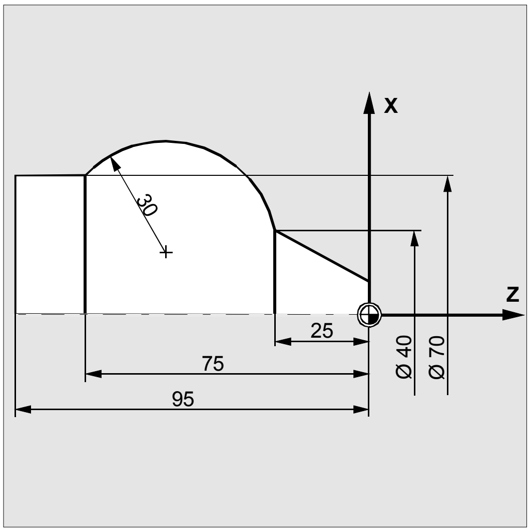
Example for incremental dimensions:

```
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 I-3.335 K-29.25
N135 G1 Z-95
```

Example for absolute dimensions:

```
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 I=AC(33.33)
     K=AC(-54.25)
N135 G1 Z-95
```

Turning:



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming a circle with radius and end point

The circular movement is described by:

- The circle radius CR= and
- the end point in Cartesian coordinates X, Y, Z.

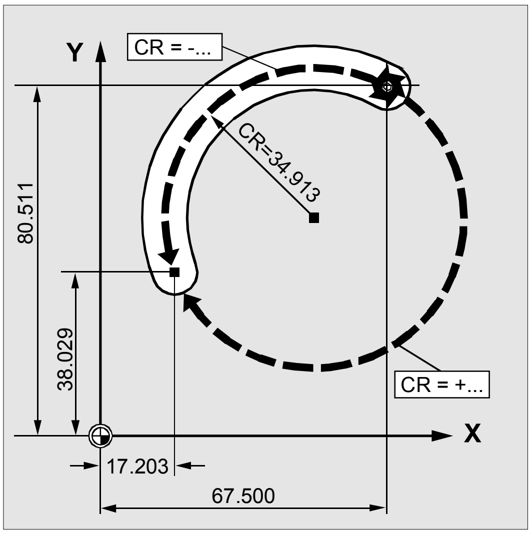
In addition to the circle radius, you must also specify the leading sign +/– to indicate whether the traversing angle is to be greater than or less than 180°. A positive leading sign can be omitted.

The identifiers have the following meanings:

`CR=+…:` Angle smaller or equal to 180°

`CR=–…:` Angle larger than 180°

Milling:



Example:

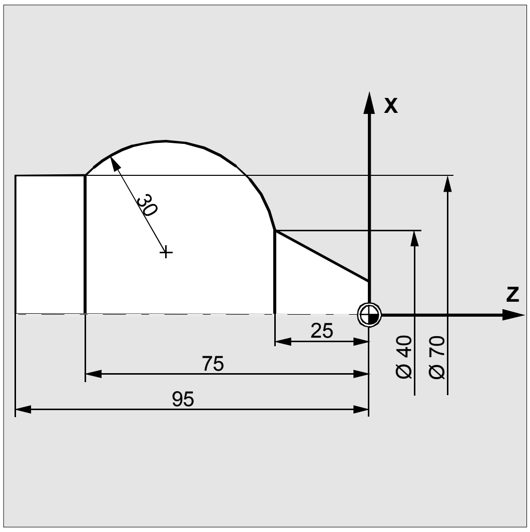
```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 CR=34.913 F500
```

You don't need to specify the center point with this procedure. Full circles (traversing angle 360°) cannot be programmed with CR=, but must be programmed using the circle end point and interpolation parameters.

Example:

```
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 CR=30
N135 G1 Z-95
```

Turning:



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming a circle with arc angle and center point or end point

The circular movement is described by:

- The arc angle AR = and
- the end point in Cartesian coordinates X, Y, Z or
- the circle center point at addresses I, J, K

The identifiers have the following meanings:
AR=: arc angle, value range 0° to 360°

Please refer to the preceding pages for the meanings of I, J, K.

Full circles (traversing angle 360°) cannot be programmed with AR=, but must be programmed using the circle end point and interpolation parameters.

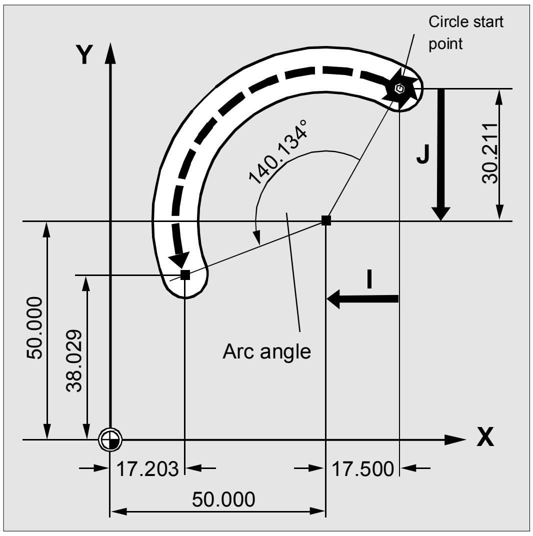
Example:

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 AR=140.134 F500
```

or

```
N20 G3 I–17.5 J–30.211 AR=140.134 F500
```

Milling:



Example:

```
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 AR=135.944
```

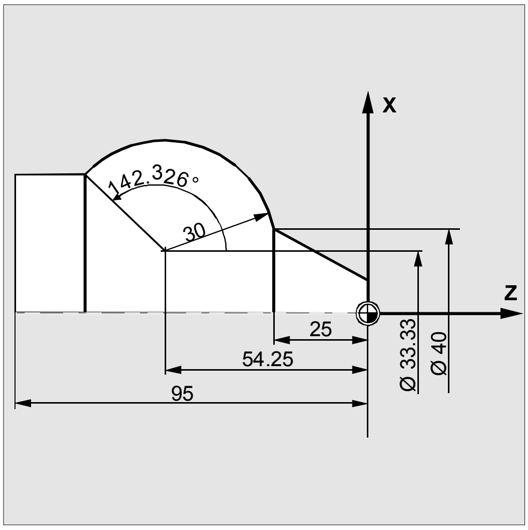
or

```
N130 G3 I-3.335 K-29.25 AR=135.944
```

or

```
N130 G3 I=AC(33.33) K=AC(-54.25)
     AR=135.944
N135 G1 Z-95
```

Turning:



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming a circle with polar coordinates

The circular movement is described by:

- The polar angle AP= and
- the polar radius RP=

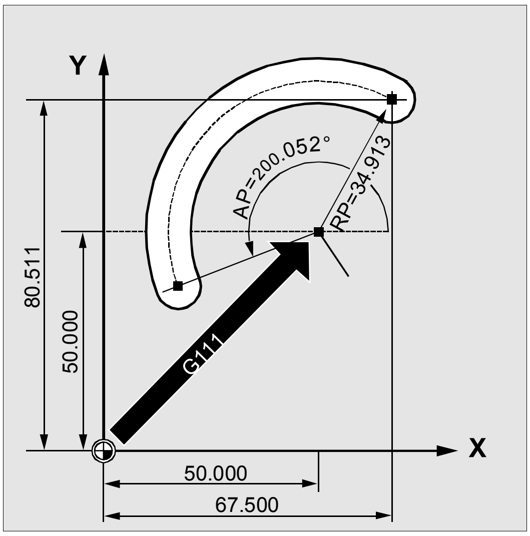
The following rule applies:

The pole lies at the circle center point.
The polar radius corresponds to the circle radius.

Example:

```
N10 G0 X67.5 Y80.211
N20 G111 X50 Y50
N30 G3 RP=34.913 AP=200.052 F500
```

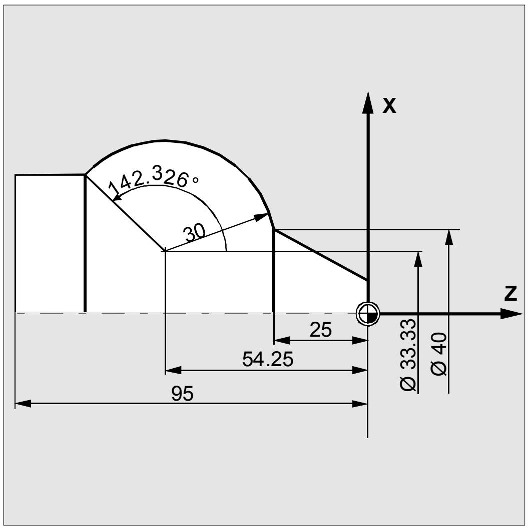
Milling:



Example:

```
N125 G1 X40 Z-25 F0.2
N130 G111 X33.33 Z-54.25
N135 G3 RP=30 AP=142.326
N140 G1 Z-95
```

Turning:

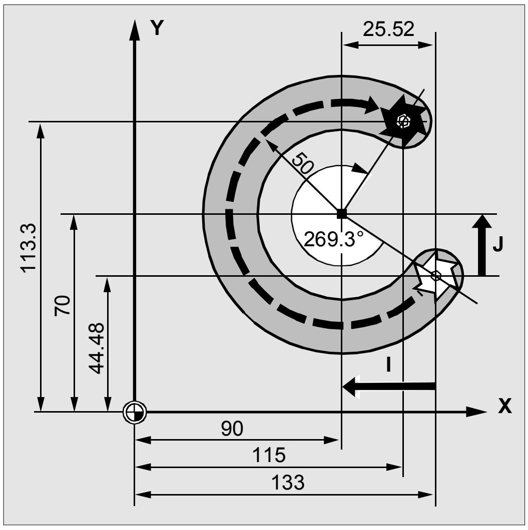


840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

The following program lines contain an example for each circular programming possibility. The necessary dimensions are shown in the opposite production drawing.

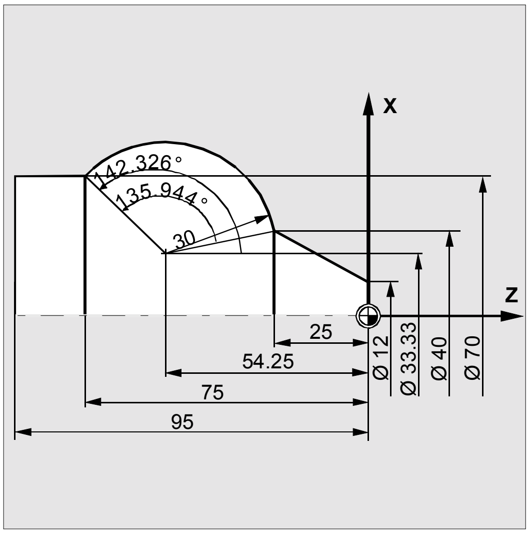
Milling:



N10	G0 G90 X133 Y44.48 S800 M3	Approach starting point
N20	G17 G1 Z-5 F1000	Tool infeed
N30	G2 X115 Y113.3 I-43 J25.52	Circle end point, center point in incremental dimensions
or		
N30	G2 X115 Y113.3 I=AC(90) J=AC(70)	Circle end point, center point in absolute dimensions
or		
N30	G2 X115 Y113.3 CR=-50	Circle end point, circle radius
or		
N30	G2 AR=269.31 I-43 J25.52	Arc angle, center point in incremental dimensions
or		
N30	G2 AR=269.31 X115 Y113.3	Arc angle, circle end point
N40	M30	End of program

840D NCU 571	840D NCU 572 NCU 573	810D	840Di

Turning:

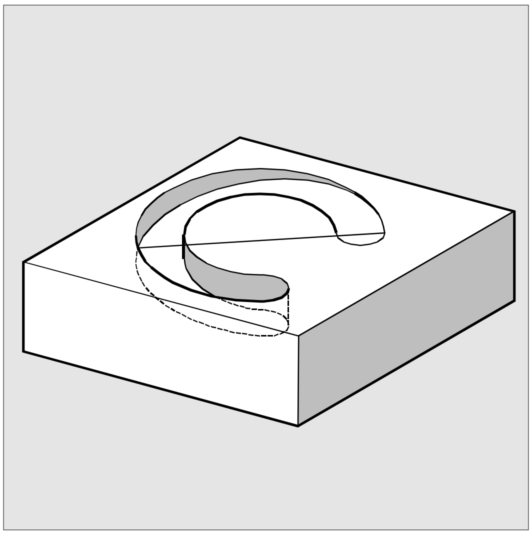


N..	...	
N120	G0 X12 Z0	
N125	G1 X40 Z-25 F0.2	
N130	G3 X70 Y-75 I-3.335 K-29.25	Circle end point, center point in incremental dimensions
or		
N130	G3 X70 Y-75 I=AC(33.33) K=AC(-54.25)	Circle end point, center point in absolute dimensions
or		
N130	G3 X70 Z-75 CR=30	Circle end point, circle radius
or		
N130	G3 X70 Z-75 AR=135.944	Arc angle, circle end point
or		
N130	G3 I-3.335 K-29.25 AR=135.944	Arc angle, center point in incremental dimensions
or		
N130	G3 I=AC(33.33) K=AC(-54.25) AR=135.944	Arc angle, center point in absolute dimensions
or		
N130	G111 X33.33 Z-54.25	Polar coordinates
N135	G3 RP=30 AP=142.326	Polar coordinates
N140	G1 Z-95	
N..	...	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming a circle with intermediate and end points

You can use CIP to program arcs. These arcs can also be inclined in space. In this case, you describe the intermediate and end points with three coordinates.



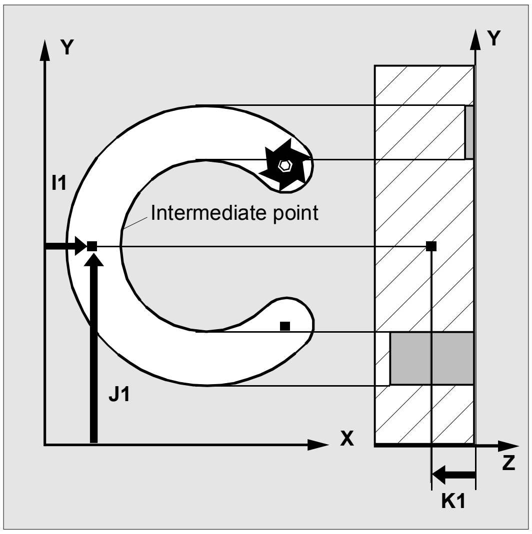
The circular movement is described by:

- The intermediate point at addresses I1=, J1=, K1= and
- the end point in Cartesian coordinates X, Y, Z.

The identifiers have the following meanings:
I1=: Coordinate of the interm. point in the X direction
J1=: Coordinate of the interm. point in the Y direction
K1=: Coordinate of the interm. point in the Z direction

Input in absolute and incremental dimensions

The G90/G91 defaults for absolute or incremental dimensions are valid for the intermediate and circle end points.
With G91, the circle starting point is used as the reference for the intermediate point and end point.



CIP is modal.

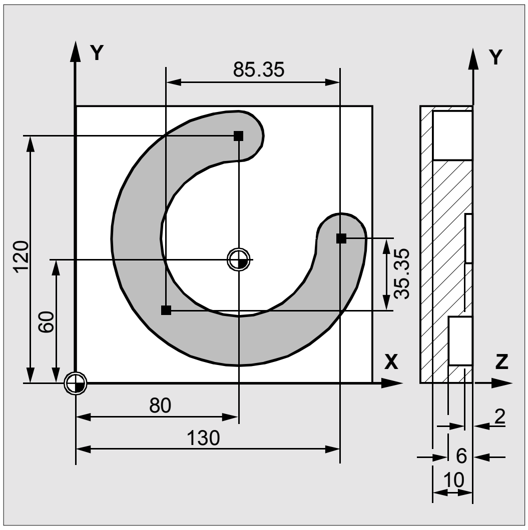
The traversing direction is determined by the order of the starting point, intermediate point and end point.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example for CIP

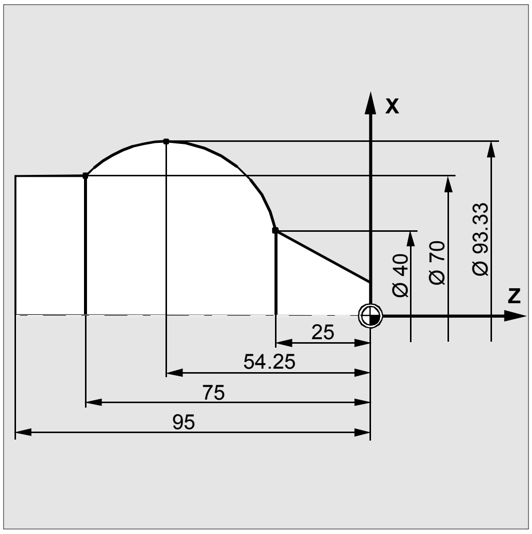
In order to machine an inclined circular groove, a circle is described by specifying the intermediate point with 3 interpolation parameters, and the end point with 3 coordinates.

Milling:



`N10 G0 G90 X130 Y60 S800 M3`	Approach starting point
`N20 G17 G1 Z-2  F100`	Tool infeed
`N30 CIP X80 Y120 Z-10` `    I1= IC(-85.35) J1=IC(-35.35) K1=-6`	Circle end point and intermediate point: Coordinates for all three geometry axes
`N40 M30`	End of program

Turning:




```
N125 G1 X40 Z-25 F0.2
N130 CIP X70 Z-75 I1=IC(26.665)
     K1=IC(-29.25)
```

or

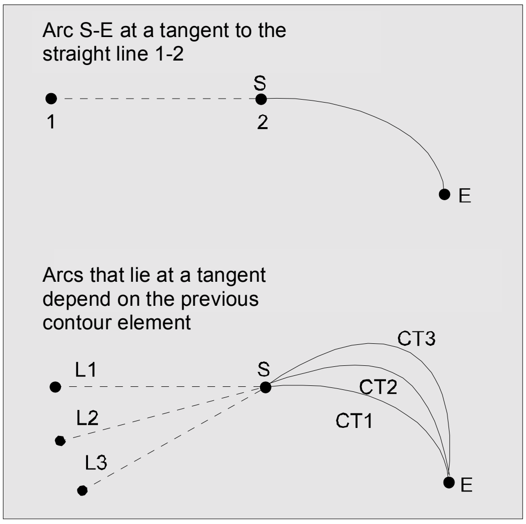
```
N130 CIP X70 Z-75 I1=93.33 K1=-54.25
N135 G1 Z-95
```

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming a circle with tangential transition

The Tangential transition function is an expansion of the circle programming.
The circle is defined by

- the start and end point and
- the tangent direction at the start point.



The G code CT produces an arc that lies at a tangent to the contour element programmed previously.

CT is modal.

As a rule, the direction of the tangent, as well as the start and end point of the circle are uniquely defined.

Position of the circle plane

The position of the circle plane depends on the active plane (G17-G19).

If the tangents of the previous block do not lie in the active plane, their projection in the active plane is used.

If the start and end points do not have the same position components perpendicular to the active plane, a helix is produced instead of a circle.

Specifying TURN=... enables you to program circles with more than one full rotation.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Determining the direction of the tangent
The direction of tangent at the start point of a CT block is determined from the end tangent of the programmed contour of the previous block with a traversing movement.

Any number of blocks without traversing information may lie between this block and the current block.

In the case of splines, the tangential direction is defined by the straight line through the last two points. This direction is generally not the same as the direction at the end point of the splines for A and C splines with active ENAT or EAUTO.

The transition of B splines is always tangential, the tangent direction is defined as for A or C splines and active ETAN.

Frame change
If a frame change takes place between the block defining the tangent and the CT block, the tangent is also subjected to this change.

Limit case
If the extension of the start tangent runs through the end point, a straight line is produced instead of a circle (limit case: circle with infinite radius). In this special case, TURN must either not be programmed or the value must be TURN=0.

Additional notes

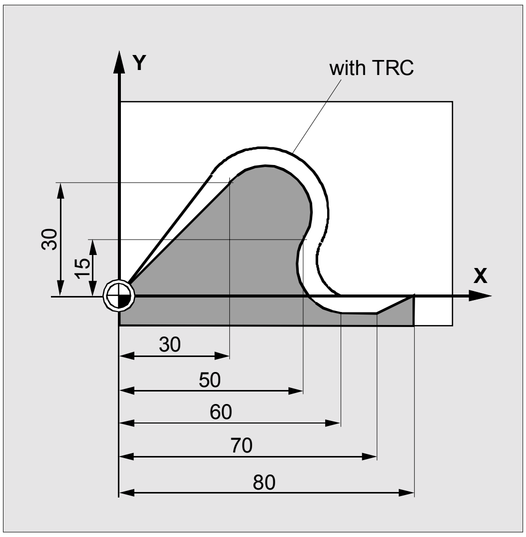
When the values tend towards this limit case, circles with an unlimited radius are produced and machining with TURN unequal 0 is generally aborted with an alarm due to violation of the software limits.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example for CT

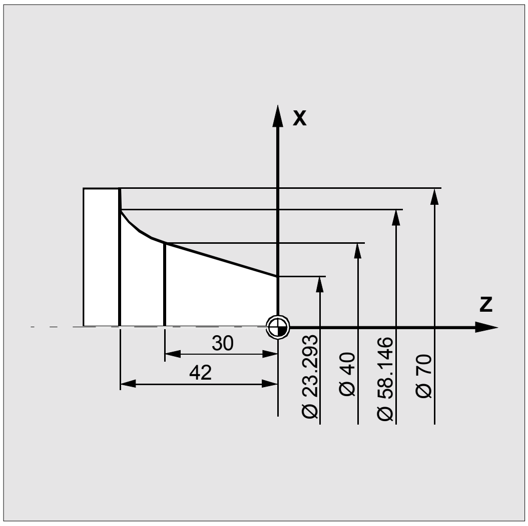
Mill a circular arc following a straight line:

Milling:



`N10 G0 X0 Y0 Z0 G90 T1 D1`	
`N20 G41 X30 Y30 G1 F1000`	Activate tool radius compensation (TRC)
`N30 CT X50 Y15`	Program circle with tangential transition
`N40 X60 Y-5`	
`N50 G1 X70`	
`N60 G0 G40 X80 Y0 Z20`	
`N70 M30`	

Turning:



`N110 G1 X23.293 Z0 F10`	
`N115 X40 Z-30 F0.2`	
`N120 CT X58.146 Z-42`	Program circle with tangential transition
`N125 G1 X70`	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.6 Helical interpolation, G2/G3, TURN

Programming

```
G2/G3 X… Y… Z… I… J… K… TURN=
G2/G3 X… Y… Z… I… J… K… TURN=
G2/G3 AR=… I… J… K… TURN=
G2/G3 AR=… X… Y… Z… TURN=
G2/G3 AP… RP=… TURN=
```

Explanation of the commands and parameters

G2	Travel on a circular path in clockwise direction
G3	Travel on a circular path in counterclockwise direction
X Y Z	End point in Cartesian coordinates
I J K	Circle center point in Cartesian coordinates
AR	Arc angle
TURN=	Number of additional circle passes within the range 0 to 999
AP=	Polar angle
RP=	Polar radius

Function

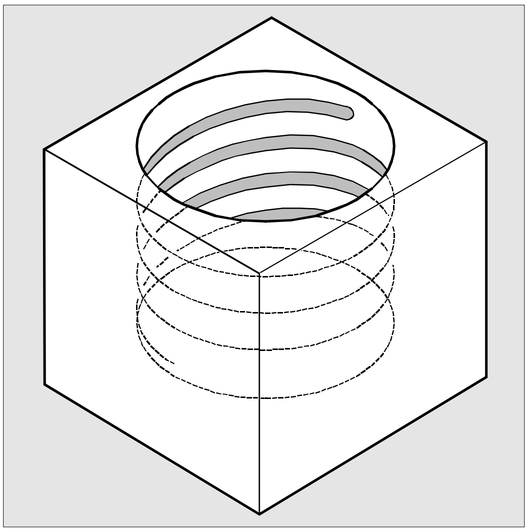
Helical interpolation (Helical interpolation) can be used to manufacture threads and oil grooves, for example.

Sequence

In helical interpolation, two movements are superimposed and executed in parallel:

- A horizontal circular movement on which
- a vertical linear movement is superimposed.

The circular movement is performed on the axes specified by the working plane.
Example: Working plane G17, axes for circular interpolation X and Y.
The infeed movement is performed on the perpendicular infeed axis, in this case Z.

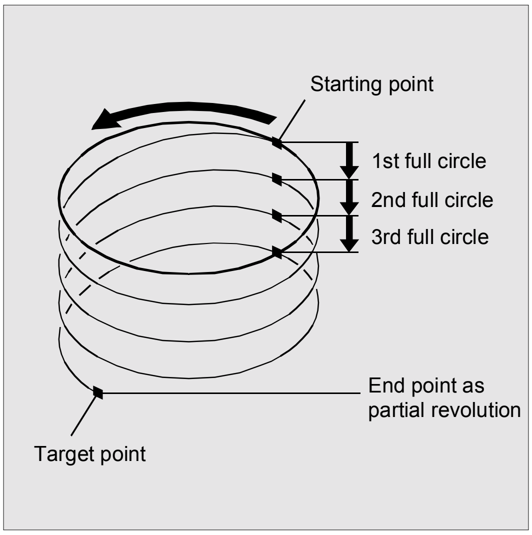


840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence of motions

1. Approach starting point
2. With TURN= execute the full circles programmed
3. Approach the circle end point, e.g. as a partial revolution
4. Execute steps 2 and 3 across the infeed depth.

The lead with which the helix is to be machined is calculated from the number of full circles plus the programmed end point – executed across the infeed depth.



Programming the end point for helical interpolation

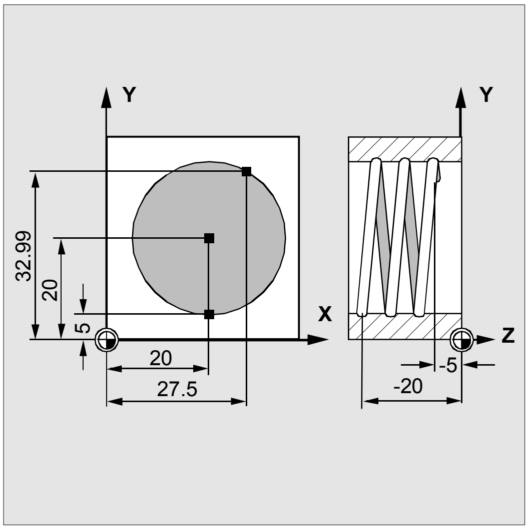
Please refer to circular interpolation for a detailed description of the interpolation parameters.

Additional notes

For helical interpolation, it is advisable to specify a programmed feedrate override (CFC). You will find more information in Chapter 5.

Programming example

Helical interpolation



`N10 G17 G0 X27.5 Y32.99 Z3`	Approach start position
`N20 G1 Z-5 F50`	Tool infeed
`N30 G3 X20 Y5 Z-20 I=AC(20) J=AC (20) TURN=2`	Helix with following parameters: Execute 2 full circles from start position, then approach end point
`N40 M30`	End of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.7 Involute interpolation, INVCW, INVCCW

Programming

INVCW X... Y... Z... I... J... K... CR=...
INVCCW X... Y... Z... I... J... K... CR=...
INVCW I... J... K... CR=... AR=...
INVCCW I... J... K... CR=... AR=...

Explanation of the commands and parameters

`INVCW`	Travel on an involute in clockwise direction
`INVCCW`	Travel on an involute path in counterclockwise direction
`X Y Z`	End point in Cartesian coordinates
`I J K`	Center point of base circle in cartesian coordinates
CR=	Radius of base circle
AR=	Arc angle (angle of rotation)

Function

The involute of the circle is a curve traced out from the end point on a "piece of string" unwinding from the curve. Involute interpolation allows trajectories along an involute.
When paths perpendicular to the active plane are also programmed, it is possible to traverse an involute in space (comparable to helical interpolation with circles).

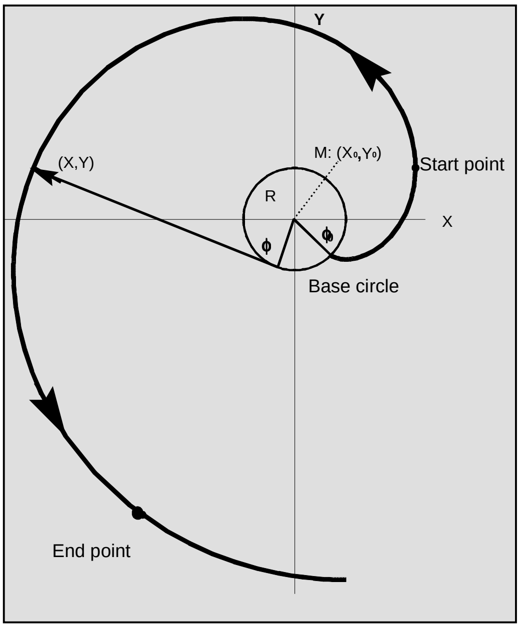
Additional notes

For further information about machine data and supplementary conditions that are relevant with respect to involute interpolation, please see
References: /FB1/, A2 Subsection 2.12.2
Settings for involute interpolation.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

Involute interpolation takes place in the plane in which the base circle is defined. Start and end points outside this plane result in superimposition on a curve in space, analogous to helical interpolation with circles.



Supplementary condition

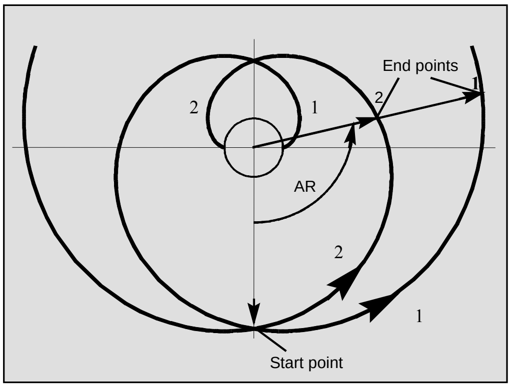
Both the start point and the end point must be outside the area of the base circle of the involute (circle with radius CR around the center point determined by I, J and K). If this condition is not fulfilled, an alarm is generated and the program run aborted.

Programming methods

1. Direct programming of the end point with X, Y or X, Y, Z

2. Programming of the angle of rotation between the start and end vectors with AR=angle (cf. also programming of the arc angle when programming circles). If the angle of rotation is positive (AR > 0), the path on the involute moves away from the base circle; with a negative angle of rotation (AR < 0), the path on the involute moves towards the base circle. The maximum angle of rotation for AR < 0 is restricted by the fact that the end point must always lie **outside** the base circle.

Options 1. and 2. are mutually exclusive. Only one of these notations may be used each block.



Additional notes

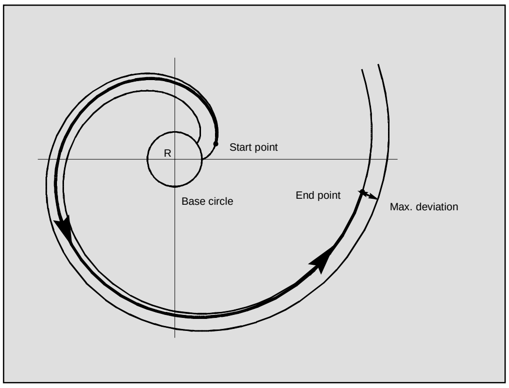
There are further options when the angle of rotation is programmed with AR. Two involutes can be implemented (see diagram) by specifying the radius and center point of the base circle as well as the start point and direction of rotation (INVCW/INVCCW).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

The selected path must be defined unambiguously by the sign of the angle.
The diagram above shows the two involutes defined by the start point and base circle. In this example, end point 1 is approached when AR > 0 is programmed and end point 2 with AR < 0.

Accuracy

If the programmed end point does not lie exactly on the involute defined by the start point and base circle, interpolation takes place between the two involutes defined by the start or end point (see diagram). The maximum deviation of the end point is determined by a machine data. If the programmed end point deviates more in the radial direction than the MD setting, an alarm is generated and the program run aborted.



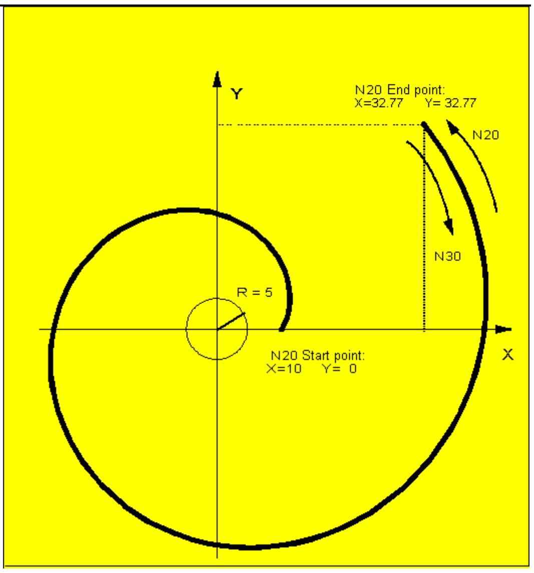
Programming examples

Example 1

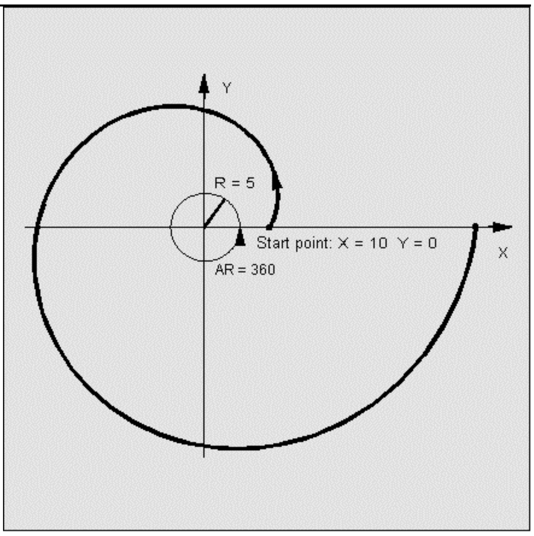
Counterclockwise involute according to programming method 1 from start to end point and back again (clockwise involute)

`N10 G1 X10 Y0 F5000`	Approach start position
`N15 G17`	Select X/Y plane
`N20 INVCCW X32.77 Y32.77 CR=5 I-10 J0`	I. counterclockwise, end point, radius, center point relative to start point
`N30 INVCW X10 Y0 CR=5 I-32.77 J-32.77`	Start point is end point from N20 End point is start point from N20, radius, center point ref. to new start point is same as old center point
`...`	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di



Example 2	Specification of end point via angle of rotation
`N10 G1 X10 Y0 F5000`	Approach start position
`N15 G17`	Select X/Y plane
`N20 INVCCW CR=5 I-10 J0 AR=360`	Counterclockwise involute, away from base circle (pos. angle setting) with one full rotation
`...`	



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.8 Contour definitions

4.8.1 Straight line with angle

Programming

```
X2… ANG…
```

Explanation of the commands and parameters

X2 or Z2	End point in Cartesian coordinates X or Z
ANG	Angle

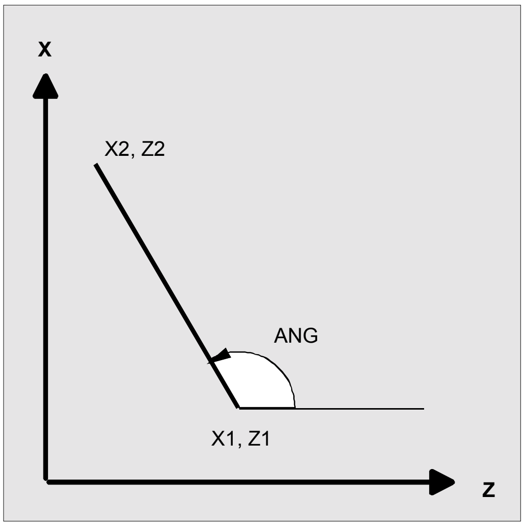
Machine manufacturer

The names for angle (ANG), radius (RND) and chamfer (CHR) can be set in MD, see /FBFA/ FB ISO Dialects, Chapter 6.

Function

The end point is defined through specification of

- the angle ANG and
- one of the two coordinates X2 or Z2.



Programming example

N10 X5 Z70 F1000 G18	Approach start position
N20 X88.8 ANG=110 or (Z39.5 ANG=110)	Straight line with specified angle
N30 ...	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.8.2 Two straight lines

Programming

```
ANG1…            or    X1… Z1…
X3… Z3 ANG2…           X3… Z3…
```

Explanation of the commands and parameters

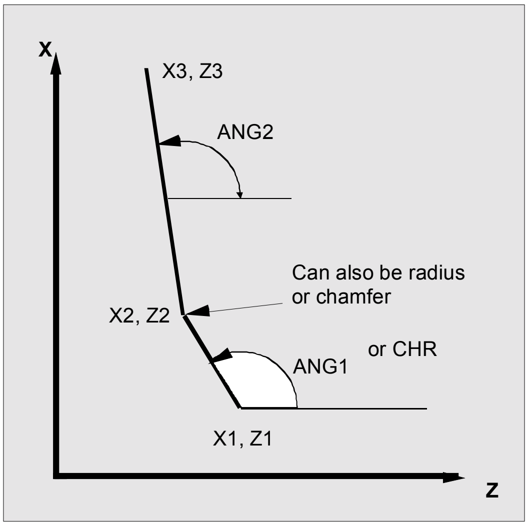
ANG1	Angle of the first straight line
ANG2	Angle of the second straight line
CHR	Chamfer
X1, Z1	Start coordinates
X2, Z2	Intersection of the two straight lines
X3, Z3	End point of the second straight line

Machine manufacturer

The names for angle (ANG), radius (RND) and chamfer (CHR) can be set in MD, see /FBFA/ FB ISO Dialects, Chapter 6.

Function

The intersection of the two straight lines can be designed as a corner, curve or chamfer. The end point of the first of the two straight lines can be programmed by defining the coordinates or specifying the angle.



Programming example

N10 X10 Z80 F1000 G18	Approach start position
N20 ANG1=148.65 CHR=5.5	Straight line with specified angle and chamfer
N30 X85 Z40 ANG2=100	Straight line with specified angle and end point
N40 ...	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.8.3 Three straight lines

Programming

`X2… Z2…`	or	`ANG1…`
`X3… Z3…`		`X3… Z3… ANG2…`
`X4… Z4…`		`X4… Z4…`

Explanation of the commands and parameters

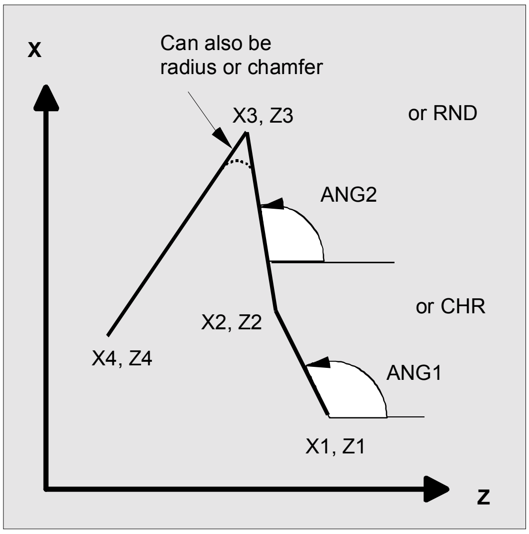
`ANG, ANG2`	Angle of the first/second straight line relative to the abscissa
`CHR`	Chamfer
`RND`	Rounding
`X1, Z1`	Start coordinates of the first straight line
`X2, Z2`	Endpoint coordinates of the first straight line or start point of the second straight line
`X3, Z3`	Endpoint coordinates of the second straight line or start point of the third straight line
`X4, Z4`	Endpoint coordinates of the third straight line

Machine manufacturer

The names for angle (ANG), radius (RND) and chamfer (CHR) can be set in MD, see /FBFA/ FB ISO Dialects, Chapter 6.

RND function

The intersection of the straight lines can be designed as a corner, a curve or a chamfer. The end point of the third straight line must always be programmed as Cartesian.



Programming example

`N10 X10 Z100 F1000 G18`	Approach start position
`N20 ANG1=140 CHR=7,5`	Straight line with specified angle and chamfer
`N30 X80 Z70 ANG2=95.824 RND=10`	Straight line on intersection with specified angle and rounding
`N40 X70 Z50`	Straight line on end point

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.8.4 End point programming with an angle

Function

If the address letter A appears in an NC block, either none, one or both of the axes in the active plane may also be programmed.
If none of the axes in the active plane is programmed, the block is either the first or second block of a contour consisting of two blocks. If it is the second block of this kind of contour, then this means that the start and end points in the active plane are identical. The contour then comprises at most one motion perpendicular to the active plane.
If exactly one axis of the active plane is programmed, it is either a single straight line with an end point uniquely defined by the angle and the programmed coordinates, or the second block of a contour that consists of two blocks. In the latter case, the missing coordinate is set to match the last (modal) position reached.
If two axes are programmed in the current plane, it is the second block of a contour that consists of two blocks. If the current block was not preceded by a block with angle programming and no programmed axes in the current plane, the block in question is not permissible.
Angle A must only be programmed for linear or spline interpolation.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.9 Thread cutting with constant lead, G33

Programming example for turning machine with longitudinal axis Z and transverse axis X

Cylindrical thread

`G33 Z… K … SF=…*`

Taper thread

`G33 X… Z… K… SF=…*` (K for taper angle < 45°)
`G33 X… Z… I… SF=…*` (I for taper angle > 45°)

Face thread

`G33 X… I… SF=…*`

* SF= only needs to be programmed for multiple threads

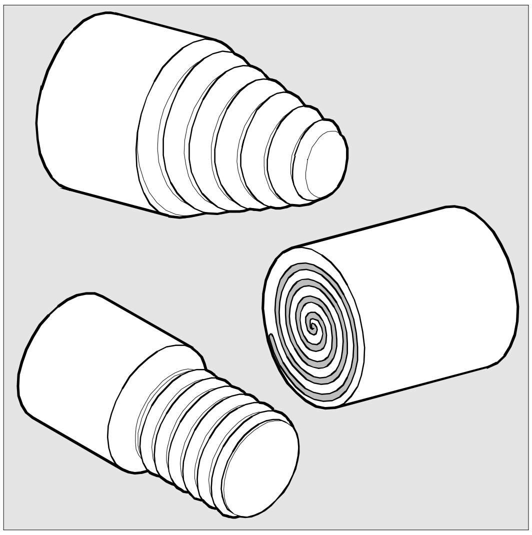
Explanation of the parameters

X Z	End point in Cartesian coordinates
I K	Thread lead (in direction X, Z)
SF=	Starting point offset, only needed for multiple threads

Function

The following types of thread can be machined with G33:
Cylindrical, taper or face threads, single or multiple threads, right or left-handed threads.

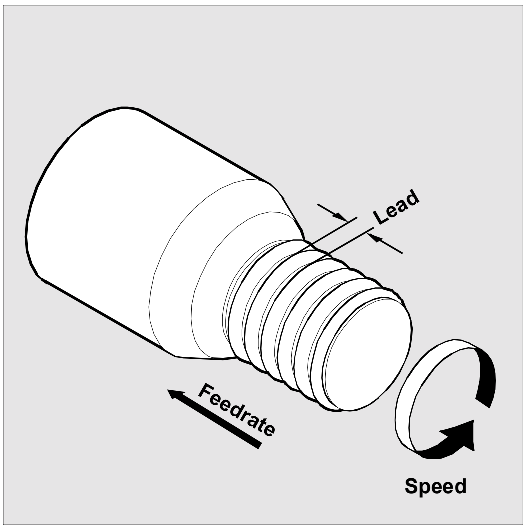
Equipment required: speed-controlled spindle with position measurement system.



4.9 Thread cutting with constant lead, G33

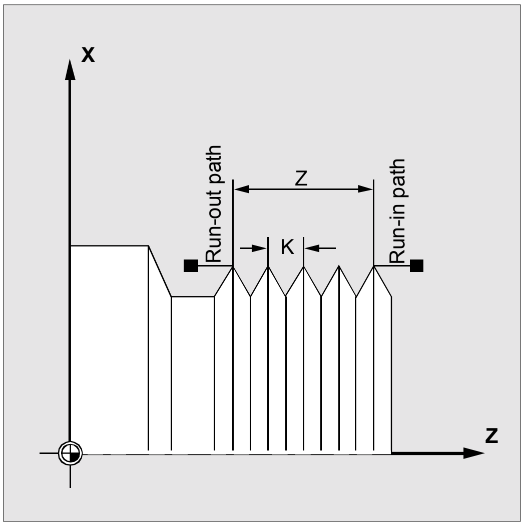
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence



Operating principle

The control calculates the required feedrate from the programmed spindle speed and the thread lead. The turning tool traverses across the length of the thread in the longitudinal and/or facing direction at this feedrate. The feedrate F is not considered for G33, the limitation to maximum axis speed (rapid traverse) is monitored by the control.



Cylinder thread

A cylinder thread is described by the thread length and thread lead.

The thread length is entered in absolute or incremental dimensions with one of the Cartesian coordinates X, Y or Z. The Z direction is used preferentially on turning machines. Allowance must also be made for the run-in and run-out paths across which the feed is accelerated or decelerated.

The thread lead is entered at addresses I, J, K, on turning machines preferentially with K.

The identifiers have the following meanings:

I	Thread lead in X direction
J	Thread lead in Y direction
K	Thread lead in Z direction

Example: K4 means 4mm lead per revolution

Value range of lead:
0.001 to 2000.00 mm/revolution

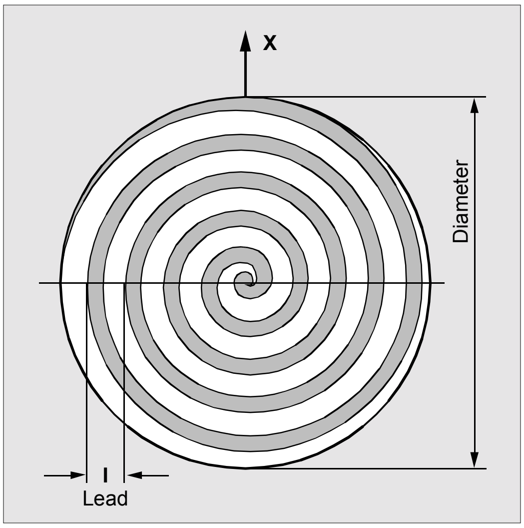
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Face thread

The face thread is described by

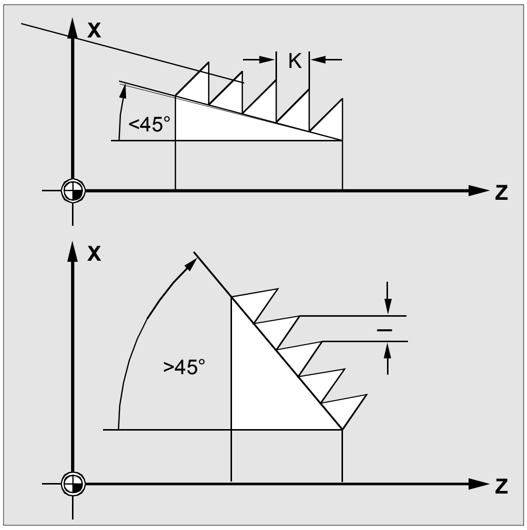
- Thread diameter, preferentially in X direction and
- Thread lead, preferentially with **I**.

Otherwise, the procedure is the same as for cylindrical threads.



Taper thread

The taper thread is described by the end point in the longitudinal and facing direction (taper contour) and the thread lead.



The taper contour is entered in Cartesian coordinates X, Y, Z in absolute or incremental dimensions – preferentially in the X and Z direction for machining on turning machines. Allowance must also be made for the run-in and run-out paths across which the feed is accelerated or decelerated.

The thread lead is entered at addresses **I**, J, K. Please see the cylindrical thread for the meaning of **I**, J, K.

The parameter for the lead is based on the taper angle (calculated from the longitudinal axis to the outside of the taper).

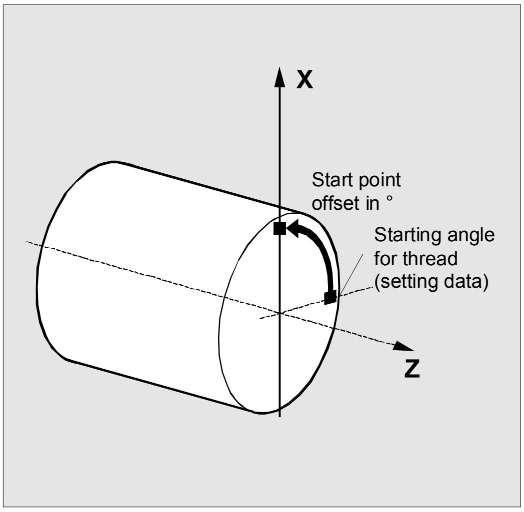
For taper angles < 45°: Lead in longitudinal direction, e.g. K
For taper angles > 45°: Lead in facing direction, e.g. **I**
For taper angles = 45° you can specify **I** or K.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Start point offset SF – production of multi-turn threads
Threads with offset cuts are programmed by specifying starting point offsets in the G33 block.

The start point offset is specified as an absolute angular position at address SF=. The associated setting data is changed accordingly.

Example: SF=45
Meaning: Start offset 45°
Value range: 0.0000 to 359.999 degrees



If no starting point offset is specified, the "starting angle for thread" defined in the setting data is used.

Right-hand/left-hand thread
Right-hand or left-hand threads are set according to the spindle direction:

M3: Clockwise
M4: Counterclockwise

In addition, the desired speed is programmed at the address S.

The spindle speed override switch must not be changed during thread cutting with G33 (dynamic speed change).

The feed override switch has no function in the G33 block.

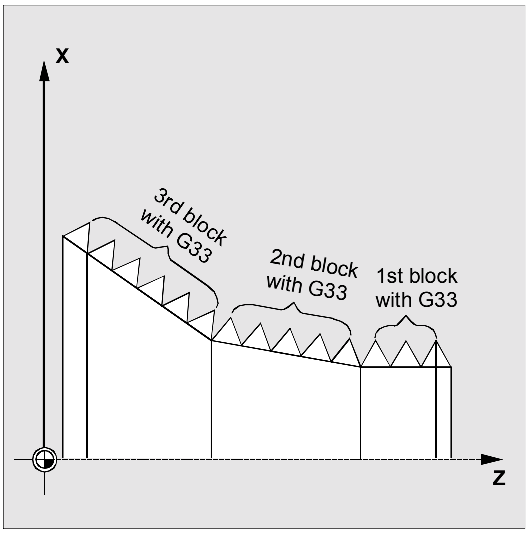
Using a position-controlled spindle
The SPCON command can be used before G33 to produce a thread in position-control mode.
You will find more information on SPCON in Chapter 7.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Thread chaining

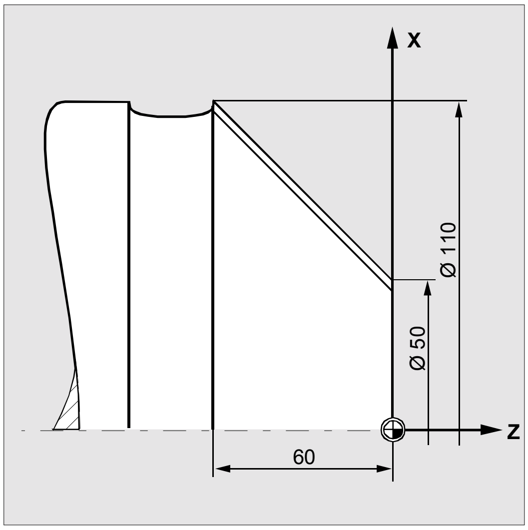
By programming several G33 blocks consecutively, you can align several sets of threads in a sequence. With G64 continuous-path mode, the blocks are interconnected in a look ahead velocity control so that no speed jumps are produced.

You will find more information on G64 in Chapter 7.



Programming example

Machining a taper thread

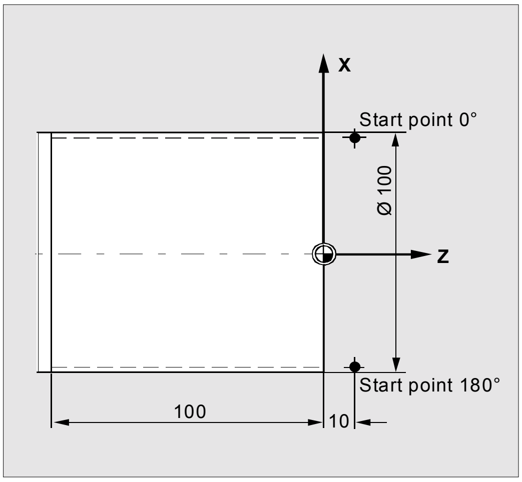


N10	G1 X50 Z0 S500 F100 M3	Approach starting point, activate spindle
N20	G33 X110 Z-60 K4	Taper thread: End point on Z and X, lead K in Z direction, since angle < 45°
N30	G0 Z0 M30	Retraction, end of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

Machining a double cylindrical thread in offset steps with starting point offset 180°.



N10	G1 G54 X99 Z10 S500 F100 M3	Zero offset, approach start point, spindle on
N20	G33 Z-100 K4	Cylindrical thread: End point in Z
N30	G0 X102	Retract to starting position
N40	G0 Z10	
N50	G1 X99	
N60	G33 Z-100 K4 SF=180	2nd cut: Starting point offset 180°
N70	G0 X110	Retract tool
N80	G0 Z10	End of program
N90	M30	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.9.1 Programmable run-in and run-out path (SW 5 and higher)

Programming

```
DITS=value
DITE=value
```

Explanation of the parameters

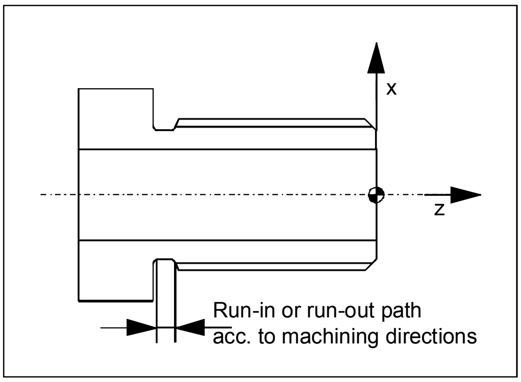
`DITS`	Thread run-in path
`DITE`	Thread run-out path
`Value`	Specification of the run-in and run-out path: -1,0,...n

Function

The commands **DITS** (Displacement Thread Start) and **DITE** (Displacement Thread End) can be used to define the path ramp for acceleration and deceleration, in order to modify the feedrate if the tool run-in and run-out paths are too short:

- **Run-in path too short**:
  The band at the thread run-in provides insufficient space for the tool start ramp – a shorter ramp must therefore be defined with DITS.
- **Run-out path too short**:
  The band at the thread run-out provides insufficient space for the tool deceleration ramp, giving rise to **danger of collision** between the workpiece and the tool edge.
  A shorter tool deceleration ramp can be defined with DITE; however a collision can still occur.
  Remedy: Program a shorter thread, reduce the spindle speed.

Only paths, not positions, are programmed with DITS and DITE.



Machine manufacturer (MH4.1)

The commands DITS and DITE correspond to setting data THREAD_RAMP_DISP[0,1], in which the programmed paths are written:
See /FB/ V1 Feeds.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

If the run-in and/or run-out path is very short, the acceleration of the thread axis is higher than the configured value. This causes an acceleration overload on the axis.
In this case, alarm 22280 "Programmed run-in path too short" is output for the thread run-in (if configured in MD 11411 ENABLE_ALARM_MASK). The alarm is purely for information and has no effect on parts program execution.

Additional notes

- DITE acts at the end of the thread as an approximate distance. This achieves a smooth change in the axis movement.
- When a block with the command DITS and/or DITE is loaded into the interpolator, the path programmed in DITS is copied into SD 42010 THREAD_RAMP_DISP[0] and the path programmed in DITE is copied into SD 42010 THREAD_RAMP_DISP[1].
- The programmed run-in path is handled according to the current setting (inches, metric).

Machine manufacturer (MH4.2)

If no run-in/deceleration path is programmed before or in the first thread block, the value is determined by the setting in SD 42010;
see References: /FB/ V1 Feeds.
MD 10710: PROG_SD_RESET_SAVE_TAB is used to set the value written by the parts program into the corresponding setting data on RESET. That retains the values through a power ON.

Programming example

`N...`	
`N40 G90 G0 Z100 X10 SOFT M3 S500`	
`N50 G33 Z50 K5 SF=180 DITS=1 DITE=3`	;Start of corner rounding with Z=53
`N60 G0 X20`	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.10 Linear progressive/degressive thread pitch change, G34, G35 (SW 5.2 and higher)

Programming

`G34 X… Y… Z… I… J… K… F…` Progressive change in thread pitch (tapping with linear pitch increase)

`G35 X… Y… Z… I… J… K… F…` Degressive change in thread pitch (tapping with linear pitch decrease)

Explanation of the parameters

X Y Z	End points in Cartesian coordinates
I J K	Thread lead (in X, Y, Z direction)
F	Thread pitch change (in mm/rev^2)

Function

Functions G35/G35 can be employed to produce self-cutting threads.

Both the functions G34 and G35 offer the functionality of G33, but provide the additional option of programming a pitch change under F.

Sequence

If you already know the initial and final pitch of a thread, you can calculate the pitch change to be programmed according to the following equation:

$$F = \frac{|k^2_e - k^2_a|}{2*I_G} \quad [mm/rev^2]$$

The identifiers have the following meanings:

k_e Pitch change of axis target point coordinate [mm/rev]

k_a Initial thread pitch (progr. under I, J and K) [mm/rev]

I_G Thread length in [mm]

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

`N1608 M3  S10`	;Spindle speed
`N1609 G0  G64  Z40  X216`	;Approach start point and thread
`N1610 G33  Z0  K100  SF=R14`	;With constant pitch 100mm/rev
`N1611 G35  Z-200  K100  F17.045455`	;Pitch decrease 17.0454mm/rev^2 ;Pitch at block end 50mm/rev
`N1612 G33  Z-240  K50`	;Traverse thread block without jerk
`N1613 G0  X218`	;
`N1614 G0  Z40`	;
`N1615 M17`	;

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.11 Rigid tapping, G331, G332

Programming

```
G331  X… Y… Z… I… J… K…        (Tapping)
G332  X… Y… Z… I… J… K…        (Tapping retraction)
```

Explanation of the parameters

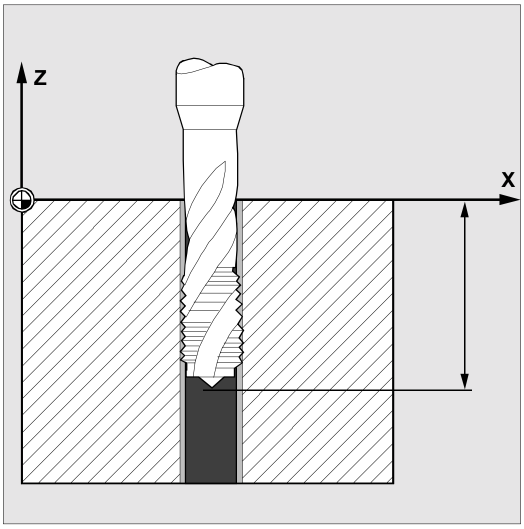
X Y Z	Drilling depth (end point) in a Cartesian coordinate
I J K	Thread lead (in X, Y, Z direction)

Function

G331/G332 enable rigid tapping.
Equipment required: Position-controlled spindle with position measuring system

Sequence

The spindle must be prepared for tapping with SPOS/SPOSA. You will find more information in Chapter 7.



G331: Tapping
Tapping is described by the drilling depth (end point of the thread) and the lead.

G332: Retraction movement
This movement is described with the same lead as the G331 movement. The reversal in the direction of the spindle is performed automatically.
Drilling depth, thread lead
Drilling in X direction, thread lead I
Drilling in Y direction, thread lead J
Drilling in Z direction, thread lead K
Value range of lead:
±0.001 to 2000.00 mm/revolution

Right-hand/left-hand threads
Right-hand or left-hand threads are defined in axis mode by the sign qualifying the lead:

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Positive lead, clockwise (same as M3)
Negative lead, counterclockwise (same as M4)

The desired speed is also programmed at address S.

Additional notes

Both functions are modal.
The spindle does not operate in axis mode but as a position-controlled spindle. You will find more information on handling of the position-controlled spindle in Chapter 5.

Programming example

After G332 (retraction), the next thread can be tapped with G331.

N10	SPOS[n]=0	Prepare tapping
N20	G0 X0 Y0 Z2	Approach starting point
N30	G331 Z-50 K-4 S200	Tapping, drilling depth 50, lead K negative = direction of spindle rotation counterclockwise
N40	G332 Z3 K-4	Retract, automatic reversal of direction
N50	G1 F1000 X100 Y100 Z100 S300 M3	Spindle reverts to operation in spindle mode
N60	M30	End of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.12 Tapping with compensating chuck G63

Programming

`G63     X… Y… Z…`

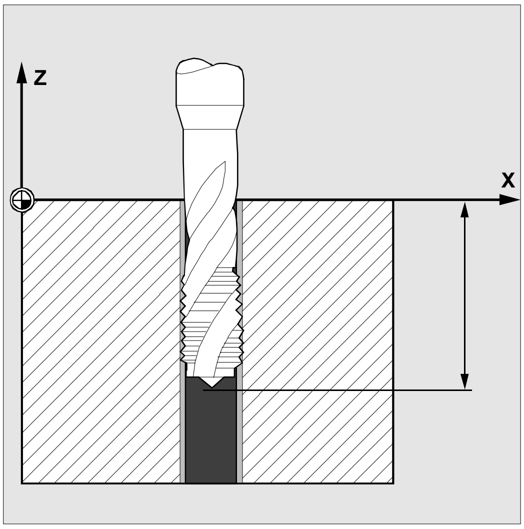
Explanation of the parameters

X Y Z	Drilling depth (end point, specified in Cartesian coordinates)

Function

You can use G63 to tap threads with compensating chuck.
The chuck compensates for any deviations occurring in the path.

Sequence



Tapping
The following are programmed

- Drilling depth in Cartesian coordinates
- Spindle speed and direction
- Feed

Retraction movement
Also programmed with G63, but with the reverse direction of spindle rotation.

Feedrate

The programmed feed must match the ratio of the speed to the thread lead of the tap.

Thumb rule:
Feed F in mm/min = spindle speed S in rpm x thread lead in mm/rev

Both the feed and the spindle speed override switch are set to 100% with G63.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Additional notes

G63 is modal.

The last programmed interpolation command G0, G1, G2, ... is reactivated after a block with programmed G63.

Programming example

Tapping with compensating chuck:
In this example, an M5 thread is to be drilled. The lead of an M5 thread is 0.8 (specified in table).

With a selected speed of 200rpm, the feed F is 160 mm/min.

N10	G1 X0 Y0 Z2 S200 F1000 M3	Approach starting point, activate spindle
N20	G63 Z-50 F160	Tap, drilling depth 50
N30	G63 Z3 M4	Retract, programmed reversal of direction
N40	M30	End of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.13 Stop during thread cutting

Programming

```
LFON
LFOF
DILF
```

Explanation of the parameters

`LFON`	Enable fast retraction for thread cutting (G33)
`LFOF`	Disable fast retraction for thread cutting (G33)
`DILF`	Determine retraction path (length)

Function

The function produces a nondestructive interruption with thread cutting (G33). The function cannot be used with tapping (G33). With mixed use of both G33 functions, the response can be parameterized for NC Stop/NC Reset via the machine data.

Trigger criteria for retraction

- Fast inputs, programmable with SETINT LIFTFAST (if LIFTFAST option enabled)
- NC Stop/NC Reset

If fast retraction is enabled with LFON, it is active for every retraction movement

Retraction path (DILF)

The retraction path can be defined in the machine data or by programming. After NC Reset, the value in MD 21200: LIFTFAST_DIST is still active.

Retraction direction (SW 4.2 and lower)

The retraction direction is determined within the threading. Retraction is always perpendicular to the direction of machining. **ALF** is not active.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Retraction direction (SW 4.3 and higher)

The retraction direction in connection with **ALF** is controlled using the following keywords:

- **LFTXT**
  The plane in which the fast retraction is executed is calculated from the path tangent and the tool direction (default setting).
- **LFWP**
  The plane in which the fast retraction is executed is the active working plane.

The direction is programmed as before in discrete steps of 45 degrees with ALF in the plane of the retraction motion. With **LFTXT**, the retraction is defined in the tool direction for ALF=1.

With **LFWP**, the direction in the working plane is assigned as follows:

• **G17**:	X/Y plane	ALF=1	Retraction in X direction
		ALF=3	Retraction in Y direction
• **G18**:	Z/X plane	ALF=1	Retraction in Z direction
		ALF=3	Retraction in X direction
• **G19**:	Y/Z plane	ALF=1	Retraction in Y direction
		ALF=3	Retraction in Z direction

Retraction speed

Retraction with maximum axis speed.

Can be configured via machine data.

The maximum permissible acceleration/jerk values are used for traversing; they are configured via the machine data.

Additional notes

Default settings for NC reset and/or NC start in MD 20150: GCODE_RESET_VALUES

LFON or LFOF can always be programmed, they are evaluated only during thread cutting (G33).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming examples

Example 1

N55	M3 S500 G90 **G18**	Active machining plane
...		
N65	MSG ("Thread cutting")	
MM_THREAD:		
N67	$AC_LIFTFAST=0	Reset before beginning of thread
N68	G0 Z5	
N68	X10	
N70	G33 Z30 K5 **LFON DILF=10 LFWP ALF=3**	Enable fast retraction for thread cutting
	Retraction path =10mm, retraction plane Z/X (due to G18) Retraction direction –X (with ALF=3; retraction direction +X)	
N71	G33 Z55 X15 K5	
N72	G1	Deactivate thread cutting
N69	IF $AC_LIFTFAST GOTOB MM_THREAD	If thread cutting was interrupted
N90	MSG("")	
...		
N70 M30		

Example 2

N55	M3 S500 G90 G0 X0 Z0	
...		
N87	MSG ("Tapping")	
N88	**LFOF**	Deactivate fast retraction before tapping.
N89	CYCLE...	Thread drilling cycle with G33
N90	MSG("")	
...		
N99	M30	

4.14 Approaching a fixed point, G75

Programming

```
G75 FP=     X1=0 Y1=0 Z1=0 U1=0  …
```

Explanation of the parameters

FP=	Number of fixed point to be approached
X1= Y1= Z1=	Machine axes to be traversed to the fixed point

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Function

G75 can be used to approach fixed points, such as tool change locations, loading points, pallet changing points, etc.

The positions of the individual points are specified in the machine coordinate system and stored in the machine parameters.

You can approach these positions from any NC program, irrespective of the current tool or workpiece position.

Sequence

The fixed point approach is described by a fixed point and axes which are to be traversed to the fixed point FP.

Number of the fixed point FP=…
If no fixed point number is specified, fixed point 1 is automatically approached.

Two fixed point positions per machine axis can be specified in the machine parameters.

Machine axis addresses X1, Y1 …
Here, you specify with value 0 the axes with which the point is to be approached simultaneously. Each axis traverses at the maximum axial velocity.

Additional notes

G75 is modal.
Kinematic transformation must be deselected before fixed point approach is performed.

Programming example

The tool change location is a fixed point which is defined in the machine data.
This point can be approached in any NC program with G75.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

`N10 G75 FP=2 X1=0 Y1=0 Z1=0`	Retract from fixed point 2 on X, Y and Z, e.g. for tool change
`N20 G75 X1=0`	Approach fixed point X1
`N30 M30`	End of program

Additional notes

SW 5.3 and higher:
G75 "Approach fixed point" applies all offset values (DRF, external ZO and overlaid motion). The fixed point corresponds to the actual value in the MCS. Changes to the DRF and external zero offset while the G75 block is being preprocessed and executed in the main run are not applied. You should prevent this problem by programming STOPPRE in front of the G75 block.

4.15 Travel to fixed stop

Programming

```
FXS[axis]=…
FXST[axis]=…
FXSW[axis]=…
```

Explanation

FXS	Activate/deactivate function "Travel to fixed stop" select/deselect 1 = select; 0 = deselect
FXST	Setting clamping torque Specification in % of maximum drive torque, parameter optional
FXSW	Window width for fixed stop monitoring in mm, inches or degrees; parameter optional
[axis]	Machine axis name

840D
NCU 571

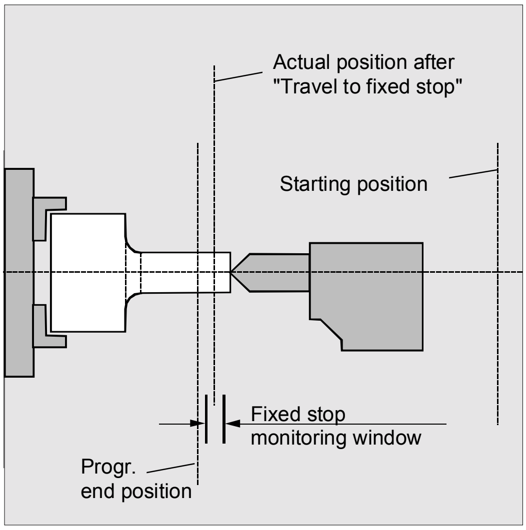
840D
NCU 572
NCU 573

810D

840Di

Function

The "Travel to fixed stop" function (FXS = Fixed Stop) enables generation of a defined power for clamping workpieces as is required, for example, for tailstocks, quills and grippers. The function can also be used to approach mechanical reference points. With sufficiently reduced torque, it is also possible to perform simple measurement operations without connecting a probe.
The "Travel to fixed stop" function can be used for axes and spindles operated as axes.



SW 5 and higher

The limit stop alarm can be suppressed from the parts program where necessary. This is done by masking the alarm in a machine data and then activating the MD by means of NEWCONF.
The "Travel to fixed stop" commands can be called from synchronized actions/technology cycles. They can be activated without initiation of a motion, the torque is limited instantaneously. As soon as the axis is moved via a setpoint, the limit stop monitor is activated.

Torque rise ramp, SW5

A rate of rise ramp for the new torque limit can be defined in MD to prevent any abrupt changes to the torque limit setting (e.g. insertion of a quill).

Link and container axes, SW 5

Travel to fixed stop may also be used on
- link axes and
- container axes.
The status of the assigned machine axis is unaffected by container switches.

References: /FB/ B3, Several Operator Panels and NCUs.

This also applies to modal torque limitation with FOCON (see "Travel with limited torque/force").

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

The commands are modal. Addresses FXST and FXSW are optional: if no parameter is specified, the last programmed value or the value set in the machine data apply.

Machine manufacturer (MH4.3)

Machine axes (X1, Y1, Z1 etc.) are programmed.
(See machine manufacturer's specifications).
Activate travel to fixed stop FXS=1
The movement to the destination point can be described as a path or positioning axis movement. With positioning axes, the function can be performed across block boundaries.

Travel to fixed stop can be performed simultaneously for several axes and parallel to the movement of other axes. The fixed stop must be located between the start and end positions.

Example:
```
X250 Y100 F100 FXS[X1]=1 FXST[X1]=12.3 FXSW[X1]=2
```

Meaning:
Axis X1 travels with feed F100 (parameter optional) to destination position X=250mm. The clamping torque is 12.3% of the maximum drive torque. Monitoring is performed in a 2mm wide window.

From the moment the "Travel to fixed point" function is activated for an axis/spindle, you must no longer program a new position for this axis.

Spindles must be switched to position-controlled mode before the function is selected.

When the fixed stop has been reached:
- The distance-to-go is deleted and the position setpoint is manipulated,
- The drive torque increases to the programmed limit value FXSW and then remains constant,
- Fixed stop monitoring is activated within the specified window width.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Activate from synchronized actions (SW 5)

Example:
If the anticipated event ($R1) occurs and travel to fixed stop is not already operative, then FXS must be activated for axis Y. The torque must correspond to 10% of the rated torque value. The width of the monitoring window is set to the default.

```
N10 IDS=1 WHENEVER (($R1=1) AND ($AA_FXS[Y]==0)) DO $R1=0 FXS[Y]=1 FXST[Y]=10
```

The normal parts program must ensure that $R1 is set at the desired point in time.

Deactivate function FXS=0
Deselection of the function triggers a preprocessor stop. Traversing movements may and should be programmed in a block with FXS=0.

Example:
```
X200 Y400 G01 G94 F2000 FXS[X1] = 0
```
Meaning:
Axis X1 is retracted from the fixed stop to position X= 200 mm.
All other parameters are optional.

The traversing movement to the retraction position must move away from the fixed stop, otherwise damage to the stop or to the machine may result.

The block change takes place when the retraction position has been reached. If no retraction position is specified, the block change takes place immediately the torque limit has been deactivated.

Deactivate from synchronized actions (SW 5)

The function can be deselected from a synchronized action.
Example:
If an anticipated event ($R3) has occurred and the status "Limit stop contacted" (system variable $AA_FXS) is reached, then FXS must be deselected.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

```
N13 IDS=4 WHENEVER (($R3==1) AND ($AA_FXS[Y]==1))
DO FXS[Y]=0 FA[Y]=1000 POS[Y]=0
```

Clamping torque FXST, monitoring window FXSW

A programmed torque limit FXST is effective from the start of the block, i.e. the fixed stop is also approached with reduced torque.

The window must be selected such that only a breakaway from the fixed stop causes the fixed stop monitoring to be addressed.

FXST and FXSW can be programmed or changed at any time in the parts program.

Example:
```
FXST[X1]=34.57
FXST[X1]=34.57 FXSW[X1]=5
FXSW[X1]=5
```

The changes take effect before traversing movements in the same block.

Programming of a new fixed stop monitoring window causes a change not only in the window width but also in the reference point for the center of the window if the axis has moved prior to reprogramming. The actual position of the machine axis when the window is changed is the new window center point.

Additional notes

Combination
"Measure and delete distance-to-go" ("MEAS" command) and "Travel to fixed stop" cannot be programmed in the same block.

Exception:
One function acts on a path axis and the other on a positioning axis or both act on positioning axes.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Contour monitoring

Contour monitoring is not performed while "Travel to fixed stop" is active.

Positioning axes

With "Travel to fixed stop" with POSA axes, the block change takes place independently of the fixed stop movement.

Restrictions

Travel to fixed stop cannot be programmed

- for vertical axes,
  (function can be used with SW 2.2. and higher on 840D with 611D)
- gantry axes,
- for concurrent positioning axes which are controlled exclusively by the PLC (FXS must be selected from the NC program).
- If the torque limit is reduced too far, the axis will not be able to follow the specified setpoint; the position controller then goes to the limit and the contour deviation increases. In this operating state, an increase in the torque limit may result in sudden, jerky movements.
  To ensure that the axis can follow the setpoint, check the contour deviation to make sure it is not greater than the deviation with an unlimited torque.

840D NCU 571 | 840D NCU 572 NCU 573 | FM-NC | 810D | 840Di

4.16 Special turning functions

4.16.1 Position of workpiece

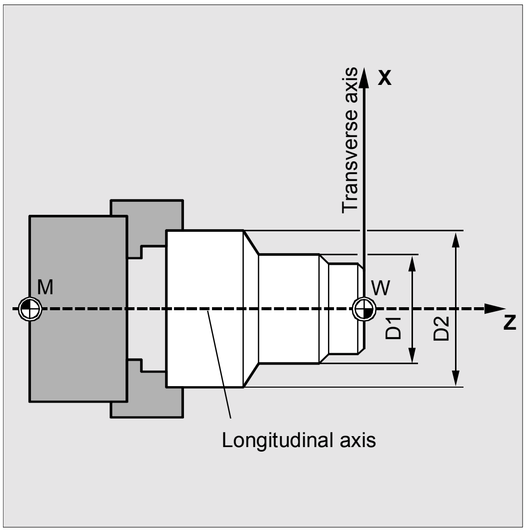
Coordinate system

The two mutually perpendicular geometry axes are usually designated as follows:

- Longitudinal axis = Z axis (abscissa)
- Transverse axis = X axis (ordinate)

The dimensions for the facing axis are generally specified as diameter measurements (double path dimension as compared to other axes).

The geometry axis to be used as a transverse axis is defined in machine data.

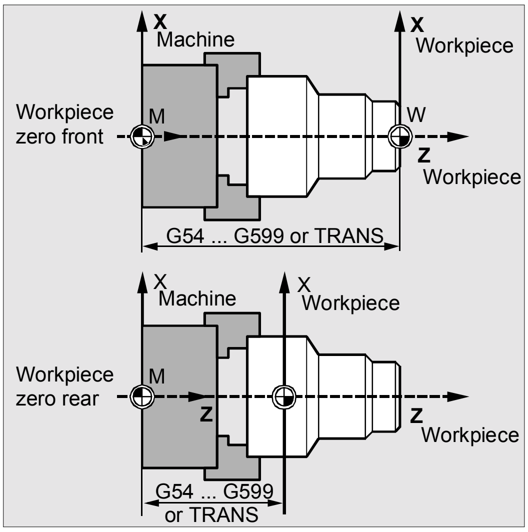


Zero points

Both the machine zero and the workpiece zero are positioned on the center of rotation. The settable offset on the X axis is thus zero.

While the machine zero is fixed, you can choose the position for the workpiece zero on the longitudinal axis. The workpiece zero is generally located on the front or rear side of the workpiece.

The position of the workpiece zero is called with commands G54 to G599 or TRANS.



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.16.2 Dimensions for: Radius, diameter

Programming

```
DIAMON
DIAMOF
DIAM90 (SW 4.4 and higher)
```

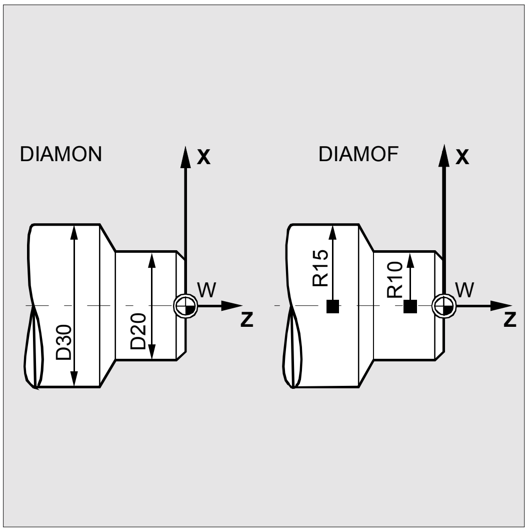
Explanation

	Absolute dimensioning (G90)	*Incremental dimensioning (G91)*
`DIAMOF`	Radius (for default, see machine manufacturer)	Radius
`DIAMON`	Diameter	Diameter
`DIAM90`	Diameter	Radius

Function

The free choice of diameter or radius dimensions allows you to program the dimensions straight from the engineering drawing without conversion.
After activating DIAMON/DIAM90, diameter dimensions are defined for the specified transverse axis.
Diameter values apply to the following data:

- Actual-value display of transverse axis in the workpiece coordinate system
- JOG mode: Increments for incremental dimension and travel with handwheel
- Programming:
  End positions, independent of G90/G91 interpolation parameters for G2/G3, if these are programmed with AC absolute
- Read actual values in workpiece coordinate system for `MEAS, MEAW, $P_EP[X], $AA_IW[X]` (see "Advanced")



By programming DIAMOF you can switch at any time to radius as dimension.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Additional notes

In SW 4.4 and higher, the command DIAM90 sets diameter programming for G90 and radius programming for G91.

After DIAM90 is activated, the actual value of the transverse axis is always displayed as a diameter, irrespective of the type of traversing (G90/G91). This also applies to reading of actual values in the workpiece coordinate system with MEAS, MEAW, $P_EP[x] and $AA_IW[x].

Programming example

`N10 G0 X0 Z0`	Approach starting point
`N20 DIAMOF`	Diameter input off
`N30 G1 X30 S2000 M03 F0.7`	X axis = transverse axis; radius dimensions active Traverse to radius position X30
`N40 DIAMON`	Diameter dimensions active
`N50 G1 X70 Z-20`	Traverse to diameter position X70 and Z–20
`N60 Z-30`	
`N70 DIAM90`	Diameter programming for absolute dimensions and radius programming for incremental dimensions
`N80 G91 X10 Z-20`	Increment
`N90 G90 X10`	Absolute dimensions
`N100 M30`	End of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

4.17 Chamfer, rounding

Programming

```
CHF=…
CHR=…
RND=…
RNDM=…
FRC=…
FRCM=…
```

Explanation of the commands

`CHF=…`	Chamfer the contour corner Value = Length of the chamfer (unit of measurement according to G70/G71)
`CHR=…`	Chamfer the contour corner (SW 3.5 and higher). Programming the chamfer in the original direction of movement. Value = width of chamfer in direction of motion (unit of measurement as above)
`RND=…`	Round the contour corner Value = Radius of the rounding (unit of measurement according to G70/G71)
`RNDM=…`	Modal rounding: Rounding several consecutive contour corners in the same way. Value = radius of roundings (unit of measurement according to G70/G71) 0: Deactivate modal rounding
`FRC=…`	Non-modal feedrate for chamfer/rounding Value = feedrate in mm/min (G94) or mm/rev (G95); FRC > 0
`FRCM=…`	Modal feedrate for chamfer/rounding Value = feedrate in mm/min (G94) or mm/rev (G95) 0: Feedrate programmed at F for chamfer/rounding active

Function

You can insert the following elements at a contour corner:

- Chamfer or
- Rounding

If you wish to round several contour corners sequentially by the same method, use command RNDM "Modal rounding".

You can program the feedrate for the chamfer/rounding with FRC (non-modal) or FRCM (modal).

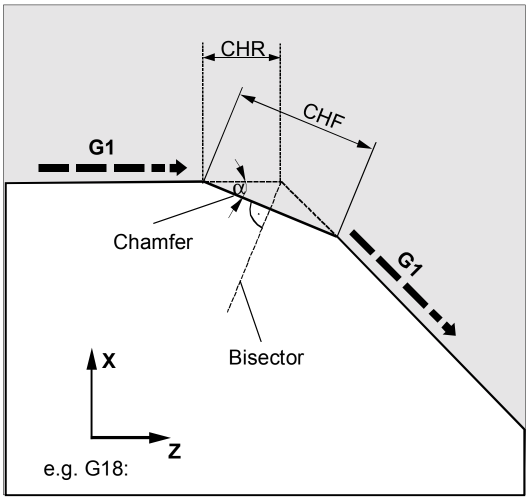
If FRC/FRCM is not programmed, then the normal path feedrate F is applicable.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

Chamfer, CHF/CHR

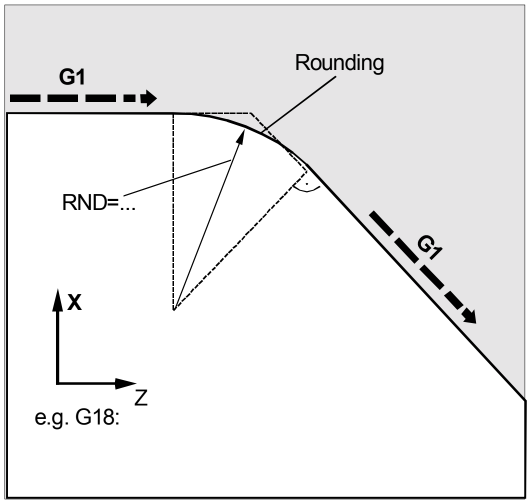
For the chamfer insert another linear part, the chamfer, between the linear and the circle contours in any combination. The chamfer is inserted after the block in which it is programmed. The chamfer is always in the plane activated with G17 to G19.



Example:
```
N30 G1 X… Z… F… CHR=2
N40 G1 X… Z…
or
N30 G1 X… Z… F… CHF=2(cos α ·2)
N40 G1 X… Z…
```

Rounding, RND

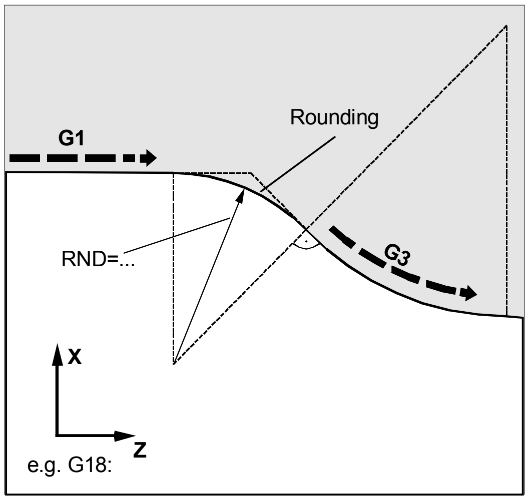
A circle contour element can be inserted with tangential link between the linear and the circle contours in any combination.
The rounding is always in the plane activated with G17 to G19.
The figure to the right shows the rounding between two straight lines.



Example:
```
N30 G1 X… Z… F… RND=2
```

Here, the figure shows the rounding between a straight line and a circle.

```
N30 G1 X… Z… F… RND=2
N40 G3 X… Z… I… K…
```



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Modal rounding, RNDM

This address is for inserting a rounding between linear and circle contours after each traversing block. This is, for example, for deburring sharp workpiece edges.

Example: `N30 G1 X… Z… F… RNDM=2`

Rounding is deactivated with RNDM=0.

Feed FRC (non-modal), FRCM (modal)

To optimize surface quality, it is possible to program a separate feedrate for the chamfer/rounding contour elements.

- FRC is non-modal in this instance,
- FRCM is modal.

See below for examples

Additional info about chamfer/rounding

If the programmed values for chamfer (CHF/CHR) or rounding (RND/RNDM) are too large for the associated contour elements, then the chamfer or rounding are automatically reduced to a suitable value.

No chamfer/rounding is inserted, if

- no straight or circle is available in the plane,
- a movement is taking place outside the plane,
- a plane change is taking place, or
- the number of blocks – which is specified in the machine data – not containing information for traversing (e.g. only command output), is exceeded.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Additional info about FRC/FRCM

- FRC/FRCM has no effect if a chamfer is being machined with G0; the command can be programmed according to the F value without error message.
- The reference to the blocks in which chamfer and rounding are programmed and to the technology is set in machine data.
- FRC is operative only if a chamfer/rounding is programmed in the same block or if RNDM has been activated.
- FRC overwrites the F or FRCM value in the current block.
- The feedrate programmed under FRC must be greater than zero.
- FRCM=0 activates the feed programmed in F for the chamfer/rounding.
- If FRCM is programmed, the FRCM value must be reprogrammed, analogous to F, on changeover G94-95 etc. If only a new F value is programmed, and if FRCM > 0 before the feed type changes, error message 10860 (no feed programmed) will be activated.

Examples

Example 1: MD CHFRND_MODE_MASK Bit 0 = 0: Accept technology from next block (default)

`N10 G0 X0 Y0 G17 F100 G94`	
`N20 G1 X10 CHF=2`	; Chamfer N20-N30 with F=100 mm/min
`N30 Y10 CHF=4`	; Chamfer N30-N40 with FRC=200 mm/min
`N40 X20 CHF=3 FRC=200`	; Chamfer N40-N60 with FRCM=50 mm/min
`N50 RNDM=2 FRCM=50`	
`N60 Y20`	; Modal rounding N60-N70 with FRCM=50 mm/min
`N70 X30`	; Modal rounding N70-N80 with FRCM=100 mm/min
`N80 Y30 CHF=3 FRC=100`	; Chamfer N80-N90 with FRC=50 mm/min (modal)

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

`N90 X40`	; Modal rounding N90-N100 with F=100 mm/min (deselect FRCM)
`N100 Y40 FRCM=0`	; Modal rounding N100-N120 with G95 FRC=1 mm/rev
`N110 S1000 M3`	
`N120 X50 G95 F3 FRC=1`	
`...`	
`M02`	

Example 2: MD CHFRND_MODE_MASK Bit 0 = 1: Accept technology from preceding block (recommended)

`N10 G0 X0 Y0 G17 F100 G94`	
`N20 G1 X10 CHF=2`	; Chamfer N20-N30 with F=100 mm/min
`N30 Y10 CHF=4 FRC=120`	; Chamfer N30-N40 with FRC=120 mm/min
`N40 X20 CHF=3 FRC=200`	; Chamfer N40-N60 with FRCM=200 mm/min
`N50 RNDM=2 FRCM=50`	
`N60 Y20`	; Modal rounding N60-N70 with FRCM=50 mm/min
`N70 X30`	; Modal rounding N70-N80 with FRCM=50 mm/min
`N80 Y30 CHF=3 FRC=100`	; Chamfer N80-N90 with FRC=100 mm/min
`N90 X40`	; Modal rounding N90-N100 with FRCM=50 mm/min
`N100 Y40 FRCM=0`	; Modal rounding N100-N120 with F=100 mm/min
`N110 S1000 M3`	
`N120 X50 CHF=4 G95 F3 FRC=1`	; Chamfer N120-N130 with G95 FRC= 1 mm/rev
`N130 Y50`	; Modal rounding N130-N140 with F=3 mm/rev
`N140 X60`	
`...`	
`M02`	

■

Path action

840D
NCU 571

840D
NCU 572
NCU 573

810D

840Di

5.1 Exact stop, G60, G9, G601, G602, G603

Explanation of the commands

`G60`	Exact stop, modal
`G9`	Exact stop, non-modal
`G601`	Step enable if positioning window fine reached
`G602`	Step enable if positioning window coarse reached
`G603`	Step enable if setpoint (end of interpolation) reached

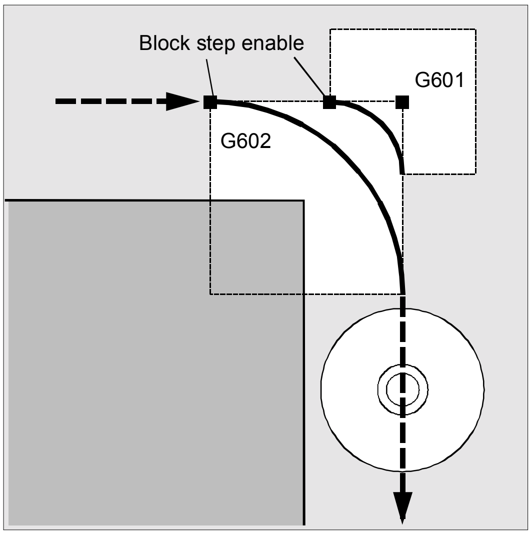
Function

The exact positioning stop functions are used to machine sharp outside corners or for finishing inside corners to size.

Sequence

Exact stop, G60, G9
G9 generates the exact stop in the current block.
G60 generates the exact stop in the current block and all subsequent blocks.
G60 is deactivated by the continuous-path mode functions G64 or G641. Positioning window.

G601/G602
The movement is decelerated and stopped briefly at the corner point. With the exact stop criteria G601 and G602, you determine how accurately the corner point is approached and when the change to the next block takes place.



Exact stops fine and coarse can be defined for each axis in machine data.

840D
NCU 571

840D
NCU 572
NCU 573

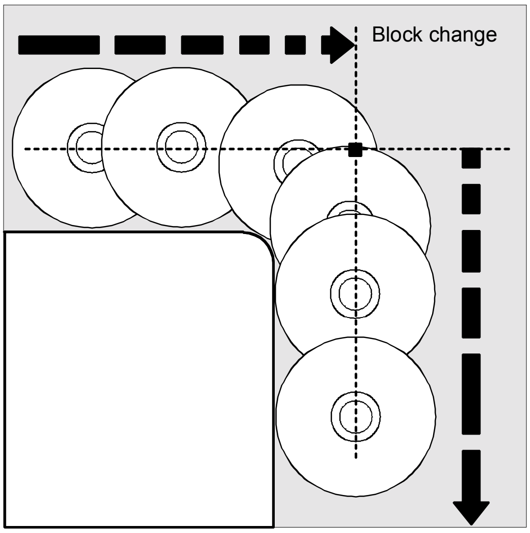
810D

840Di

Note: Set the exact stop limits no tighter than you require.
The tighter the limits, the longer it takes to position and approach the target position.

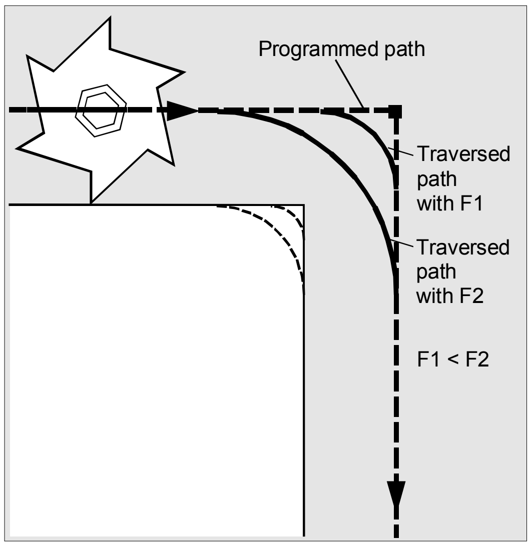
End of interpolation, G603

The block change is initiated when the control has calculated a set speed of zero for the axes involved. At this point, the actual value lags behind by a proportionate factor depending on the dynamic response of the axes and the path velocity. The workpiece corners can now be rounded.



Command output

In all three cases the following applies:
The auxiliary functions programmed in the NC block are enabled after the end of the movement.



G601, G602 and G603 are only effective if G60 or G9 are active.

Example:

```
N10 G601
…
N50 G1 G60 X… Y…
```

5.1 Exact stop, G60, G9, G601, G602, G603

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

In SW version 6 and higher, a machine data can be set for specific channels which determines that the default exact stop criteria, which deviate from the programmed criteria, will be applied automatically. These are given priority over the programmed criteria in some cases. Criteria for G0 and the other G commands in the 1st G code group can be stored separately.
See Description of Functions, Part1, B1.

840D NCU 571	840D NCU 572 NCU 573	810D	840Di

5.2 Continuous-path mode, G64, G641, G642, G643

Programming

```
G64
G641 AIDS=…
G641 ADISPOS=…
G642
G643
```

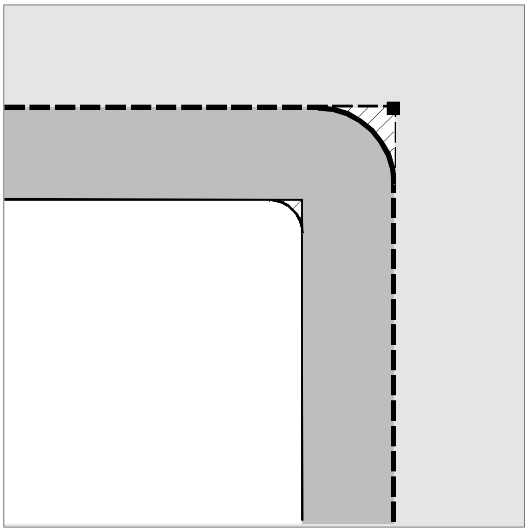
Explanation of the commands

`G64`	Continuous-path mode
`G641`	Continuous-path mode with programmable transition rounding
`G642`	Rounding with axial tolerance
`G643`	Block-internal corner rounding
`AIDS=`	Rounding clearance for path functions G1, G2, G3, …
`ADIPOSE=`	Approximate distance for rapid traverse G0

Function

In continuous-path mode, the contour is machined with a constant path velocity.

The uniform velocity also establishes better cutting conditions, improves the surface quality and reduces the machining time.

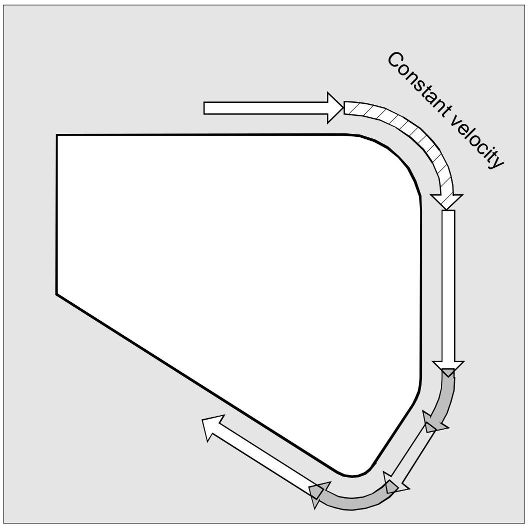


In continuous-path mode, travel does not take place exactly to the programmed contour transitions. You can generate sharp corners with G60 or G09. The continuous-path mode is interrupted by text outputs with "MSG" and blocks that implicitly trigger a preprocessor stop (e.g. access to certain machine status data ($A...)). The same applies to auxiliary function outputs; see Chapter 9 Special Functions.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

Continuous-path mode, G64
In continuous-path mode, the tool travels across tangential contour transitions with as constant a path velocity as possible (no deceleration at block boundaries). Look Ahead deceleration takes place before corners (G09) and blocks with exact stop (Look Ahead, see following pages).



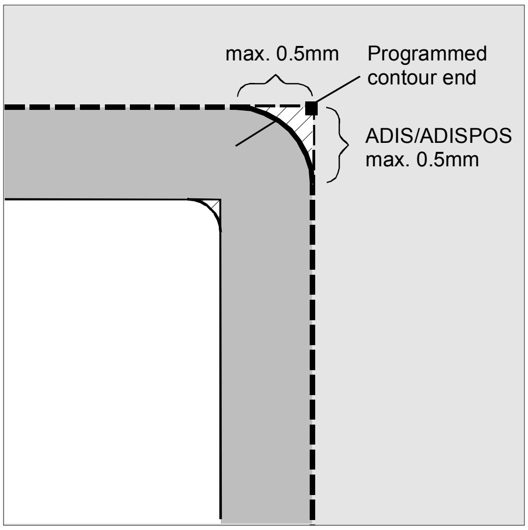
Corners are also traversed at a constant velocity. In order to minimize the contour error, the velocity is reduced according to an acceleration limit and an overload factor, see

References: /FB/ B1 Continuous-path mode

The overload factor can be set in machine data 32310 (see /FB/ B1, Continuous-path mode).
The extent of smoothing of the contour transitions depends on the feedrate and the overload factor. With G641 you can specify the desired rounding area explicitly (see following pages).

Rounding cannot and should not replace the functions for defined smoothing: RND, RNDM, ASPLINE, BSPLINE, CSPLINE.

Continuous-path mode with programmable transition rounding, G641
With G641, the control inserts transition elements at contour transitions. With `AIDS=…` or `ADIPOSE=…` you can specify the extent to which the corners are rounded. The effect of G641 is similar to RNDM, however it is not restricted to the axes of the working plane.



Example: `N10 G641 ADIS=0.5 G1 X… Y…`

The approximate positioning block can begin 0.5mm before the programmed end of block at the earliest and must be finished 0.5mm after the end of the block.
This setting remains modal.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

G641 also operates with Look Ahead speed control (see following pages). Corner rounding blocks with a high degree of curvature are approached at reduced velocity.

Additional notes

Rounding cannot be used as a substitute for smoothing (RND). The user should not make any assumptions with respect to the appearance of the contour within the rounding area. The type of rounding can depend on dynamic conditions, e.g. on the tool path velocity. Rounding on the contour is therefore only practical with small AIDS values.
RND must be used if a defined contour is to be followed at the corner in all circumstances.
ADIPOSE is used between G0 blocks. This enables the axis movement to be smoothed substantially and the traversing time to be reduced during positioning.
If ADIS/ADISPOS is not programmed, the value zero is valid and therefore the same traverse behavior as of G64. With shorter paths, the approximate distance is reduced automatically (up to max. 36%).

Continuous-path mode G64/G641 over more than one block

The following points should be noted, in order to prevent an undesired stop in the path motion (backing off):

- Auxiliary function outputs trigger a stop (exception: high-speed auxiliary functions and auxiliary functions during movements)
- Intermediate blocks which contain **only** comments, calculation blocks or subprogram calls do not affect the movement.

Extension of corner rounding

If FGROUP does not contain all the path axes, there is often a step change in the velocity at block boundaries for those axes excluded from FGROUP; the control limits this change in velocity to the permissible values set in MD 32300:

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

MAX_AX_ACCEL and MD 32310: _MAX_ACCEL_OVL_FACTOR by reducing the velocity during block changes. This braking operation can be avoided through the application of a rounding function which "smoothes" the specific positional interrelationship between the path axes.

Corner rounding with G641

You can activate a modal rounding action by programming G641 and specifying a rounding radius with ADIS (or ADISPOS in rapid traverse). Within this radius about the block change point, the control is free to ignore the path construct and replace it with a dynamically optimized path.

Disadvantage: Only **one** ADIS value is available for all axes.

Corner rounding with axial precision using G642

G642 activates corner rounding with modal axial tolerances. Corner rounding is not carried out inside a defined ADIS area but the **axial** tolerances defined with MD 33100: COMRESS_POS_TOL are met.

The mode of operation is otherwise identical to G641.

With G642, the rounding path is calculated from the **shortest** rounding path of all axes.

This value is taken into account when a **corner rounding block** is generated.

Block-internal corner rounding with G643 (SW 5.3 and higher)

The **maximum** deviations from the exact contour are defined by machine data MD 33100: COMRESS_POS_TOL[...] for each axis during corner rounding with G643.

No separate rounding block is generated for G643; instead, axis-specific block-internal rounding motions are inserted.

The rounding path can be **different** for each axis with G643.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Example for corner rounding with G643, see also:
References /PGA/ Programming Guide Advanced,
Chapter 5, Settable Path Reference, SPATH, UPATH

Extension of corner rounding with SW 6

The functional extensions described below refine the action of G642 and G643 and incorporate a new function, i.e.
corner rounding with contour tolerance. With corner rounding in conjunction with G642 and G643, the permissible deviations for each axis are normally specified.
With machine data
MD 20480: SMOOTHING_MODE
it is possible to configure rounding with G642 and G643 in such a way that a contour tolerance and an orientation tolerance can be specified instead of the axial tolerances. In this case, the tolerance for the **contour** and the **orientation** are set with two mutually independent setting data that can be programmed in the NC program; the settings can therefore be programmed differently for each block transition.
Setting data:
SD 42465: SMOOTH_CONTUR_TOL
This setting data defines the **maximum** rounding tolerance for the contour.

SD 42466: SMOOTH_ORI_TOL
This setting data defines the **maximum** rounding tolerance for the **tool orientation** (angular displacement).
This data is operative only if an **orientation transformation** is also active.
Large differences in the settings for the contour tolerance and tool orientation tolerance will only have an effect in conjunction with G643.

References: /FB/, B1, Continuous-path Mode, Exact Stop and Look Ahead

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

No rounding block/
no rounding motion

No corner rounding is performed in the following three situations:

1. A **halt is made** between the two blocks. This occurs when ...
   - The following block contains an auxiliary function output before the movement.
   - The following block does not contain a path movement.
   - An axis is traversed for the first time as a path axis for the following block and it was previously a positioning axis.
   - An axis is traversed for the first time as a positioning axis for the following block and it was previously a path axis.
   - The previous block traverses geo axes and the following block does not (this is no longer the case in SW 4 and higher).
   - Prior to thread cutting: The following block has G33 as a preparatory function and the previous block does not.
   - A changeover between BRISK and SOFT occurs.
   - Axes involved in the transformation are not completely assigned to the path motion (e.g. for oscillation, positioning axes).

2. The rounding block would **slow down** parts program execution. This occurs when ...
   - A rounding block is inserted between very short blocks. Since each block requires at least one interpolation cycle, the added intermediate block would double the machining time.
   - A block transition G64 (continuous-path mode without rounding) can be traversed without speed reduction. Rounding would increase the machining time.
     This means that the value of the permissible overload factor (MD 32310: MAX_ACCEL_OVL_FACTOR) would partly determine whether or not a block transition is rounded. The overload factor is taken into account only in conjunction with G641/G642.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

The overload factor is ignored in corner rounding with G643.

- In SW version 6 and higher, MD 20490: IGNORE_OVL_FACTOR_FOR_ADIS can be set to TRUE to ignore the overload factor in connection with G654 and G642 as well.

3. Rounding is **not parameterized**.
   This occurs with G641 when ...
   - ADIPOSE == 0 in G0 blocks (default!)
   - AIDS == 0 in non-G0 blocks (default!)
   - For transition from G0 to non-G0 or non-G0 to G0, the smaller value of ADIPOSE and AIDS applies.

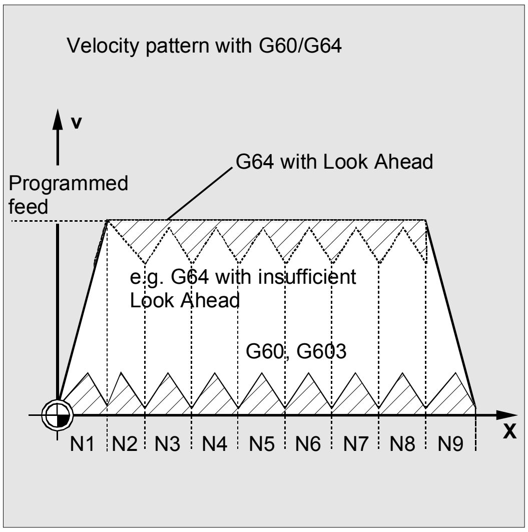
   With G642/G643, when all axis-specific tolerances equal zero.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Positioning axes
Positioning axes always traverse according to the exact stop principle, positioning window fine (as for G601). If an NC block has to wait for positioning axes, continuous-path mode is interrupted on the path axes.

Command outputs
Auxiliary functions which are enabled after the end of the movement or before the next movement interrupt continuous-path mode.

Look Ahead velocity control, Look Ahead
In continuous-path mode with G64 or G641, the control automatically detects the velocity control in advance for several NC blocks. This enables acceleration and deceleration across multiple blocks with almost tangential transitions.
Look Ahead is particularly suitable for the machining of movement sequences comprising short traverse paths with high path feedrates.
The number of NC blocks included in the Look Ahead calculation can be defined in machine data.



Look Ahead across more than one block is an option.

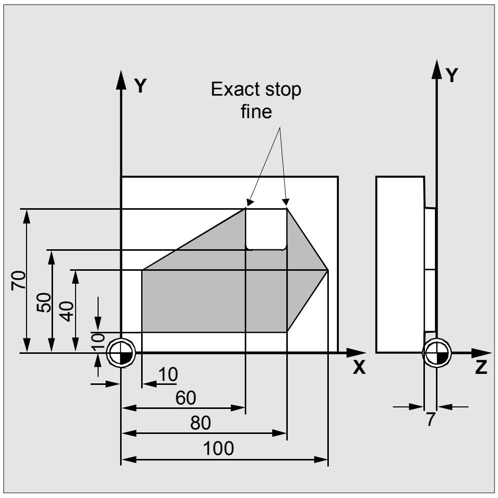
Continuous-path mode in rapid traverse G0
One of the functions G60/G9 or G64/G641 must also be specified for rapid traverse. Otherwise, the default in the machine data is used.

Setting MD 20490 IGNORE_OVL_FACTOR_FOR_ADIS results in block transitions being smoothed irrespective of the programmed overload factor.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

With this workpiece, the two outside corners at the groove are approached exactly. All other machining takes place in continuous-path mode.



N05	DIAMOF	Radius as dimension
N10	G17 T1 G41 G0 X10 Y10 Z2 S300 M3	Approach starting position, activate spindle, path compensation
N20	G1 Z-7 F8000	Tool infeed
N30	G641 ADIS=0.5	Contour transitions are smoothed
N40	Y40	
N50	X60 Y70 G60 G601	Approach exact position with exact stop fine
N60	Y50	
N70	X80	
N80	Y70	
N90	G641 ADIS=0.5 X100 Y40	Contour transitions are smoothed
N100	X80 Y 10	
N110	X10	
N120	G40 G0 X-20	Deactivate path compensation
N130	Z10 M30	Retract tool, end of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

5.3 Acceleration pattern, BRISK, SOFT, DRIVE

5.3.1 Acceleration modes

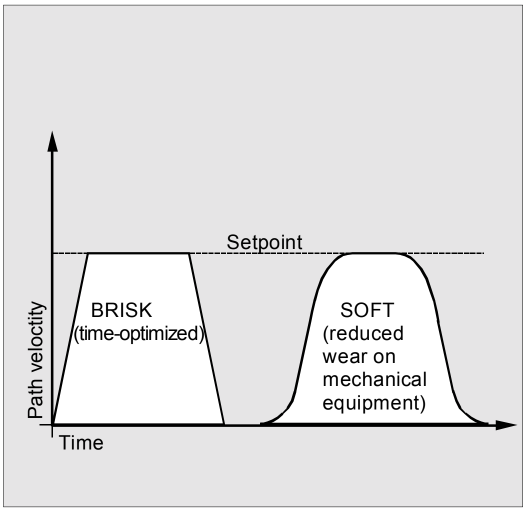
Explanation of the commands

`BRISK`	Brisk acceleration of path axes
`BRISKA(axis1,axis2,…)`	Switch on brisk axis acceleration for the programmed axes
`SOFT`	Soft acceleration of path axes
`SOFTA(axis1,axis2,…)`	Switch on soft axis acceleration for the programmed axes
`DRIVE`	Reduction of acceleration above a speed for path axes that can be set in `$MA_ACCEL_REDUCTION_SPEED_POINT` (only applicable for FM-NC)
`DRIVEA(axis1,axis2,…)`	Reduction of acceleration above a speed for programmed axes that can be set in `$MA_ACCEL_REDUCTION_SPEED_POINT` (only applicable for FM-NC)
`JERKA(axis1,axis2,…)`	The acceleration behavior set in machine data `$MA_POS_AND JOG_JERK_ENABLE` or `$MA_ACCEL_TYPE_DRIVE` is active for the programmed axes

Function

BRISK, BRISKA

The axis slides travel with maximum acceleration until the feedrate is reached. BRISK enables time-optimized machining, but with jumps in the acceleration curve.



SOFT, SOFTA

The axis slides travel with constant acceleration until the feedrate is reached.
Soft acceleration enables higher path accuracy and less wear and tear on the machine.

Example:
```
N10 G1 X… Y… F900 SOFT
N20 BRISKA(AX5,AX6)
```

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

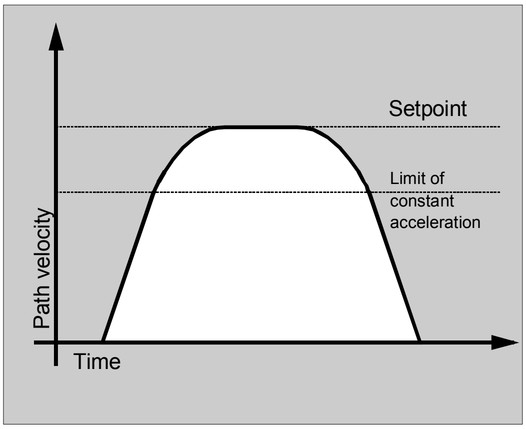
Additional notes

A change between BRISK and SOFT causes a stop at the block transition. The acceleration response for the path axes can be defined in machine data.

Function

DRIVE, DRIVEA

The axis slides traverse at the maximum acceleration rate up to the speed limit set in the machine data. The acceleration rate is then reduced according to machine data until the feedrate speed is reached. This function allows the acceleration characteristic to be optimally adapted to a specific motor characteristic, for example, for stepper motor applications.



Example:

```
N05 DRIVE
N10 G1 X… Y… F1000
N20 DRIVEA (AX4, AX6)
```

5.3.2 Influence of acceleration modes on following axes

Programming

`VELOLIMA[AX4]=75`	75 % of the maximum axial velocity stored in the machine data
`ACCLIMA[AX4]=50`	50 % of the maximum axial acceleration stored in the machine data

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Explanation of the commands

VELOLIMA[Ax]	Change to limit for maximum velocity for following axis
ACCLIMA[Ax]	Change to limit for maximum acceleration for following axis

Function

The coupled axes described in the Programming Guide Advanced Chapter 9 and Chapter 13.3, 13.4: Tangential correction, coupled-motion axes, master value coupling, and electronic gearbox have the property following axes/spindles are moved in dependence on one or more leading axes/spindles.

The commands for correction of limitation for the dynamic response of the following axis must be given from the parts program or from synchronous actions. The commands for correction of limitations of the following axis can already be given while axis coupling is active.

Additional notes

Details about the function are described in
References: /FB/, M3 Axis Coupling and ESR
/FB/, S3 Synchronous Spindles

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example 1

Electronic gear

Axis 4 is coupled to axis X via an electronic gear coupling. The acceleration capability of the following axis is limited to 70% of maximum acceleration. The maximum permissible velocity is limited to 50% of maximum velocity. After successful switch-on of coupling, the maximum permissible velocity is set to 100% again.

```
...
N120 ACCLIMA[AX4]=70              Reduced maximum acceleration
N130 VELOLIMA[AX4]=50             Reduced maximum velocity
...
N150 EGON(AX4, "FINE", X, 1, 2)   Switch-on of the EG coupling
...
N200 VELOLIMA[AX4]=100            Maximum velocity
```

Programming example 2

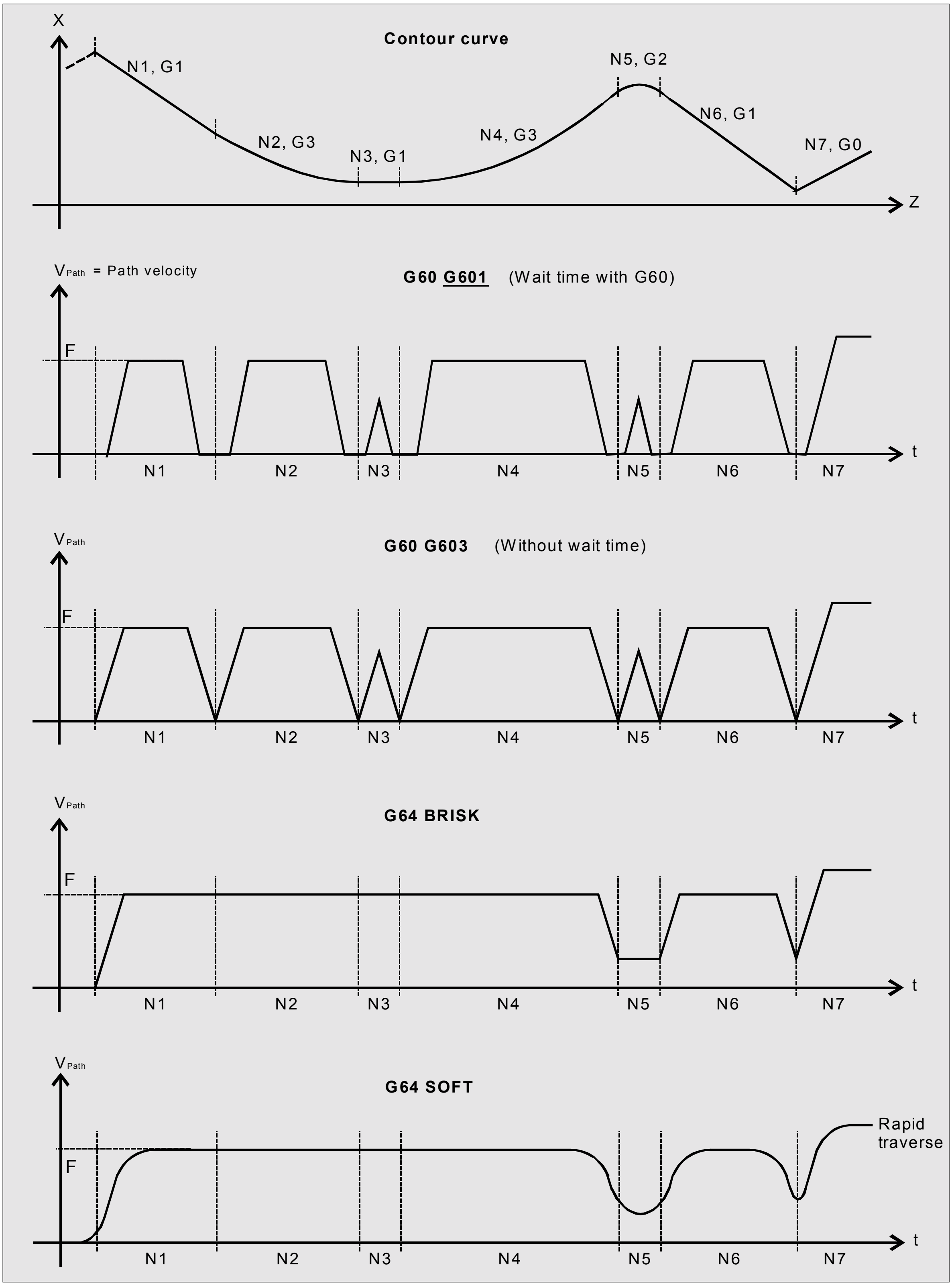
Master value coupling with influence by static synchronized action

Axis 4 is coupled to X by master value coupling. The acceleration response is limited to 80% by static synchronized action 2 from position 100.

```
...
N1220 IDS=2 WHENEVER $AA_IM[AX4] > 100      Synchronized action
            DO ACCLIMA[AX4]=80
N130 LEADON(AX4, X, 2)                      Master value coupling on
```

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

5.4 Overview of the various velocity controls



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

5.5 Path velocity smoothing

Function

The velocity control function utilizes the specified axial dynamic response. If an axis cannot reach the programmed feedrate, the path velocity is controlled according to the parameterized, axial limit values and the path-specific limits (i.e. velocity, acceleration and jerk). This action can give rise to frequent braking and acceleration on the path.
If, for example, during a machining operation at a high path velocity, the axis accelerates briefly, but brakes again almost immediately afterwards, the machining time will not be significantly reduced. Acceleration of this kind can, however, have undesirable effects if, for example, it results in machine resonance. A smoother path velocity profile can be achieved with the "Path velocity smoothing" function which allows special machine data and the character of the parts program to be taken into account.

Additional notes

References: /FB/, B1, "Path Velocity Smoothing (SW 5.3 and higher)"

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

5.6 Traversing with feedforward control, FFWON, FFWOF

Explanation of the commands

`FFWON`	Activate feedforward control
`FFWOF`	Deactivate feedforward control

Function

Feedforward control reduces the speed-dependent overtravel when contouring towards zero.
Traversing with feedforward control permits higher path accuracy and thus improved machining results.

Example:

```
N10 FFWON
N20 G1 X… Y… F900 SOFT
```

Additional notes

The type of feedforward control and which path axes feedforward traversing is to be applied to are determined via machine data.

Default: Velocity-dependent feedforward control
Optional: Acceleration-dependent feedforward control (not possible with FM-NC, 810D)

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

5.7 Programmable contour accuracy, CPRECON, CPRECOF

Explanation of the commands

`CPRECON`	Activate programmable contour accuracy
`CPRECOF`	Deactivate programmable contour accuracy

Function

In machining operations without feedforward control (FFWON), errors may occur on curved contours as the result of speed-related differences between setpoint and actual positions.
The programmable contour accuracy function CPRECON makes it possible to store a maximum permissible contour error in the NC program which must never be smoothed. The magnitude of the contour error is specified with setting data `$SC_CONTPREC`.
On the basis of this data and the servo gain factor (speed / following error ratio) of the path axes concerned, the control calculates the maximum path speed at which the contour error produced by the overtravel does not exceed the minimum value stored in the setting data.
The Look Ahead function allows the entire path to be traversed with the programmed contour accuracy.

Example:

```
N10 X0 Y0 G0
N20 CPRECON                 ;Activate contour accuracy
N30 F10000 G1 G64 X100      ;Machine contour at 10 m/min in
                            continuous-path mode
N40 G3 Y20 J10              ;Automatic feed limitation in circle block
N50 X0                      ;Feed without limitation to 10 m/min
```

Additional notes

A minimum velocity can be defined in the setting data `$SC_MINFEED`; the feed must not be less than this value.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

5.8 Dwell time, G4

Programming

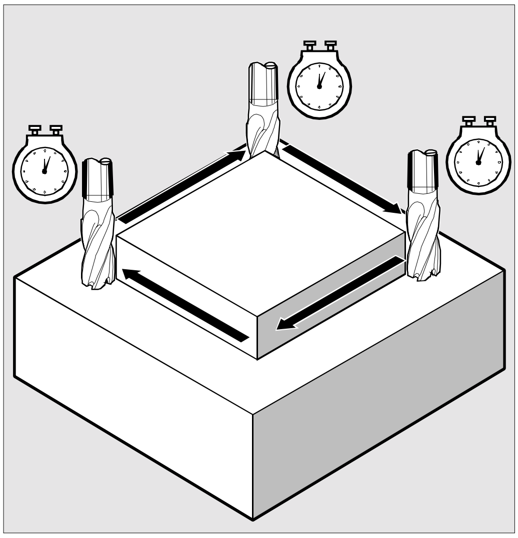
```
G4 F…
G4 S…
```
(Programmed in a separate NC block)

Explanation of the commands

G4	Activate dwell time
F…	Time specified in seconds
S…	Time specified in revolutions of the master spindle

Function

You can use G4 to interrupt workpiece machining between two NC blocks for the programmed length of time, e.g. for relief cutting.



Sequence

Example:

```
N10 G1 F200 Z-5 S300 M3
               ;Feed F, spindle speed S
N20 G4 F3      ;Dwell time 3s
N30 X40 Y10
N40 G4 S30     ;Dwell for 30 revolutions of
               spindle, with S=300rpm and 100%
               speed override, equals: t=0.1min
N40 X...       ;Feed and spindle speed still active
```

The words with F... and S... are used for time specifications in the block with G4 only.
Any previously programmed feed F and spindle speed S remain valid.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

5.9 Program sequence: Internal preprocessor stop

Function

The control generates an internal preprocessor stop on access to machine status data ($A…).
If a command which generates an implicit preprocessor stop is read in the following block, the following block is not executed until all previously prepared and stored blocks have been processed completely. The preceding block is halted in an exact stop (as for G9).

Example:

```
N40 POSA[X]=100
N50 IF $AA_IM[X]==R100 GOTOF LABEL1     ; Access to machine status data ($A), the
                                        control generates an internal
                                        preprocessor stop
N60 G0 Y100
N70 WAITP(X)
N80 LABEL1:
```

Machining stops in block N50.

■

5.9 Program sequence: Internal preprocessor stop

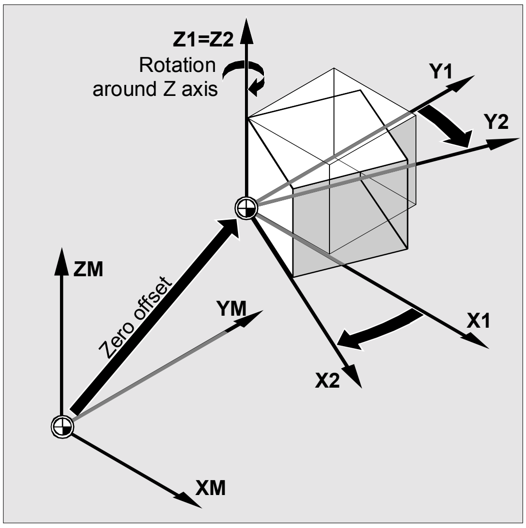
840D NCU 571	840D NCU 572 NCU 573	810D	840Di

Frames

840 D
NCU 572
NCU 573

6.1 General

What is a frame?
Frame is the conventional term for a geometrical expression that describes an arithmetic rule, such as translation or rotation.



Frames are used to describe the position of a destination coordinate system by specifying coordinates or angles starting from the current workpiece coordinate system.

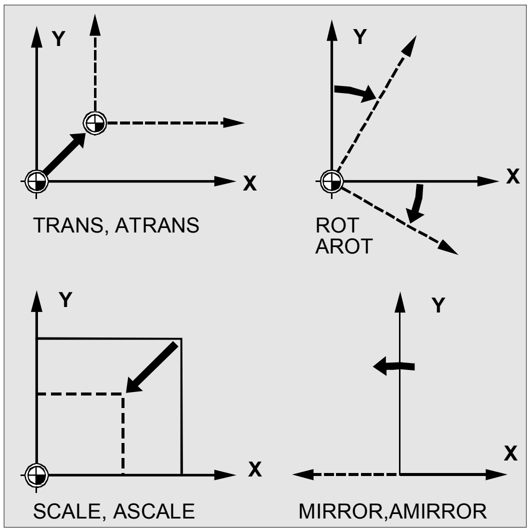
Possible frames
- Basic frame (basic offset)
- Settable frames (G54...G599)
- Programmable frames
  References: /PG/, Programming Guide Advanced

Machine manufacturer (MH6.1)

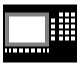
Settable frames: See machine manufacturer's specifications

Frame components
A frame can consist of the following arithmetic rules:
- Translation, TRANS, ATRANS
- Rotation, ROT, AROT
- Scale, SCALE, ASCALE
- Mirroring, MIRROR, AMIRROR

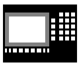
Milling:



The above frame instructions are programmed in separate NC blocks and executed in the programmed order.

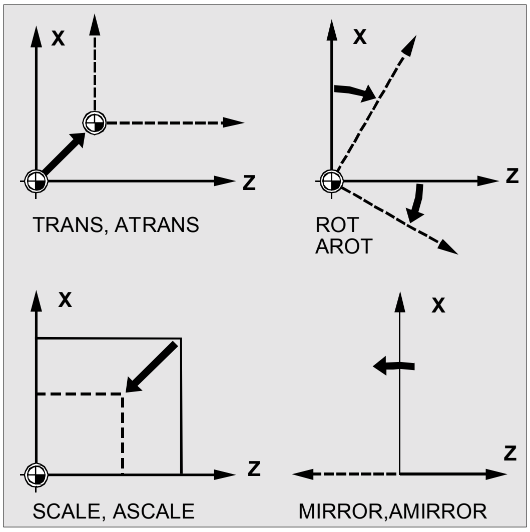


840D
NCU 572
NCU 573



840Di

Turning:



6.2 Frame instructions

Basic frame (basic offset)
The basic frame describes the coordinate transformation from the Basic Coordinate System (BCS) to the Basic Zero System (BZS) and has the same effect as for settable frames.

Settable instructions
Settable instructions are the zero offsets which can be called from any NC program with the commands G54 to G599. The offset values are predefined by the user and stored in the zero offset memory on the control.
This is used to define the Workpiece Coordinate System (WCS).

Programmable instructions
Programmable instructions (TRANS, ROT, ...) are valid in the current NC program and refer to the settable instructions. The programmable frame is used to define the Workpiece Coordinate System (WCS).

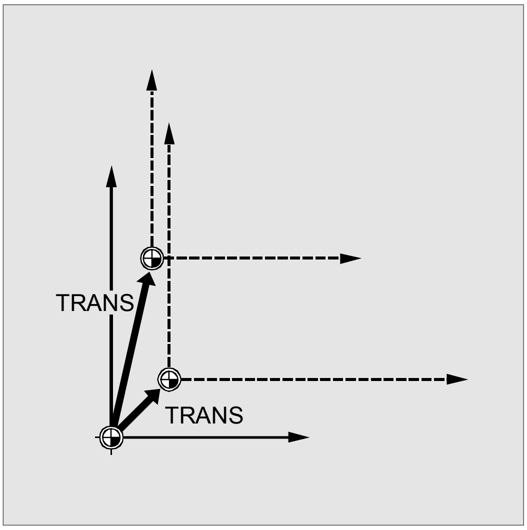
840 D
NCU 572
NCU 573

Substituting instructions

TRANS, ROT, SCALE and MIRROR are substituting instructions.

*This means that each of these instructions cancels **all** other previously programmed frame instructions.*

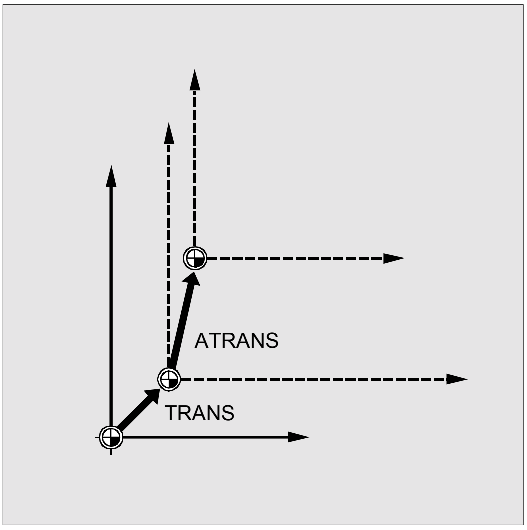
The last called settable zero offset G54 to G599 is used as the reference.



Additive instructions

ATRANS, AROT, ASCALE and AMIRROR are additive instructions.

The currently set zero point or the last workpiece zero to be programmed with frame instructions is used as the reference. The above instructions are added to existing frames.



Note: Additive instructions are frequently used in subprograms. The basic functions defined in the main program are not lost after the end of the subprogram if the subprogram has been programmed with the SAVE attribute.

References

/PGA/ Programming Guide Advanced, Section "Subroutines, Macros"

840D
NCU 572
NCU 573

840Di

6.3 Programmable zero offset

6.3.1 TRANS, ATRANS

Programming

TRANS X… Y… Z… (programmed in a separate NC block)
ATRANS X… Y… Z… (programmed in a separate NC block)

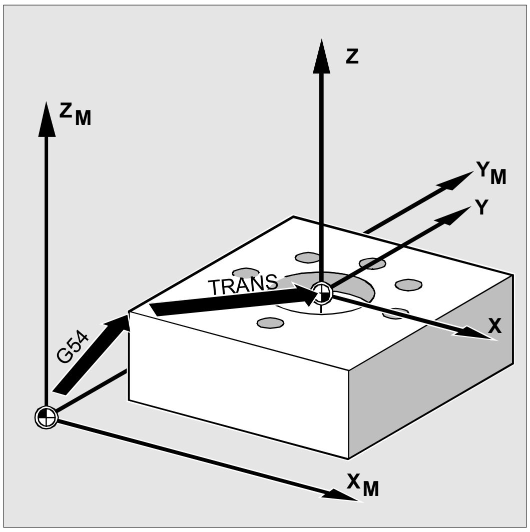
Explanation of the commands and parameters

TRANS	Absolute zero offset, with reference to the currently valid workpiece zero set with G54 to G599
ATRANS	as TRANS, but with additive zero offset
X Y Z	Offset value in the direction of the specified geometry axis

Function

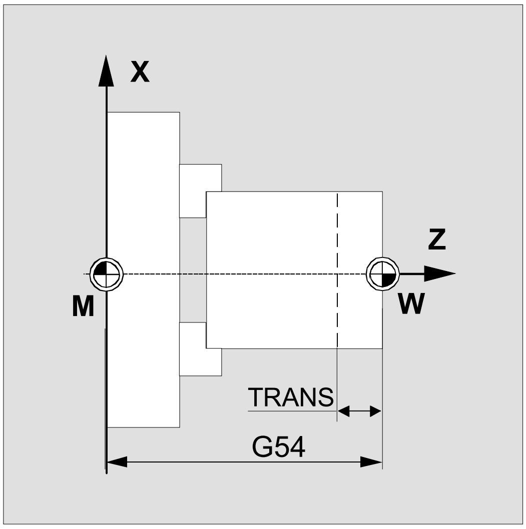
TRANS/ATRANS can be used to program translations for all path and positioning axes in the direction of the specified axis. This allows you to work with different zero points, for example when performing recurring machining processes at different workpiece positions.

Milling:



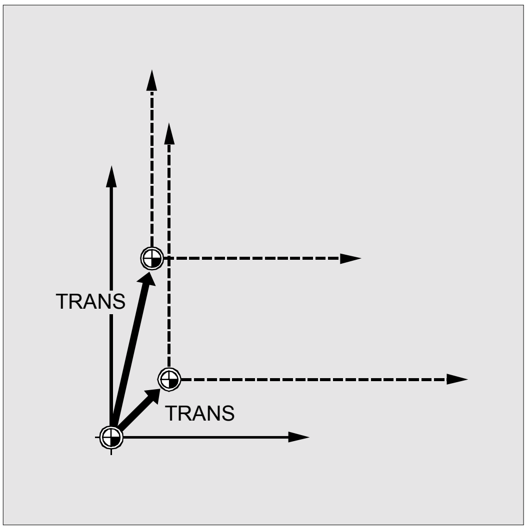
840 D
NCU 572
NCU 573

Turning:



Sequence

Substituting instruction, TRANS X Y Z
Translation through the offset values programmed in the specified axis directions (path, synchronized axes and positioning axes).
The last specified settable zero offset (G54 to G599) is used as a reference.



The TRANS command cancels all frame components of the previously activated programmable frame.

You can use ATRANS to program a translation which is to be added to existing frames.

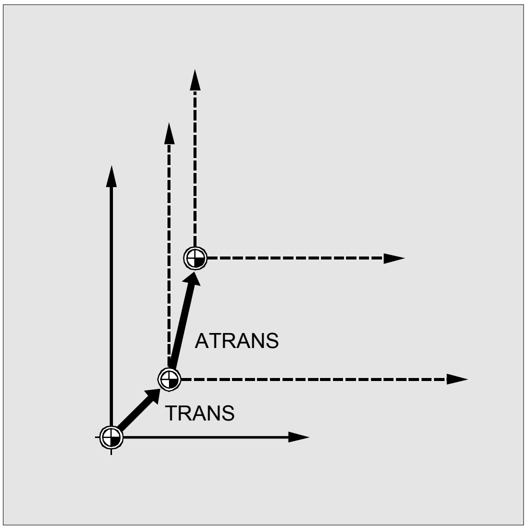
840D
NCU 572
NCU 573

840Di

Additive instruction, ATRANS X Y Z
Translation through the offset values programmed in the specified axis directions.

The currently set or last programmed zero point is used as the reference.

Deactivate programmable zero offset
For all axes:
TRANS (without axis parameter)



Here, all previously programmed frames are canceled. The settable zero offset remains programmed.

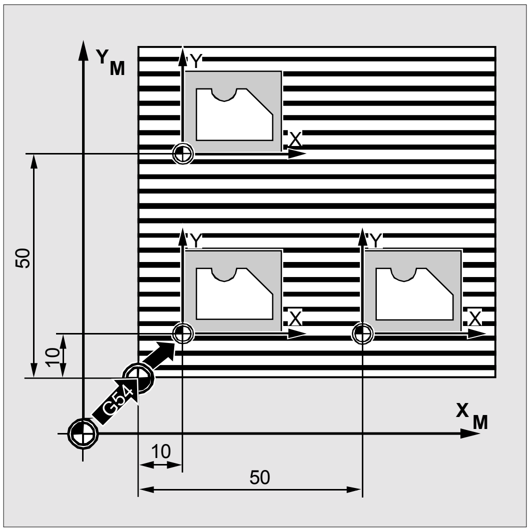
Programming example

With this workpiece, the illustrated shapes recur several times in the same program.

The machining sequence for this shape is stored in a subprogram.

You use the translation to set only those workpiece zeroes and then call up the subprogram.

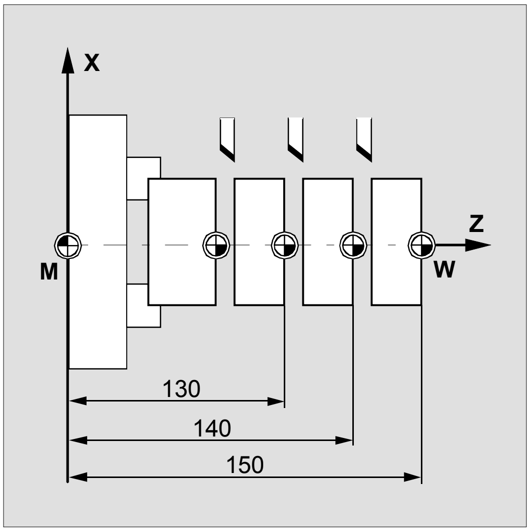
Milling:



N10	G1 G54	Working plane X/Y, workpiece zero
N20	G0 X0 Y0 Z2	Approach starting point
N30	TRANS X10 Y10	Absolute offset
N40	L10	Subprogram call
N50	TRANS X50 Y10	Absolute offset
N60	L10	Subprogram call
N70	M30	End of program

840 D
NCU 572
NCU 573

Turning:



N..	...	
N10	TRANS X0 Z150	Absolute offset
N15	L20	Subprogram call
N20	TRANS X0 Z140 (or ATRANS Z-10)	Absolute offset
N25	L20	Subprogram call
N30	TRANS X0 Z130 (or ATRANS Z-10)	Absolute offset
N35	L20	Subprogram call
N..	...	

840D
NCU 572
NCU 573

840Di

6.3.2 G58, G59: Axial programmable ZO (SW 5 and higher)

Programming

`G58` X... Y... Z... A... (programmed in a separate NC block)
`G59` X... Y... Z... A... (programmed in a separate NC block)

Explanation of the commands and parameters

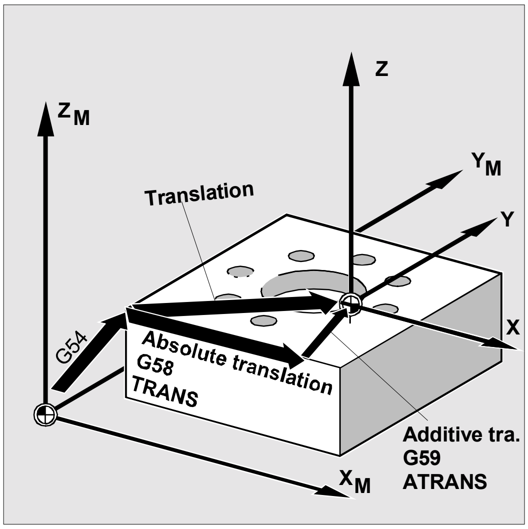
`G58`	Replaces the absolute translation component of the programmable zero offset for the specified axis, but the programmed additive offset remains valid, (in relation to the workpiece zero set with G54 to G599)
`G59`	Replaces the absolute translation component of the programmable zero offset for the specified axis, but the programmed absolute offset remains valid
`X Y Z`	Offset value in the direction of the specified geometry axis

Function

G58 and G59 allow translation components of the programmable zero offset (frame) to be replaced for specific axes. The translation function comprises the

- absolute component (G58, coarse offset)
- additive component (G59, fine offset)

These functions can only be used when the fine offset is configured.
If G58 or G59 is used without a configured fine offset, alarm "18312 channel %1 block %2 frame: Fine offset not configured" is output.



Machine manufacturer (MH6.2)

The fine offset must be configured via MD for this function.

Note MD24000:FRAME_ADD_COMPONENTS=1, or else an alarm is generated in response to G58, G59.

840 D
NCU 572
NCU 573

The absolute translation component is modified by the following commands:

- TRANS,
- G58
- CTRANS
- CFINE
- $P_PFRAME[X,TR]

The additive translation component is modified by the following commands:

- ATRANS,
- G59
- CTRANS
- CFINE
- $P_PFRAME[X,FI]

The table below describes the effect of various program commands on the absolute and additive offsets.

Effect of the additive/absolute offset:

Command	Rough or absolute offset	Fine or additive offset	Comment
TRANS X10	10	Unchanged	Absolute offset for X
G58 X10	10	Unchanged	Overwrites absolute offset for X
$P_PFRAME[X,TR] = 10	10	Unchanged	Progr. offset in X
ATRANS X10	Unchanged	Fine (old) + 10	Additive offset for X
G59 X10	Unchanged	10	Overwrites additive offset for X
$P_PFRAME[X,FI] = 10	Unchanged	10	Progr. fine offset in X
CTRANS(X,10)	10	0	offset for X
CTRANS()	0	0	Deselection of offset (including fine offset component)
CFINE(X,10)	0	10	Fine offset in X

840D
NCU 572
NCU 573

840Di

Programming example

```
N...
N50 TRANS X10 Y10 Z10     ; absolute translation component X10 Y10 Z10
N60 ATRANS X5 Y5          ; additive translation component X5 Y5
                                = total offset X15 Y15 Z10
N70 G58 X20               ; absolute translation component X20 + addit. X5 Y5
                                = total offset X25 Y15 Z10
N80 G59 X10 Y10           ; additive translation component X10 Y10 + absolute
                          X20 Y 10
                                = total offset X30 Y20 Z10
N...
```

840 D
NCU 572
NCU 573

6.4 Programmable rotation, ROT, AROT

Programming

```
ROT X… Y… Z…
ROT RPL=…

AROTX… Y… Z…
AROT RPL=…
```

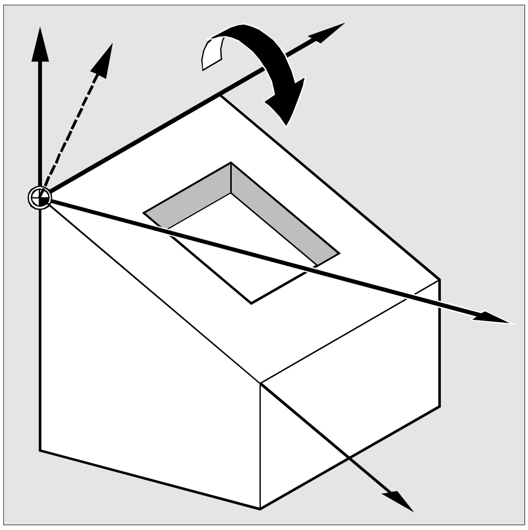
Each instruction must be programmed in a separate NC block.

Explanation of the commands and parameters

ROT	Absolute rotation with reference to the currently valid workpiece zero set with G54 to G599
AROT	Additive rotation with reference to the currently valid set or programmed zero point
X Y Z	Rotation in space: geometry axes around which the rotation takes place
RPL	Rotation in the plane: Angle through which the coordinate system is rotated (plane set with G17-G19)

Function

ROT/AROT can be used to rotate the workpiece coordinate system around each of the geometry axes X, Y, Z or through an angle RPL in the selected working plane G17 to G19 (or around the perpendicular infeed axis).

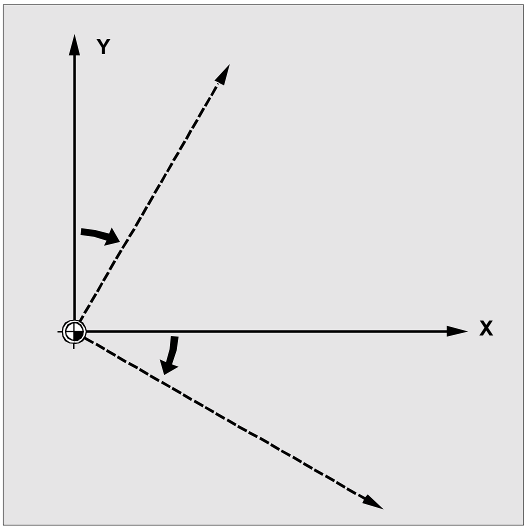


This allows inclined surfaces or several workpiece sides to be machined in one setting.

840D
NCU 572
NCU 573

840Di

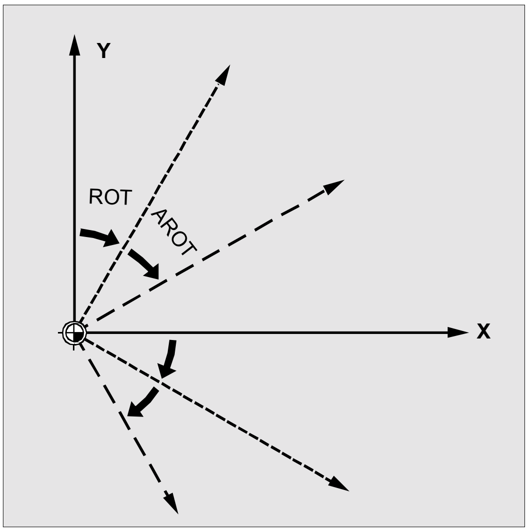
Sequence: Rotation in space



Substituting instruction, ROT X Y Z
The coordinate system is rotated through the programmed angle around the specified axes.
The center of rotation is the last specified settable zero offset (G54 to G599).

The ROT command cancels all frame components of the previously activated programmable frame.

A new rotation based on existing frames is programmed with AROT.



Additive instruction, AROT X Y Z
Rotation through the angle values programmed in the axis direction parameters.

The center of rotation is the currently set or last programmed zero point.

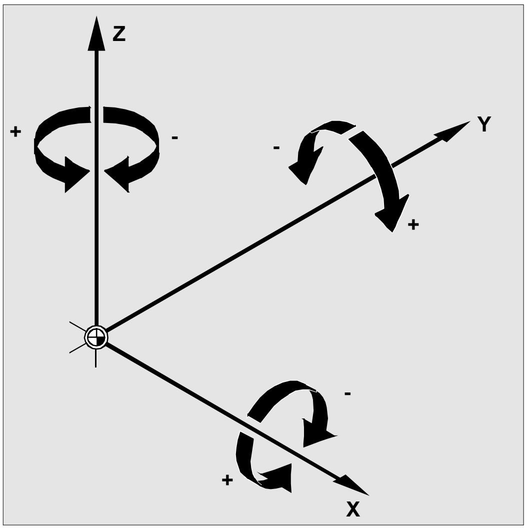
Note

For both instructions, please note the order and direction of rotation in which the rotations are performed (see next page)!

840 D
NCU 572
NCU 573

Direction of rotation

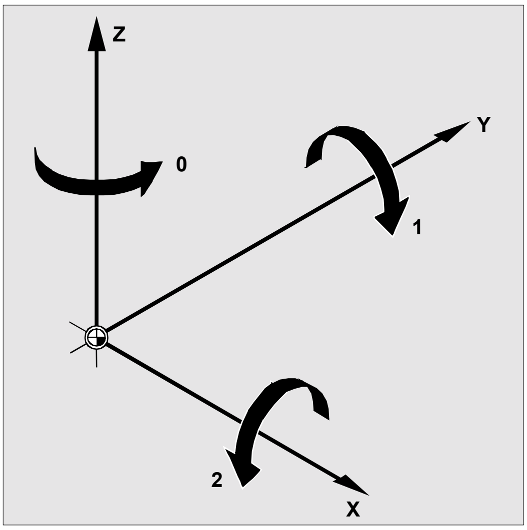
The following is defined as the positive direction of rotation:
The view in the direction of the positive coordinate axis and clockwise rotation.



Order of rotation

You can rotate up to three geometry axes simultaneously in one NC block.

The order of the RPY notation (= Roll, Pitch, Yaw) or Euler angle through which the rotations are performed can be defined in machine data.



MD 10600: FRAME_ANGLE_INPUT_MODE =
- 1: RPY notation
- 2: Euler angle

RPY notation is the default setting. After that, the sequence Z, Y, Z of the rotation is defined as follows:
1. Rotation around the 3rd geometry axis (Z)
2. Rotation around the 2nd geometry axis (Y)
3. Rotation around the 1st geometry axis (X)

This order applies if the geometry axes are programmed in a **single** block. It also applies irrespective of the input sequence.
If only two axes are to be rotated, the parameter for the 3rd axis (value zero) can be omitted.

Value range with RPY angle

The angles are **only** defined ambiguously in the following value ranges:
Rotation around 1st geometry axis: $-180° \leq X \leq +180°$
Rotation around 2nd geometry axis: $-90° < Y < +90°$
Rotation around 3rd geometry axis: $-180° \leq Z \leq +180°$

840D
NCU 572
NCU 573

840Di

All possible rotations can be represented with this value range. Values outside the range are normalized by the control into the above range during writing and reading. This value range applies to all frame variables.

Examples of reading back in RPY
$P_UIFR[1] = CROT(X, 10, Y, 90, Z, 40)
returns on reading back
$P_UIFR[1] = CROT(X, 0, Y, 90, Z, 30)

$P_UIFR[1] = CROT(X, 190, Y, 0, Z, -200)
returns on reading back
$P_UIFR[1] = CROT(X, -170, Y, 0, Z, 160)
On writing and reading frame rotation components, the limits of the value range must be observed so the same results are achieved on writing and reading, or on repeated writing.

Value range with Euler angle
The angles are **only** defined ambiguously in the following value ranges:
Rotation around 1st geometry axis: $0° < X < +180°$
Rotation around 2nd geometry axis: $–180° \leq Y \leq +180°$
Rotation around 3rd geometry axis: $–180° \leq Z \leq +180°$
All possible rotations can be represented with this value range. Values outside the range are normalized by the control into the above range. This value range applies to all frame variables.

To ensure the angles written are read back unambiguously, it is necessary to observe the defined values ranges.

If you want to define the order of the rotations individually, program the desired rotation successively for each axis with AROT.

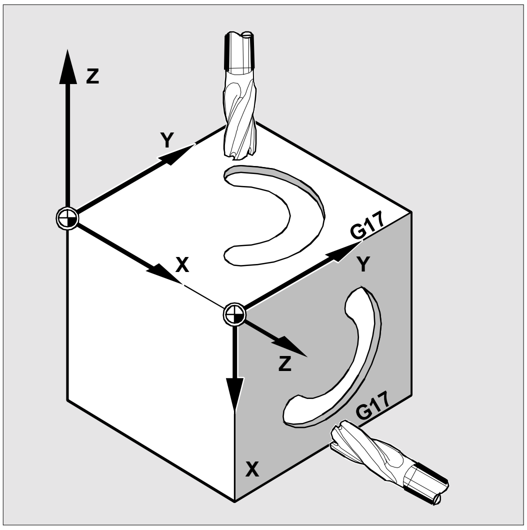
References

/FB1/ Function Description of the Basic Machine, Chapter "Frames"

840 D
NCU 572
NCU 573

The working plane also rotates
The working plane defined with G17, G18 or G19 rotates with the spatial rotation.

Example:
Working plane G17 X/Y, the workpiece coordinate system is positioned on the top surface of the workpiece. Translation and rotation is used to move the coordinate system to one of the side surfaces. Working plane G17 also rotates.

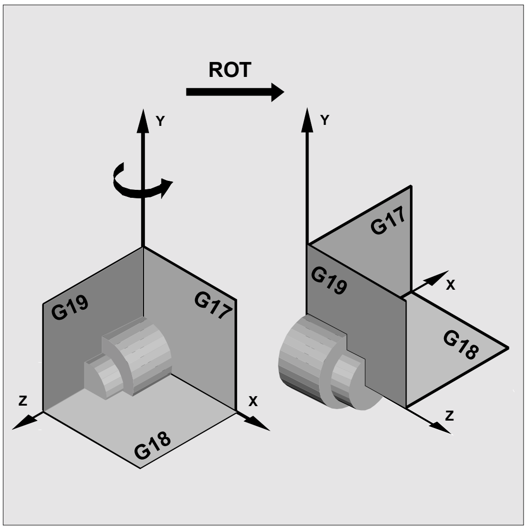


This feature can be used to program plane destination positions in X/Y coordinates and the infeed in the Z direction.

Precondition:
The tool must be positioned perpendicular to the working plane. The positive direction of the infeed axis points in the direction of the toolholder. Specifying CUT2DF activates the tool radius compensation in the rotated plane. For more information please refer to Section "2 1/2 D Tool Compensation, CUT2D CUT2DF".

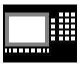
Sequence: Rotation in the plane

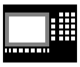
The coordinate system is rotated in the plane selected with G17 to G19.



Substituting instruction, ROT RPL
Additive instruction, AROT RPL
The coordinate system is rotated through the angle programmed with RPL= in the current plane.

See "Rotation in space" for more information.


840D
NCU 572
NCU 573


840Di

Change of plane

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.

Deactivate rotation

For all axes:

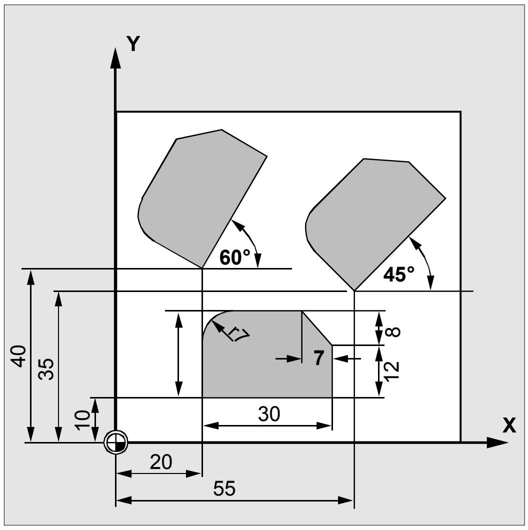
ROT (without axis parameter)

In both cases, all frame components of the previously programmed frame are reset.

Programming example: Rotation in the plane

With this workpiece, the illustrated shapes recur several times in the same program.

Rotations have to be performed in addition to the translation, because the shapes are not arranged parallel to the axes.

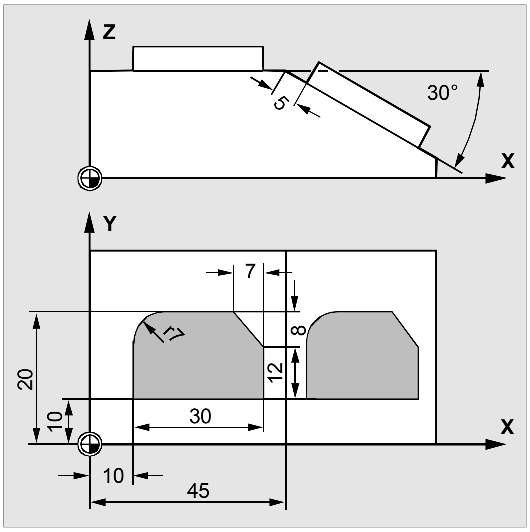


N10	G17 G54	Working plane X/Y, workpiece zero
N20	TRANS X20 Y10	Absolute offset
N30	L10	Subprogram call
N40	TRANS X55 Y35	Absolute offset
N50	AROT RPL=45	Rotation of the coordinate system through 45°
N60	L10	Subprogram call
N70	TRANS X20 Y40	Absolute offset (cancels all previous offsets)
N80	AROT RPL=60	Additive rotation through 60°
N90	L10	Subprogram call
N100	G0 X100 Y100	Retraction
N110	M30	End of program

840 D
NCU 572
NCU 573

Programming example:
3D rotation

In this example, paraxial and inclined workpiece surfaces are to be machined in one setting.
Precondition: The tool must be aligned perpendicular to the inclined surface in the rotated Z direction.



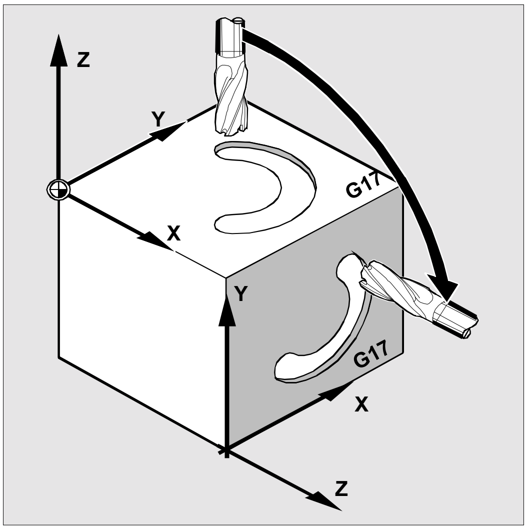
N10	G17 G54	Working plane X/Y, workpiece zero
N20	TRANS X10 Y10	Absolute offset
N30	L10	Subprogram call
N40	ATRANS X35	Additive offset
N50	AROT Y30	Rotation through the Y axis
N60	ATRANS X5	Additive offset
N70	L10	Subprogram call
N80	G0 X300 Y100 M30	Retraction, end of program

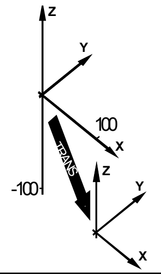
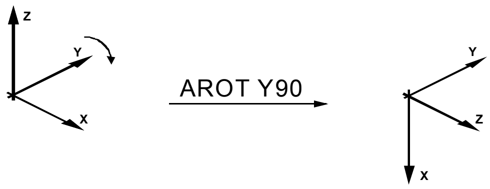
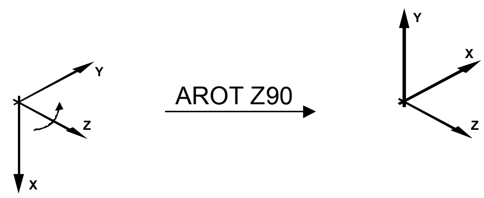
840D
NCU 572
NCU 573

840Di

Programming example: Multi-face machining

In this example, identical shapes on two perpendicular workpiece surfaces are machined by using subprograms.
The setup of the infeed direction, working plane and zero point in the new coordinate system on the right-hand workpiece surface matches that of the top surface.
The conditions required for subprogram execution apply as before: working plane G17, coordinate plane X/Y, infeed direction Z.



N10	G17 G54	Working plane X/Y, workpiece zero
N20	L10	Subprogram call
N30	TRANS X100 Z-100	Absolute offset 
N40	AROT Y90	Rotation of the coordinate system through Y 
N50	AROT Z90	Rotation of the coordinate system through Z 
N60	L10	Subprogram call
N70	G0 X300 Y100 M30	Retraction, end of program

840 D
NCU 572
NCU 573

6.5 Programmable frame rotations with solid angles, ROTS, AROTS and CROTS

Programming

`ROTS X... Y...`
`AROTS X... Y...`
`CROTS X... Y...`
	When solid angles X and Y are programmed, the new X axis is located in the old Z-X plane (SW 5.3 and higher).

`ROTS Z... X...`
`AROTS Z... X...`
`CROTS Z... X...`
	When solid angles Z and X are programmed, the new Z axis is located in the old Y-Z plane (SW 5.3 and higher).

`ROTS Y... Z...`
`AROTS Y... Z...`
`CROTS Y... Z...`
	When solid angles Y and Z are programmed, the new X axis is located in the old X-Y plane (SW 5.3 and higher).

Explanation of the commands and parameters

ROTS	Frame rotations with solid angles for spatial orientation of a plane absolute, referred to the currently valid frame with set workpiece zero for G54 to G599.
AROTS	Frame rotations with solid angles for spatial orientation of a plane additive, referred to the currently valid frame with set or programmed zero point.
CROTS	Frame rotations with solid angles for spatial orientation of a plane, referred to the valid frame in the data management with rotation in the specified axes.
X Y Z	A maximum of two solid angles may be specified
RPL	Rotation in the plane: Angle through which the coordinate system is rotated (plane set with G17-G19)

Function

Spatial orientations can be defined via frame rotations with solid angles ROTS, AROTS, CROTS.
Programming commands ROTS and AROTS behave analogously to ROT and AROT.

840D
NCU 572
NCU 573

840Di

6.6 Programmable scale factor, SCALE, ASCALE

Programming

SCALE X… Y… Z… (programmed in a separate NC block)
ASCALE X… Y… Z… (programmed in a separate NC block)

Explanation of the commands and parameters

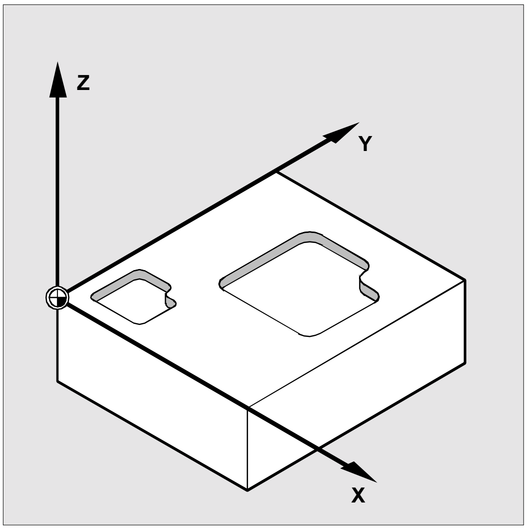
SCALE	Absolute enlargement/reduction with reference to the currently valid coordinate system set with G54 to G599
ASCALE	Additive enlargement/reduction with reference to the currently valid set or programmed coordinate system
X Y Z	Scale factor in the direction of the specified geometry axis

Function

SCALE/ASCALE enables you to program scaling factors in the direction of the axis specified for all path, synchronous and positioning axes.
This enables the size of a shape to be changed. You can thus program similar geometrical shapes in different sizes.

Sequence

Substituting instruction, SCALE X Y Z
You can specify an individual scale factor for each axes, by which the shape is to be reduced or enlarged. The scale refers to the workpiece coordinate system set with G54 to G57.



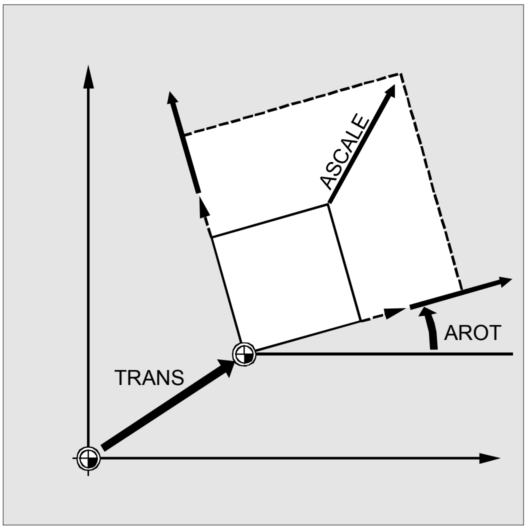
The SCALE command cancels all frame components of the previously activated programmable frame.

840 D
NCU 572
NCU 573

Additive instruction, ASCALE X Y Z

You can program scale changes which are to be added to existing frames by using the ASCALE command.
In this case, the last valid scale factor is multiplied by the new one.

The currently set or last programmed coordinate system is used as the reference for the scale change.



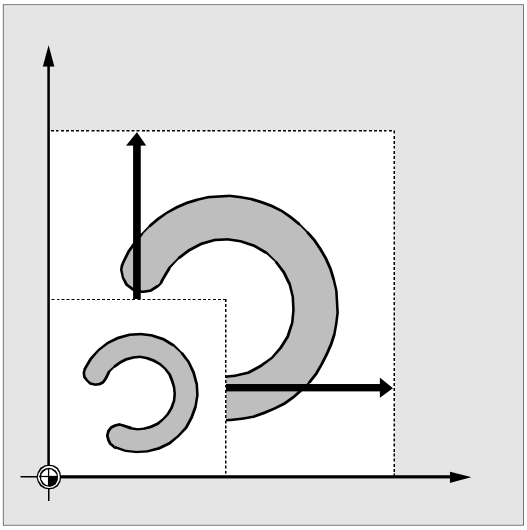
Deactivate scaling factor

For all axes:
SCALE (without axis parameter)

All frame components of the previously programmed frame are reset.

Additional notes

If you program an offset with ATRANS after SCALE, the offset values are also scaled.



Please take great care when using different scale factors! Example: Circular interpolations can only be scaled using identical factors.

You can, however, use different scale factors to program distorted circles, for example.

840D
NCU 572
NCU 573

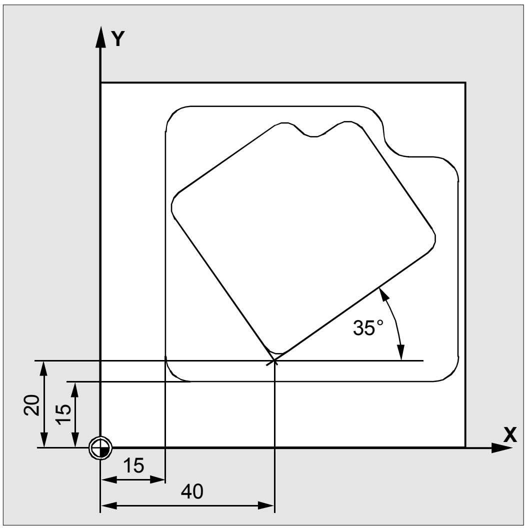
840Di

Programming example

With this workpiece, the two pockets occur twice, but in different sizes and at different angles to each other.

The machining sequence is stored in a subprogram.

Use translation and rotation to set each of the workpiece zeroes, reduce the contour with a scale and then call the subprogram up again.



N10	G17 G54	Working plane X/Y, workpiece zero
N20	TRANS X15 Y15	Absolute offset
N30	L10	Machine large pocket
N40	TRANS X40 Y20	Absolute offset
N50	AROT RPL=35	Rotation in the plane through 35°
N60	ASCALE X0.7 Y0.7	Scale factor for the small pocket
N70	L10	Machine small pocket
N80	G0 X300 Y100 M30	Retraction, end of program

840 D
NCU 572
NCU 573

6.7 Programmable mirroring, MIRROR, AMIRROR

Programming

`MIRROR X0 Y0 Z0` (programmed in a separate NC block)
`AMIRROR X0 Y0 Z0` (programmed in a separate NC block)

Explanation of the commands and parameters

`MIRROR`	Absolute mirror image with reference to the currently valid coordinate system set with G54 to G599
`AMIRROR`	Additive mirror image with reference to the currently valid set or programmed coordinate system
`X Y Z`	Geometry axis whose direction is to be changed. The value specified here can be chosen freely, e.g. X0 Y0 Z0

Function

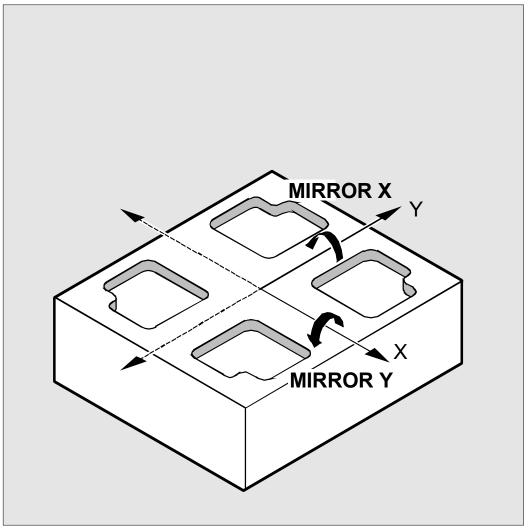
MIRROR/AMIRROR can be used to mirror workpiece shapes on coordinate axes. All traversing movements which are programmed after the mirror call, e.g. in the subprogram, are executed in the mirror image.

Sequence

Substituting instruction, MIRROR X Y Z
The mirror is programmed by means of an axial change of direction in the selected working plane.

Example: working plane G17 X/Y
The mirror (on the Y axis) requires a change of direction on X and is subsequently programmed with MIRROR X0.

The contour is then mirrored on the opposite side of the mirror axis Y.



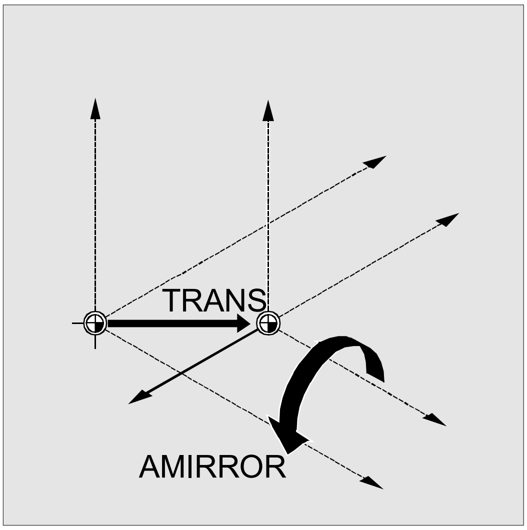
840D
NCU 572
NCU 573

840Di

The mirror image refers to the coordinate axes set with G54 to G57.

The MIRROR command cancels all previously set programmable frames.

Additive instruction, AMIRROR X Y Z
A mirror image which is to be added to an existing transformation is programmed with AMIRROR.



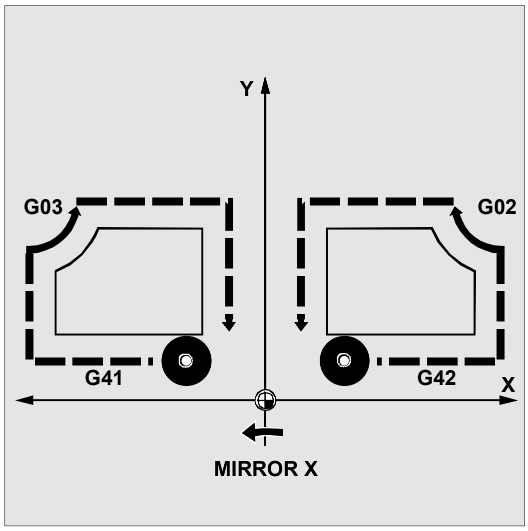
The currently set or last programmed coordinate system is used as the reference.

Deactivate mirroring
For all axes:
MIRROR (without axis parameter)

All frame components of the previously programmed frame are reset.

Additional notes

The mirror command causes the control to automatically change the path compensation commands (G41/G42 or G42/G41) according to the new machining direction.



The same applies to the direction of circle rotation (G2/G3 or G3/G2).

If you program an additive rotation with AROT after MIRROR, you may have to work with reversed directions of rotation (positive/negative or negative/positive).
Mirrors on the geometry axes are converted automatically by the control into rotations and, where appropriate, mirrors on the mirror axis specified in the machine data. This also applies to settable zero offsets.

840 D
NCU 572
NCU 573

Machine manufacturer (MH 6.3)

SW 5 and higher

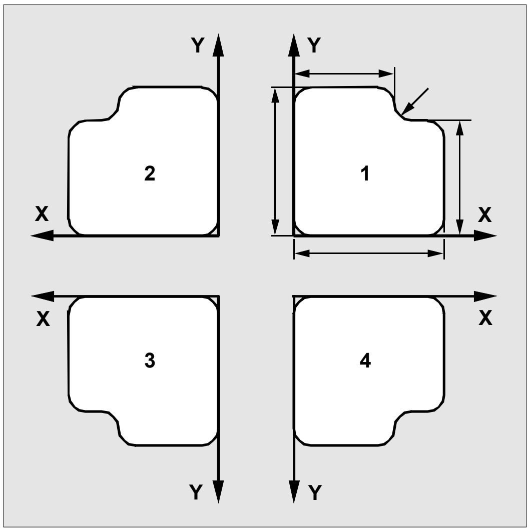
- You can set the axis around which mirroring is performed via MD.
  MD10610 = 0:
  Mirroring is performed in relation to the programmed axis (negation of values).
  MD10610 = 1 or 2 or 3:
  Depending on the data setting, mirroring is performed in relation to a specific reference axis (1=X axis; 2=Y axis; 3=Z axis) and rotations of two other geometry axes.
- MD10612 MIRROR_TOGGLE = 0 can be used to define that the programmed values are always evaluated. With a value of 0, as with MIRROR X0, the axis mirroring is deactivated and, with values not equal to 0, the axis is mirrored if it is not yet mirrored.

Programming example

Program the contour shown here once as a subprogram and generate the three other contours with a mirror operation.

The workpiece zero is located at the center of the contours.

Milling:



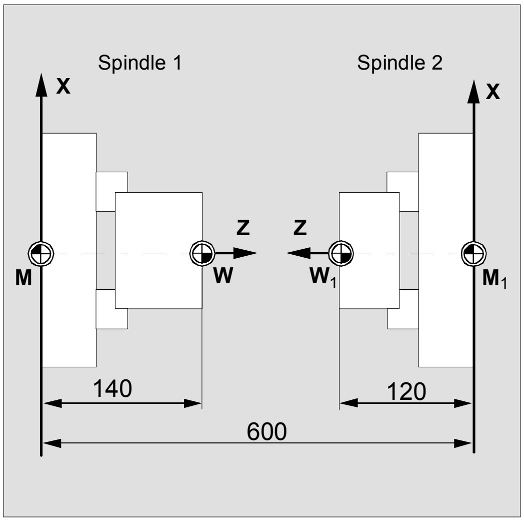
`N10`	`G17 G54`	Working plane X/Y, workpiece zero
`N20`	`L10`	Machine first contour, top right.
`N30`	`MIRROR X0`	Mirror X axis (the direction is changed in X).
`N40`	`L10`	Machine second contour, top left.
`N50`	`AMIRROR Y0`	Mirror Y axis (the direction is changed in Y).

840D
NCU 572
NCU 573

840Di

N60	L10	Machine third contour, bottom left
N70	MIRROR Y0	MIRROR cancels previous frames. Mirror Y axis (the direction is changed in Y).
N80	L10	Machine fourth contour, bottom right.
N90	MIRROR	Deactivate mirroring.
N100	G0 X300 Y100 M30	Retraction, end of program

Turning:



N10	TRANS X0 Z140	Zero offset to W
N..	...	Machine first side with spindle 1
N30	TRANS X0 Z600	Zero offset to spindle 2
N40	AMIRROR Z0	Mirroring of the Z axis
N50	ATRANS Z120	Zero offset to W_1
N..	...	Machine second side with spindle 2

840 D
NCU 572
NCU 573

6.8 Frame generation according to tool orientation, TOFRAME, TOROT

Programming

```
TOFRAME
TOROT
```

Explanation

The new frame, whose Z axis is pointing in the tool direction, is applicable after the block containing TOFRAME.
The rotation defined by TOROT is the same as for TOFRAME.

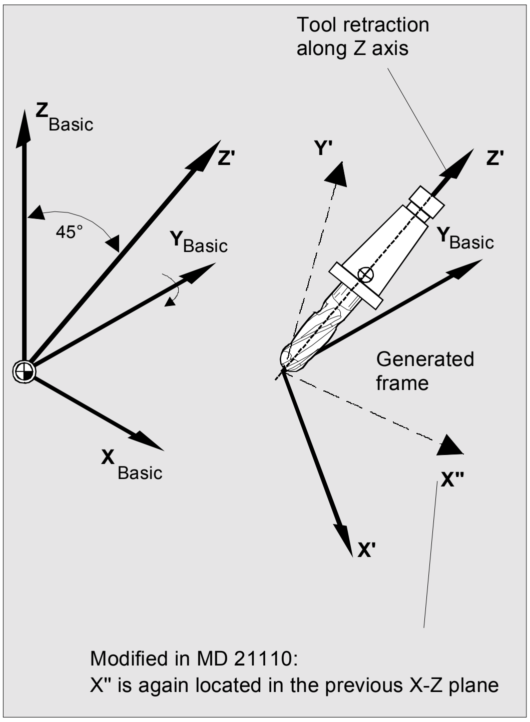
Function

TOFRAME generates a rectangular frame whose Z axis coincides with the current tool orientation.
You can use this function to retract the tool after a tool breakage in a 5-axis program without collision simply by retracting the Z axis.

The resulting frame describing the orientation is written in the system variable for the programmable frame `$P_PFRAME`.

Only the rotation component is overwritten with TOROT in the programmed frame. All other components remain unchanged.

The position of the two axes X and Y can be defined in MD21110: X_AXES_IN_OLD_X_Z_PLANE; X is rotated about Z in the previous X-Z plane.



Example:

```
N100 G0 G53 X100 Z100 D0
N120 TOFRAME
N140 G91 Z20        ; Frame TOFRAME is included; all programmed geometry
N160 X50            axis movements refer to TOFRAME
...
```

840D
NCU 572
NCU 573

840Di

Additional notes

After tool orientation has been programmed with TOFRAME, all the programmed geometry axis movements refer to the frame generated by this programming.

SW 6.1 and higher
Separate system frame for TOFRAME or TOROT.
The frames generated by TOFRAME or TOROT can be written to a separate system frame `$P_TOOLFRAME`.

Bit 3 in machine data MD 28082: MM_SYSTEM_FRAME_MASK must be set for this purpose.

The programmable frame remains unchanged. Differences occur when the programmable frame is processed further elsewhere.

NC command TOROT ensures consistent programming with active orientable tool carriers for each kinematic type. See also the description of NC command PAROT.

References

/PGA/ Programming Guide Advanced, Section "Tool Orientation"

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

6.9 Deselect frame SUPA, DRFOF, CORROF, TRAFOOF

Programming

`CORROF(axis,string[axis,string])` **or**
`CORROF(axis,string)` **or**
`CORROF(axis)` **or** `CORROF()`

Explanation of the commands

Deactivate coordinate transformation
A distinction must be made here between
- non-modal deactivation and
- modal deactivation.

	Non-modal deactivation:
G53	of all programmable and settable frames
G153	of all programmable, settable and basic frames
SUPA	of all programmable, settable frames, DRF handwheel offsets, external zero offsets and preset offset
	Modal deactivation:
G500	Deactivate all settable frames if G500 does not contain a value
DRFOF	Deactivation (deselection) of DRF handwheel offsets for all active axes in the channel
CORROF(axis,DRF[AXIS,AA_OFF])	Deactivation (deselection) of axial DRF offsets and the position offset for individual axes as a result of $AA_OFF (SW 6 and higher)
CORROF(axis)	All active overlaid motions are deselected (SW 6 and higher)
CORROF()	All active overlaid motions for all channel axes are deselected (SW 6 and higher)
TRAFOOF	Deactivate transformation

Explanation of the parameters

Axis	Axis identifiers for (channel, geometry or machine axis)
String == DRF	DRF_offsets of axis are deselected
String == AA_OFF	Position offset of axis is deselected due to $AA_OFF
	The following expansions are possible:
String == ETRANS	An active zero offset is deselected
String == FTOCOF	Acts like FTOCOF (deactivate online tool offset)
TRANS, ROT, SCALE, MIRROR	Delete programmable frames without axis specification

840D NCU 571 | 840D NCU 572 NCU 573 | 810D

Sequence

SW 6 and higher CORROF

A preprocessing stop is initiated and the position component of the deselected overlaid motion (DRF offset or position offset) is transferred to the position in the basic coordinate system. Since no axis is traversed, the value of $AA_IM[axis] does not change. Owing to the deselected overlaid motion, only the value of system variable $AA_IW[axis] is altered.

After the position offset, e.g. for one axis, has been deselected by $AA_OFF, the system variable $AA_OFF_VAL of this axis is zero.

In JOG mode as well, it is possible to activate interpolation of the position offset as an overlaid motion when $AA_OFF changes by setting bit 2 = 1 of MD 36750: AA_OFF_MODE.

Additional notes

CORROF is possible only from the parts program, not via synchronized actions.

Alarm 21660 is output if a synchronized action is active when the position offset is deselected via parts program command CORROF(axis,"AA_OFF"). $AA_OFF is deselected simultaneously and not set again. If the synchronized action becomes active later in the block after CORROF, $AA_OFF remains set and a position offset is interpolated.

If CORROF has been programmed for an axis that is active in another channel, then this axis is fetched to the other channel with MD 30552: AUTO_GET_TYPE = 0 on axis exchange. This causes the DRF offset and any other position offset to be deselected.

The programmable frames are cleared by specifying the TRANS, ROT, SCALE, MIRROR component without an axis.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

For more information about TRAFOOF see /PGA/ Programming Guide Advanced, Chapter 7 "5-Axis Transformation".

Programming examples

- **Axial DRF deselection**

A DRF offset is generated in the X axis by DRF handwheel traversal.
No DRF offsets are operative for any other axes in the channel.

N10 CORROF(X,"DRF") acts like DRFOF( )

A DRF offset is generated in the X and Y axes by DRF handwheel traversal.
No DRF offsets are operative for any other axes in the channel.

`N10   CORROF(X,"DRF")`	Only the DRF offset of the X axis is deselected, the X axis does not move
	the DRF offset of the Y axis is retained
	both offsets would have been deselected with DRFOF()

- **Axial DRF selection and $AA_OFF deselection**

A DRF offset is generated in the X axis by DRF handwheel traversal.
No DRF offsets are operative for any other axes in the channel.

`N10   WHEN TRUE DO $AA_OFF[X] = 10 G4 F5`	A position offset == 10 is interpolated for the X axis
`N70   CORROF(X,"DRF",X,"AA_OFF")`	Only the DRF offset of the X axis is deselected, the X axis does not move
	the DRF offset of the Y axis is retained

840D NCU 571 | 840D NCU 572 NCU 573 | 810D

- **$AA_OFF deselection**

A position offset of the X axis is deselected with: CORROF(X,"AA_OFF") with $AA_OFF[X] = 0 and added to the current position of the X axis.

The following programming example shows the relevant programming examples for the X axis that was previously interpolated with a position offset of 10:

`N10 WHEN TRUE DO $AA_OFF[X] = 10 G4 F5`	A position offset == 10 is interpolated for the X axis
`N80 CORROF(X,"AA_OFF")`	Delete position offset of X axis the X axis does not move

■

6.9 Deselect frame SUPA, DRFOF, CORROF, TRAFOOF

840D
NCU 571

840D
NCU 572
NCU 573

810D

840Di

Feedrate Control and Spindle Motion

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.1 Feedrate

Programming

```
G93 or G94 or G95
F…
FGROUP(X, Y, Z, A, B, …)
FL[axis]=…
FGREF[axis name]=reference radius (SW 5
and higher)
```

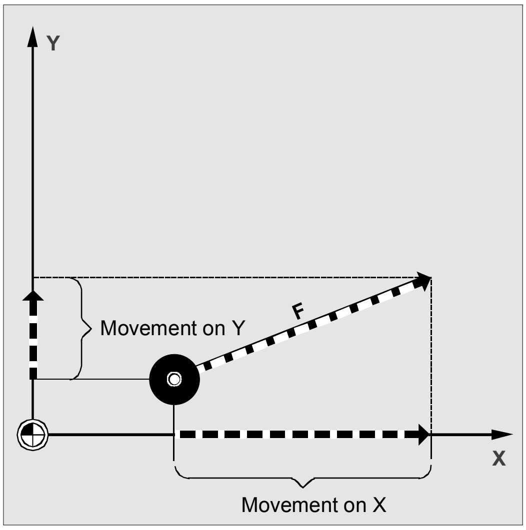
Explanation of the commands

`G93`	Inverse-time feedrate in rpm (**SW 5.2** and higher **on 840D NCU 572/573** and **SW 3.2** and higher **on 810D CCU2**)
`G94`	Feedrate in mm/min or inches/min or in deg/min
`G95`	Feedrate in mm/rev or inches/rev
`F…`	Feedrate value in unit defined by G93, G94, G95
`FGROUP`	Feedrate value F valid for all axes specified in FGROUP
`FGREF`	Effective radius (reference radius) for the rotary axes entered in FGROUP (SW 5 and higher)
`FL`	Limit speed for synchronized axes; the unit set with G94 applies (max. rapid traverse)
`Axis`	Channel axes or geometry axes

Function

You can use the above commands to set the feedrates in the NC program for all axes participating in the machining sequence.

The path feed is generally composed of the individual speed components of all geometry axes participating in the movement and refers to the center point of the cutter or the tip of the turning tool.



Note:
The inverse-time feedrate rpm G93 is not implemented for 802D and up to SW 3.1 for 810D CCU1.

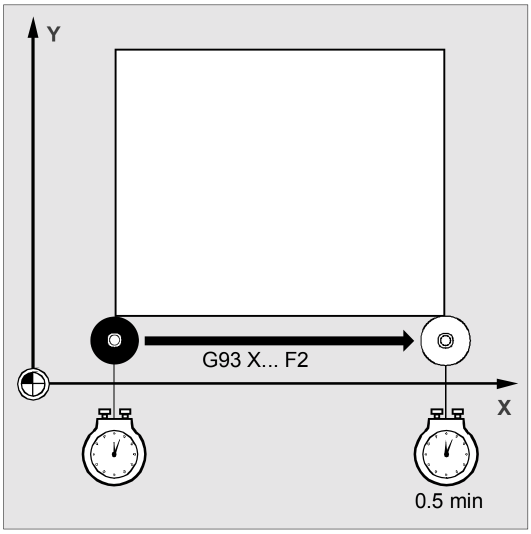
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

Units of measurement for feedrate F
You can use the following G commands to define the units of measurement for the feed input. All of the commands are modal. Input is in mm or inches, according to the default setting in the machine data. Feed parameters are not affected by G70/G71.

With G700/G710 in SW 5 and higher, feed values F are interpreted like geometrical parameters in the system of units set by G function (G700: [inch/min]; G710: [mm/min]).

Feedrate G93
Unit rpm. The inverse-time feedrate specifies the time required to execute the motion commands in a block.
Example:
N10 G93 G01 X100 F2 means: the programmed path is traversed in 0.5min.



Note: If the path lengths vary greatly from block to block, a new F value should be specified in each block with G93. The feed can also be specified in deg/rev when machining with rotary axes.

Feedrate G94
mm/min or inches/min and deg/min

Feedrate G95
mm/rev or inches/rev with reference to the speed of the master spindle – generally the cutting spindle or the main spindle on the turning machine.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

If the G feedrate command is switched between G93, G94 or G95 the path feedrate must be reprogrammed.
The feed can also be specified in deg/rev when machining with rotary axes.

Feedrate F for path axes
The feedrate is specified with address F. One F value can be programmed per NC block. The unit for the feedrate is defined in one of the above G commands.
The feed F acts only on path axes and remains active until a new feedrate is programmed.
Separators are permitted after address F.
Example:
`F100` or `F 100` or `F.5` or `F=2*FEED`

Feedrate for synchronized axes
The feed F programmed at address F applies to all the path axes programmed in the block, but not to synchronized axes.

The synchronized axes are controlled such that they require the same time for their path as the path axes, and all axes reach their end point at the same time.

Traverse synchronized axes with path velocity F, FGROUP
With FGROUP, you define whether a path axis is to be traversed with path feed or as a synchronized axis.
In helical interpolation, for example, you can define that only two geometry axes, X and Y, are to be traversed at the programmed feedrate.
The infeed axis Z is the synchronized axis in this case.

Example: `N10 FGROUP(X, Y)`

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Change FGROUP

1. By reprogramming another FGROUP instruction.

Example: `FGROUP(X, Y, Z)`

2. With FGROUP () without axis parameter

Afterwards, the initial setting in the machine data applies – the geometry axes again traverse in the path axis grouping.

You must program channel axis names with FGROUP.

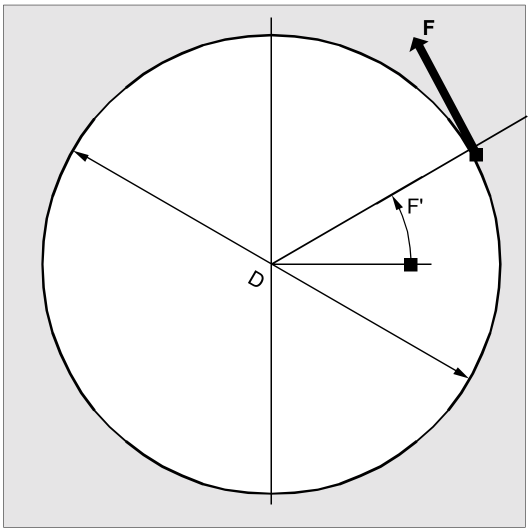
Machine manufacturer (MH7.1)

See machine manufacturer's specifications

Unit of measurement for rotary and linear axes

For linear and rotary axes which are combined with FGROUP and traverse a path together, the feed is interpreted in the unit of measurement of the linear axes.

Depending on the default for G94/G95: mm/min or inch/min and mm/rev or inch/rev.



The tangential velocity of the rotary axis in mm/min or inch/min is calculated according to the following formula:

$$F[mm/min] = \frac{F'[degree/min] * \pi * D[mm]}{360\,[degree]}$$

F: Tangential velocity

F': Angular velocity

π : Circle constant

D: Diameter

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Traverse rotary axes with path velocity F, FGREF (SW 5 and higher)

For machining operations in which the tool or the workpiece or both are moved by a rotary axis, the effective machining feedrate is to be interpreted as a path feed in the usual way by reference to the F value. This requires the specification of an effective radius (reference radius) **FGREF** for each of the rotary axes involved.

The unit of the reference radius depends on the G70/G71/G700/G710 setting.

All axes involved must be included in the FGROUP command, as before, in order to be evaluated in the calculation of the path feed.

In order to maintain compatibility with the behavior without FGREF programming, the following setting is activated after system power-up and on RESET:
1 degree = 1mm.
This corresponds to a reference radius of
FGREF=360mm/(2π)=57.296mm.

This default setting is independent of the active basic system MD 10240: SCALING_SYSTEM_IS_METRIC and of the currently active inch/metric G code.

Special situations:
With the following programming:

```
N100 FGROUP(X,Y,Z,A)
N110 G1 G91 A10 F100
N120 G1 G91 A10 X0.0001 F100
```

the F value programmed in N110 is evaluated as a rotary axis feedrate in deg/min, while the feedrate weighting in N120 is either 100inch/min or 100mm/min depending on the currently active inch/metric setting.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

The FGREF evaluation is active even if only rotary axes are programmed in the block. The normal F-value interpretation as degree/min applies in this case only if the radius reference corresponds to the FGREG default, i.e.

- *G71/G710: FGREF[A]=57.296*
- *G70/G700: FGREF[A]=57.296/25.4*

The following example illustrates the effect of FGROUP on the path and the path feedrate.
The variable $AC_TIME contains the time from the start of the block in seconds. It can be used only in synchronized actions. See /FBSY/, Synchronized Actions

Example

N100 G0 X0 A0			
N110 FGROUP(X,A)			
N120 G91 G1 **G710** F100	Feed=100 mm/min or 100 deg/min		
N130 DO $R1=$AC_TIME			
N140 X10	Feed=100 mm/min	Path=10mm	R1=approx. 6s
N150 DO $R2=$AC_TIME			
N160 X10 A10	Feed=100 mm/min	Path=14.14mm	R2=approx. 8s
N170 DO $R3=$AC_TIME			
N180 A10	Feed=100 deg/min	Path=10 degrees	R3=approx. 6s
N190 DO $R4=$AC_TIME			
N200 X0.001 A10	Feed=100 mm/min	Path=10mm	R4=approx. 6s
N210 **G700** F100	Feed=2540 mm/min or 100 deg/min		
N220 DO $R5=$AC_TIME			
N230 X10	Feed=2540 mm/min	Path=254mm	R5=approx. 6s
N240 DO $R6=$AC_TIME			
N250 X10 A10	Feed=2540 mm/min	Path=254.2mm	R6=approx. 6s
N260 DO $R7=$AC_TIME			
N270 A10	Feed=100 deg/min	Path=10 degrees	R7=approx. 6s
N280 DO $R8=$AC_TIME			
N290 X0.001 A10	Feed=2540 mm/min	Path=10mm	R8=approx. 0.288s
N300 FGREF[A]=360/(2*$PI)	Set 1 degree=1 inch via the effective radius		
N310 DO $R9=$AC_TIME			
N320 X0.001 A10	Feed=2540 mm/min	Path=254mm	R9=approx. 6s
N330 M30			

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Traverse synchronized axes with limit speed FL

With this command, synchronized/path axes are traversed at their limit speed FL. The path velocity of the path axes is reduced if the synchronized axis reaches the limit speed.

Example, Z is a synchronized axis:

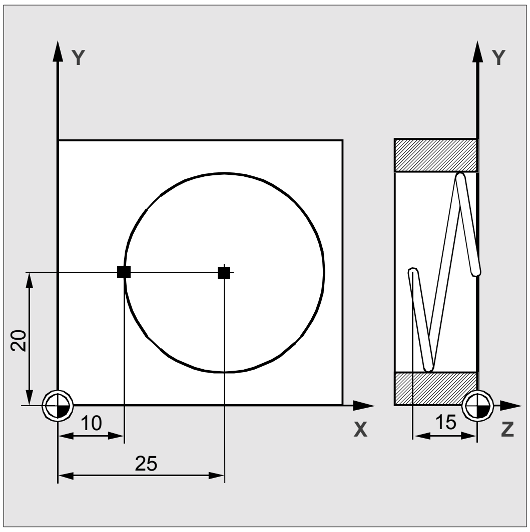
```
N10 G0 X0 Y0
N20 FGROUP(X)
N30 G1 X1000 Y1000 G94 F1000 FL[Y]=500
N40 Z-50
```

One FL value can be programmed per axis. The axis identifiers of the basic coordinate system must be used (channel axes or geometry axes). The unit of measurement set for F (G70/G71) using the G command is also valid for FL. If no FL is programmed, the rapid traverse velocity applies. FL is deselected by assignment to MD $MA_AX_VELO_LIMIT.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

Helical interpolation. Path axes X and Y traverse with the programmed feedrate, the infeed axis Z is a synchronized axis.



`N10 G17 G94 G1 Z0 F500`	Tool infeed
`N20 X10 Y20`	Approach start position
`N25 FGROUP(X, Y)`	Axes X/Y are path axes, Z is a synchronized axis
`N30 G2 X10 Y20 Z-15 I15 J0 F1000 FL[Z]=200`	On the circular path, the feed is 1000 mm/min. Traversing in the Z direction is synchronized.
`...`	
`N100 FL[Z]=$MA_AX_VELO_LIMIT[0,Z]`	The limit velocity is deselected when the velocity value is read from the MD.
`N110 M30`	End of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.2 Traversing positioning axes, POS, POSA, POSP

Programming

```
POS[axis]=…
POSA[axis]=…
POSP[axis]=(…,…,…)
FA[axis]=…
WAITP(axis)=…              (programmed in a separate NC block)
WAITMC(mark)=...
```

Explanation of the commands

`POS [axis]=`	Position the axis; the next NC block is not enabled until the position has been reached
`POSA [axis]=`	Position the axis; the next NC block is enabled, even if the position has not been reached
`POSP [axis]=(,,)`	Approach end position in sections. The first value indicates the end position; the second the length of the section. Approaching the end position is defined in the third value with 0 or 1
`FA[axis]=`	Feedrate for the positioning axis, up to 5 per NC block
`WAITP(axis)`	Wait for the axis to finish traversing; WAITP must be programmed in a separate NC block
`WAITMC(marker)`	During the braking ramp, WAIITMC loads the next NC block immediately when the WAIT marker is received.
`Axis`	Channel axes or geometry axes
`Mark, ,`	An axis is only decelerated if the marker has not yet been reached or if an different search criteria prevents the block change.

Function

Positioning axes are traversed independently of the path axes at a separate, axis-specific feedrate. There are no interpolation commands.

Example for positioning axes: Pallet feeding mechanisms, measuring stations, etc.

Sequence

With the POS/POSA/POSP commands, the positioning axes are traversed and the sequence of motions coordinated at the same time.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Traveling with POSA[…]=

The axis indicated in brackets is traversed to the end position. The block step enable or program execution is not affected by POSA. The movement to the end position can be performed during execution of subsequent blocks.

Internal preprocessor stop

If a command which generates an implicit preprocessor stop is read in the following block, the following block is not executed until all previously prepared and stored blocks have been processed completely. The preceding block is halted in an exact stop (as for G9).

Example:

```
N40 POSA[X]=100
N50 IF $AA_IM[X]==R100 GOTOF LABEL1   ; On accessing status data of the machine ($A...), the control generates an internal preprocessing stop, processing is halted until all previously prepared and stored blocks have been executed in full.
N60 G0 Y100
N70 WAITP(X)
N80 LABEL1:
N..
```

Traveling with POS[…]=

The next block is only executed when all axes programmed under POS have reached their end positions.

Traveling with POSP[…]=

POSP is used specifically for programming oscillating movements
(see /PGA/ Programming Guide Advanced, Chapter 11).

Wait for end of travel with WAITP(…)

WAITP can be used for:

- Identifying a position in the NC program where the program is to wait until an axis programmed with POSA in a previous NC block has reached its end position.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

- Making an axis available as a reciprocating axis.
- Making an axis available for traversing as a concurrent positioning axis (by PLC).

After WAITP, assignment of the axis to the NC program is no longer valid; this applies until the axis is programmed again.
This axis can then be operated as a positioning axis through the PLC, or as a reciprocating axis from the NC program/PLC or MMC.

Programming example

Axis U: Pallet store, transporting the pallet to the working area

Axis V: Transfer line to a measuring station, where sampling controls are carried out.

`N10 FA[U]=100 FA[V]=100`	Axis-specific feed functions for each positioning axis U and V
`N20 POSA[V]=90 POSA[U]=100 G0 X50 Y70`	Traverse positioning and path axes
`N50 WAITP(U)`	Execution of the program is only continued when axis U has reached the end position programmed in N20.
`N60 …`	

Block change in the braking ramp with IPOBRKA and WAITMC(…)

In SW 6.4 and higher, WAITMC can be used to
- to load the next NC block immeditaely when the wait marker is received.
- to decelerate an axis only if the marker has not yet been reached or if an different search criteria prevents the block change.

After a WAITMC, the axes start immediately if no other search criterion prevent block change.

840D NCU 571	840D NCU 572 NCU 573	810D	840Di

7.3 Position-controlled spindle operation, SPCON, SPCOF

Programming

`SPCON` or `SPCON(n)`
`SPCOF` or `SPCOF(n)`

Explanation of the commands

`SPCON` `SPCON(n)`	Switch master spindle or spindle number n from speed control to position control
`SPCOF` `SPCOF(n)`	Switch master spindle or spindle number n back from position control to speed control
`SPCON` `SPCON(n, m, 0)`	SW 3.5 and higher: Several spindles with number n can be switched from closed-loop speed control to position control in one block
`SPCOF` `SPCOF(n, m, 0)`	SW 3.5 and higher: Several spindles with number n can be switched from closed-loop position control back to speed control in one block
`n` `m`	Integers from 1 ... n Integers from 1 ... m

Function

In certain cases, it is practical to operate the spindle in position control mode.
For example, a higher quality can be achieved for thread cutting operations with G33 and a large pitch.

Note
The command requires up to three interpolation cycles.

Sequence

The speed is specified with S... M3, M4 and M5 apply to the directions of rotation and spindle stop. SPCON is modal and remains active until SPCOF.

Additional notes

With synchronized spindle setpoint value linkage, the master spindle must be operated in position control mode.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.4 Positioning spindles (position-controlled axis operation): SPOS, M19 and SPOSA

Programming

`SPOS=…` or `SPOS[n]=…`
`M19` or `M[n]=19`
`SPOSA=…` or `SPOSA[n]=…`
`M70` or `Mn=70`
`FINEA=…` or `FINEA[n]=…`
`COARSEA=…` or `COARSEA[n]=…`
`IPOENDA=…` or `IPOENDA[n]=…`
`IPOBRKA=…` or `IPOBRKA(axis[,REAL])` (programming in a separate NC block)
`WAITS` or `WAITS(n,m)` (programmed in a separate NC block)

Explanation of the commands

`SPOS=` `SPOS[n]=`	Position master spindle (SPOS) or spindle number n (SPOS[n]); the next NC block is not enabled until the position has been reached
`M19` `M[n]=19`	Position master spindle (M19) or spindle number n (M[n]=19); the next NC block is not enabled until the position has been reached. (SW 5.3 and higher)
`SPOSA=` `SPOSA[n]=`	Position master spindle with SPOSA or spindle number n (SPOSA[n]). The next NC block is enabled, even if the position has not been reached
`M70` `Mn=70`	Switch over master spindle (M70) or spindle number n (Mn=70) to axis operation. No defined position is approached. The NC block is enabled after the switchover has been performed.
`FINEA=` `FINEA[Sn]=`	End of motion when "Exact stop fine" is reached (SW 5.1 and higher)
`COARSEA=` `COARSEA[Sn]=`	End of motion when "Exact stop coarse" is reached (SW 5.1 and higher)
`IPOENDA=` `IPOENDA[Sn]=`	End of motion when "IPO stop" is reached (SW 5.1 and higher)

`IPOBRKA=`
`IPOBRKA(axis`
`[,real])=`
	End of motion criterion from moment of application of braking ramp at 100% down to end of braking ramp at 0% and identical to IPOENDA (as of SW 6) IPOBKRA must be programmed in round parenthesis "()".

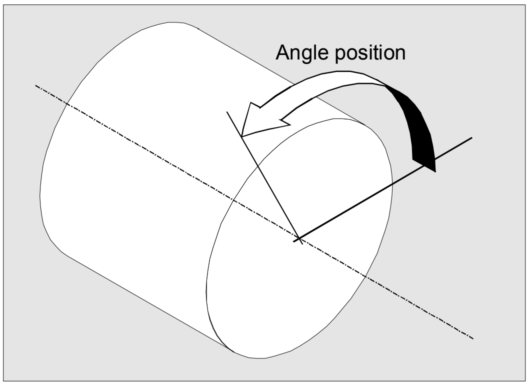
`WAITS` `WAITS(n,m)`	Wait for spindle position to be reached. WAITS applies to the master spindle or the specified spindle numbers

`n`
`m`
`Sn`
`Axis`
 	
`Real`
	Integers from 1 ... n
	Integers from 1 ... m
	nth Spindle number, 0 to max. spindle number
	Channel identifier
	Percentage specification 100-0% referred to the braking ramp for block changes. If no % is specified, the current value of the setting data is applied.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Function

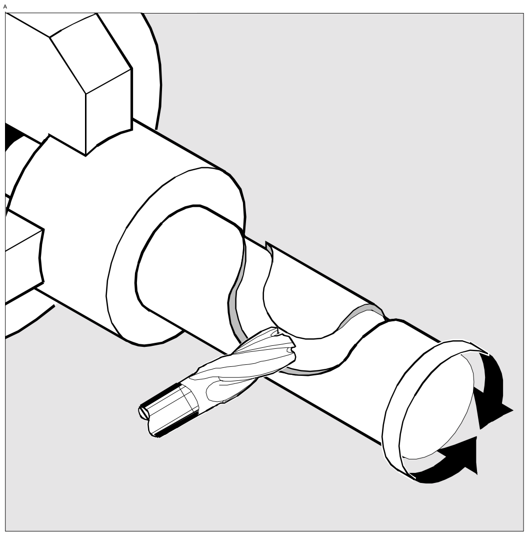
SPOS, M19 and SPOSA can be used to position spindles at specific angles, e.g. for tool-changing operations. The spindle can also be operated as a path axis, synchronized axis or positioning axis at the address defined in the machine data. When the axis identifier is specified, the spindle is in axis mode. M70 switches the spindle directly to axis mode.
Example:



N10	M3 S500	
...		
N90	SPOS[2]=0 or	Position control on, spindle 2 positioned to 0, axis mode can be used in the next block
	M2=70	Spindle 2 is switched to axis mode
N100	X50 C180	Spindle 2 (C axis) is traversed with linear interpolation synchronous to X.
N110	Z20 SPOS[2]=90	Spindle 2 is positioned to 90 degrees.

Sequence

Precondition
The spindle must be capable of operation in position control mode.
Position with SPOSA=, SPOSA[n]=
The block step enable or program execution is not affected by SPOSA. The spindle positioning can be performed during execution of subsequent blocks. The program moves onto the next block if all the functions (except for spindle) programmed in the current block have reached their block end criterion. The spindle positioning operation may be programmed over several blocks (see WAITS).



If a command which implicitly causes a preprocessor stop is read in a subsequent block, execution of this block is delayed until all positioning spindles are stationary.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Position with SPOS=, SPOS[n]= and position with M19=, M19[n]=

The block step enabling condition is fulfilled when all functions programmed in the block have reached their block end criterion (e.g. all auxiliary functions acknowledged by the PLC, all axes have reached end point) and the spindle has reached the programmed position.

Speed of the movements

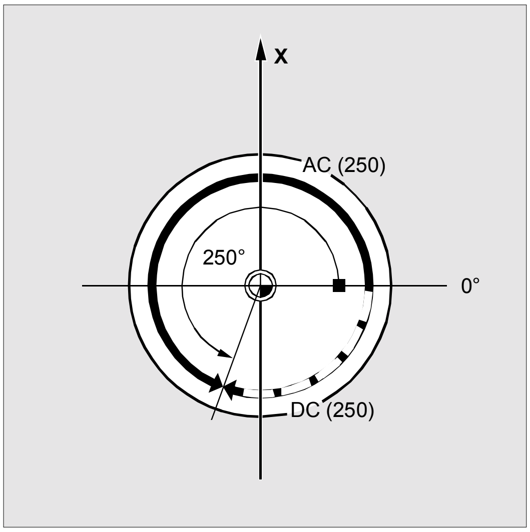
The speed or delay response for positioning is stored in the machine data and can be programmed.

Specify spindle position

The spindle position is specified in degrees. Since the commands G90/G91 do not apply here, the following explicit references apply:

`AC(…)`	Absolute dimensions
`IC(…)`	Incremental dimensions
`DC(…)`	Approach absolute value directly
`ACN(…)`	Absolute dimensions, approach in negative direction
`ACP(…)`	Absolute dimensions, approach in positive direction

With IC, spindle positioning can take place over several revolutions.



Example:

Position spindle 2 at 250° in negative direction of rotation.

`N10 SPOSA[2]=ACN(250)` The spindle decelerates if necessary and accelerates in the opposite direction to the positioning movement (SW 4 and higher)

If nothing is specified, traversing automatically takes place as for DC. Three spindle positions are possible for each NC block.

Value range

Absolute dimensions AC: 0…359.9999 degrees

Incremental dimensions IC: 0…±99,999.999 degrees

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

End of positioning (SW 5.1 and higher)
Programmable by means of the following commands:
FINEA[Sn], COARSEA[Sn], IPOENDA[Sn].

Settable block change time (SW 6 and higher)
For single axis interpolation mode, a new end of motion can be set in addition to the existing end of motion criteria based on FINEA, COARSEA, IPOENDA.
The new criterion can be set within the braking ramp (100-0%) **using IPOBRKA**.
The program advances to the next block if the end of motion criteria for all spindles or axes programmed in the current block plus the block change criterion for path interpolation are fulfilled.

Example:

```
N10 POS[X]=100
N20 IPOBRKA(X,100)
N30 POS[X]=200
N40 POS[X]=250
N50 POS[X]=0
N60 X10 F100
N70 M30
```

Block changes if the X axis has reached position 100 and exact stop fine.
Activate **block change criterion IPOBRKA** braking ramp. Block change commences as soon as the X axis starts to decelerate.
The X axis does not brake at position 200, but moves on to position 250; as soon as the X axis starts to brake, the block changes.
The X axis brakes and returns to position 0, the block is changed at position 0 and exact stop fine.

Deactivate
SPOS, M19 and SPOSA cause a temporary changeover to position-control mode until the next M3 or M4 or M5 or M41 to M45. If the position control was activated with SPCON prior to SPOS, then this remains active until SPCOF is issued.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Synchronize spindle movements, WAITS, WAITS(n,m)

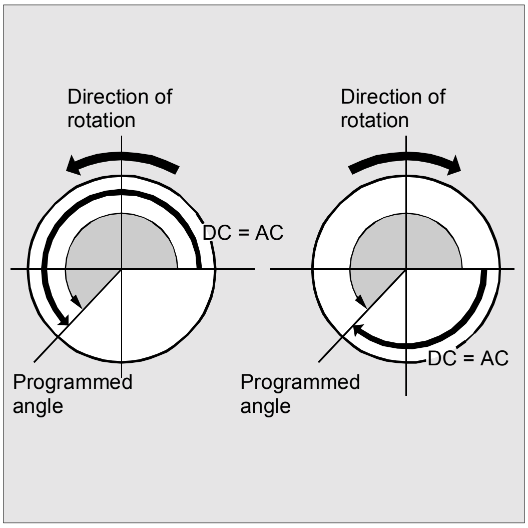
WAITS can be used to identify a point at which the NC program waits until one or more spindles programmed with SPOSA in a previous NC block have reached their positions.

Example:
```
N10 SPOSA[2]=180 SPOSA[3]=0
N20…N30
N40 WAITS(2,3)
```

The block waits until spindles 2 and 3 have reached the positions specified in block N10.

Position spindle from rotation (M3/M4)

When M3 or M4 active, the spindle comes to a standstill at the programmed value.



There is no difference between DC and AC dimensioning. In both cases, rotation continues in the direction selected by M3/M4 until the absolute end position is reached.
With ACN and ACP, deceleration takes place if necessary, and the appropriate approach direction is followed.

With IC, the spindle rotates additionally to the specified value starting at the current spindle position.
When M3 or M4 is active, the spindle decelerates if necessary, and accelerates in the programmed direction of rotation.

Position a spindle from standstill (M5)

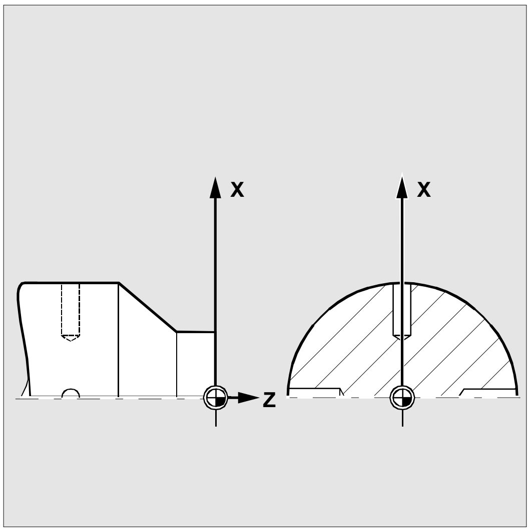
The exact programmed distance is traversed from standstill (M5).

If the spindle has not yet been synchronized with synchronization marks, the positive direction of rotation is taken from the machine data (state on delivery).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

Cross holes are to be drilled in this turned part. The running drive spindle (master spindle) is stopped at zero degrees and then successively turned through 90°, stopped and so on.



....		
N110	S2=1000 M2=3	; switch on cross drilling attachment
N120	**SPOSA=DC(0)**	; position main spindle **directly** at 0°, the program will advance to the next block **immediately**
N125	G0 X34 Z-35	; switch on the drill while the spindle is being positioned
N130	**WAITS**	; wait until the main spindle reaches its position
N135	G1 G94 X10 F250	; feedrate in mm/min (G96 is suitable only for the multi-edge turning tool and synchronous spindle, but not for power tools on the cross slide)
N140	G0 X34	
N145	**SPOS=IC(90)**	; the spindle is positioned **through 90°** with read halt in a positive direction
N150	G1 X10	
N155	G0 X34	
N160	**SPOS=AC(180)**	; the spindle is **positioned at 180°** with respect to the spindle zero point
N165	G1 X10	
N170	G0 X34	
N175	**SPOS=IC(90)**	; the spindle turns in a positive direction **through** 90° from the absolute 180° position, ending up in the absolute 270° position
N180	G1 X10	
N185	G0 X50	
...		

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.5 Milling on turned parts: TRANSMIT

Programming

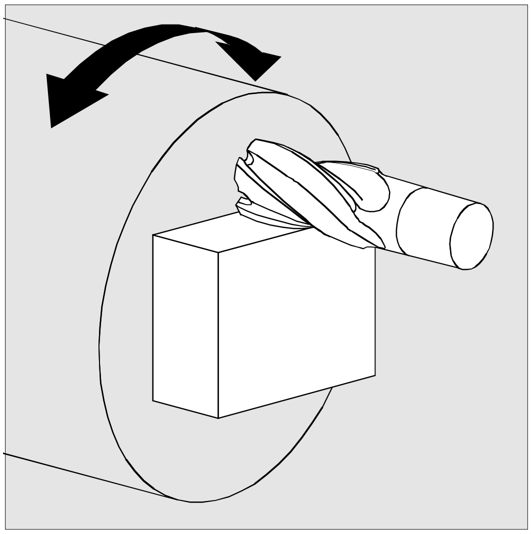
`TRANSMIT` or `TRANSMIT(n)`
`TRAFOOF`

Explanation of the commands

`TRANSMIT`	Activates the first declared TRANSMIT function
`TRANSMIT(n)`	Activates the nth declared TRANSMIT function; n can be up to 2 (TRANSMIT(1) is the same as TRANSMIT).
`TRAFOOF`	Deactivates an active transformation

An active TRANSMIT transformation is likewise deactivated if one of the other transformations is activated in the relevant channel (e.g. TRACYL, TRAANG, TRAORI).

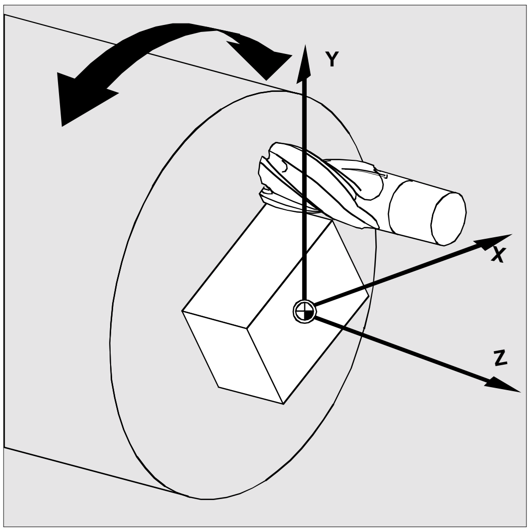
The TRANSMIT function enables the following:



- Face machining on turned parts in the turning clamp (drill-holes, contours).
- A Cartesian coordinate system can be used to program these operations.
- The control maps the programmed traversing movements of the Cartesian coordinate system onto the traversing movements of the real machine axes (standard situation):
  – Rotary axis
  – Infeed axis perpendicular to the axis of rotation
  – Longitudinal axis parallel to the axis of rotation
  The linear axes are perpendicular to each other.
- A tool center offset relative to the turning center is permitted.
- The velocity control makes allowance for the limits defined for the rotations.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example



`N10 T1 D1 G54 G17 G90 F5000 G94`	Tool selection
`N20 G0 X20 Z10 SPOS=45`	Approach start position
`N30 TRANSMIT`	Activate TRANSMIT function
`N40 ROT RPL=–45` `N50 ATRANS X–2 Y10`	Set frame

`N60 G1 X10 Y–10 G41`
`N70 X–10`
`N80 Y10`
`N90 X10`
`N100 Y–10`
`N110 ...`
	Four-edge roughing

References

/PGA/ Programming Guide Advanced, Section "Transformations"

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.6 Cylinder surface transformation: TRACYL

Programming

TRACYL(d) or TRACYL(d,t)
TRAFOOF

Explanation of the commands

TRACYL(d)	Activates the first declared TRACYL function
TRACYL(d,n)	Activates the nth declared TRACYL function. N can be up to 2, TRACYL(d,1) is the same as TRACYL(d).
d	Value for the current diameter of the cylinder to be machined.
TRAFOOF	Transformation off

An active TRACYL transformation is likewise deactivated if one of the other transformations is activated in the relevant channel (e.g. TRANSMIT, TRAANG, TRAORI).

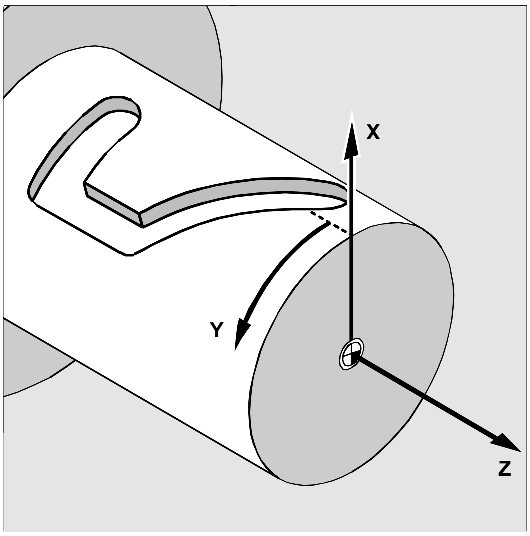
Function

Cylinder surface transformation TRACYL
The TRACYL cylinder surface transformation function can be used to

Machine

- longitudinal grooves on cylindrical bodies,
- transverse grooves on cylindrical bodies,
- grooves with any path on cylindrical bodies.

The path of the grooves is programmed with reference to the unwrapped, level surface of the cylinder.



Workpiece coordinate system

References

/PGA/ Programming Guide Advanced, Section "Transformations"

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.7 Feedrate for positioning axes/spindles: FA, FPR, FPRAON, FPRAOF

Programming

`FA[axis]=`
`FA[SPI(spindle)]=…` or `FA[S…]=…`
`FPR (rotary axis)` or `FPR(SPI(spindle))` or `FPR(S…)`
`FPRAON (axis,rotary axis)` or
`FPRAON(axis,SPI(spindle))` or `FPRAON(axis,S…)` or
`FPRAON(SPI(spindle),rotary axis)` or `FPRAON(S…,rotary axis)` or
`FPRAON(SPI(spindle),SPI(spindle))` or `FPRAON(S…,S…)` or
`FPRAOF(axis,SPI(spindle),…)` or `FPRAOF(axis,S…,…)`

Explanation of the commands

`FA[axis]`	Feedrate for the specified positioning axis in mm/min or inch/min or deg/min
`FA[SPI(spindle)]` `FA[S…]`	Positioning velocity (axial feed) for the specified spindles in deg/min
`FPR`	Identification of the rotary axis or spindle whose revolutional feedrate programmed in G95 is to be used as the basis for the revolutional feedrate of the **path and synchronized axes**.
`FPRAON`	Activate revolutional feedrate for **positioning axes and spindles** axially. The first command identifies the positioning axis/spindle that is to be traversed at a revolutional feedrate. The second command identifies the rotary axis/spindle from which the feedrate must be derived.
`FPRAOF`	Deactivate revolutional feedrate. Specification of axis or spindle that is to stop traversing at a revolutional feedrate.
`SPI`	Converts the spindle number into an axis identifier; the transfer parameter must contain a valid spindle number. SPI is used for the indirect definition of a spindle number.
`Axis`	Positioning axes or geometry axes

Function

Positioning axes, such as workpiece transport systems, tool turrets and end supports, are traversed independently of the path and synchronized axes. A separate feed is therefore defined for each positioning axis.

Example: `FA[A1]=500`

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

With synchronous spindle link, the positioning speed of the following spindle can be programmed independently of the master spindle – for example, for positioning operations.

Example: `FA[S2]=100`

The spindle identifiers SPI(...) and S... are identical in terms of function.

Sequence

Feed FA[…]

The programmed feed is modal. The feed is always G94.

When G70/G71 is active, the unit of measurement is metric/inches according to the default setting in the machine data. G700/G710 can be used to modify the unit of measurement in the program.

If no FA is programmed, the value defined in the machine data applies.

Up to 5 feeds for positioning axes or spindles can be programmed in each NC block.

Value range

0.001…999,999.999mm/min, deg/min
0.001…39,999.9999inch/min

Feed FPR(…)

As an extension of the G95 command (revolutional feedrate referring to the master spindle), FPR allows the revolutional feedrate to be derived from any chosen spindle or rotary axis.

G95 FPR(...) is valid for path and synchronized axes.

If the rotary axis/spindle specified in the FPR command is operating on position control, then the setpoint linkage is active. Otherwise the actual-value linkage is effective.

The derived feedrate is calculated according to the following formula:

Derived feedrate =
programmed feedrate * Absolute master feedrate

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Example:
Path axes X, Y must be traversed at the revolutional feedrate derived from rotary axis A:

```
N40 FPR(A)
N50 G95 X50 Y50 F500
```

Feed FPRAON(…,…), FPRAOF(…,…)

The FPRAON command makes it possible to derive the revolutional feedrate for specific positioning axes and spindles from the current feedrate of another rotary axis or spindle.

The first command identifies the axis/spindle that must be traversed at a revolutional feedrate. The second command identifies the rotary axis/spindle that is to supply the feedrate. The command need not be specified a second time. If it is not, the feedrate is derived from the master spindle.

The revolutional feedrate can be deactivated for one or several axes/spindles simultaneously with the FPRAOF command.

The feedrate is calculated in the same way as for FPR(…).

Examples:
The revolutional feedrate for master spindle 1 must be derived from spindle 2.

```
N30 FPRAON(S1,S2)
N40 SPOS=150
N50 FPRAOF(S1)
```

The revolutional feedrate for positioning axis X must be derived from the master spindle. The positioning axis is traversing at 500 mm/revolution of the master spindle.

```
N30 FPRAON(X)
N40 POS[X]=50 FA[X]=500
N50 FPRAOF(S1)
```

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.8 Percentage feedrate override, OVR, OVRA

Programming

```
OVR=…
OVRA[axis]=…
OVRA[SPI(spindle)]=… or OVRA[S…]=…
```

Explanation of the command

OVR	Feed change in percent for path feed F
OVRA	Feed change in percent for positioning feed FA or for spindle speed S
SPI	Converts the spindle number into an axis identifier; the transfer parameter must contain a valid spindle number. The spindle identifiers SPI(...) and S... are identical in terms of function.
Axis	Positioning axes or geometry axes

Function

You can use the programmable feedrate override to change the velocity/speed of path axes, positioning axes and spindles via a command in the NC program.

Example:

```
N10 OVR=25 OVRA[A1]=70        ; Path feed 25%,
                              Positioning feed for A1 70%.
N20 OVRA[SPI(1)]=35           ; Spindle speed for spindle 1 35%.
or
N20 OVRA[S1]=35
```

Sequence

The programmed feed change refers to or is combined with the feed override set on the machine control panel.

Example:
Set feed override 80%
Programmed feed override OVR=50
The programmed path feed F1000 is changed to F400 (1000 * 0.8 * 0.5).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Value range

1…200%, integers; with path and rapid traverse override, the maximum velocities set in the machine data are not exceeded.

7.9 Feedrate with handwheel override, FD, FDA

Programming

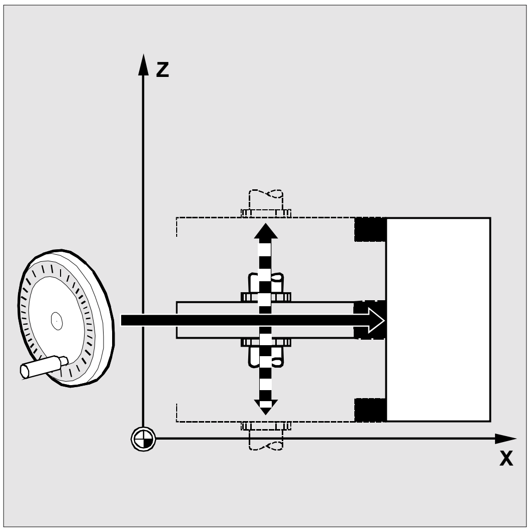
```
FD=…
FDA[axis]=0 or FDA[axis]=…
```

Explanation of the command

`FD=…`	Handwheel travel for path axes with feed override
`FDA [axis]=0`	Handwheel travel for positioning axes according to position parameter
`FDA[axis]=…`	Handwheel travel for positioning axes with feed override
`Axis`	Positioning axes or geometry axes

Function

With these functions, you can use the handwheel to traverse path and positioning axes (position parameter) or change the axis velocities (speed override) during program execution.
The handwheel override is frequently used for grinding operations.



Example for position parameter:
The grinding wheel oscillating in the Z direction is moved to the workpiece in the X direction using the handwheel. The operator can then adjust the position of the tool until the spark generation is constant. When "Delete distance-to-go" is activated, the program goes to the next NC block and machining continues in NC mode.

Only speed override can be used for path axes.

7.9 Feedrate with handwheel override, FD, FDA

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

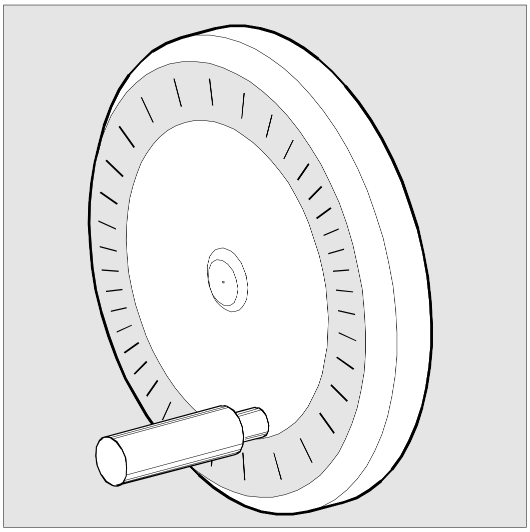
Sequence

Preconditions

A handwheel must be assigned to the axes to be traversed for the handwheel override function. Please see the Operator's Guide for the procedure. The number of handwheel pulses per graduated position is defined in machine data.

Non-modal operation

The handwheel override function is non-modal. The function is deactivated in the next NC block and the NC program continues to be executed.



Handwheel travel with path default for positioning axes, FDA[axis]=0

In NC blocks with programmed `FDA[axis]=0`, the feed is set to zero in order that the program does not generate any travel movement. The programmed travel movement to the target position is now controlled exclusively by the operator rotating the handwheel.

Example: `N20 POS[V]=90 FDA[V]=0`

The automatic travel movement is stopped in block N20. The operator can now move the axis manually using the handwheel.

Direction of movement, travel velocity

The axes accurately follow the path set by the handwheel in the direction of the leading sign. Depending on the direction of rotation, you can travel forwards or backwards – the faster you turn the handwheel, the higher the travel velocity.

Traversing range

The traversing range is limited by the starting position and the end point programmed with the positioning command.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Handwheel travel with velocity overlay, FDA[axis]=…

Where `FDA[…]=…` is programmed in an NC block, the feed from the last programmed FA value is accelerated or reduced to the value programmed at FDA.

Starting with the current feed FDA, you can accelerate or reduce the programmed travel to zero by rotating the handwheel. The values defined in the machine data are used for the maximum velocity.

Example:
```
N10 POS[U]=10 FDA[U]=100
POSA[V]=20 FDA[V]=150
```

Traverse path axes with handwheel override, FD

The following preconditions apply to handwheel overrides for path axes:

In the NC block with the programmed handwheel override

- an active G1, G2 or G3 motion command must be active,
- exact stop G60 must be switched on, and
- the path feedrate must be specified with G94 mm/min or inch/min.

The path feed F and the handwheel override FD may not be programmed in the same NC block.

Feedrate override

The feed override acts only on the programmed feed, not on the travel movement generated with the handwheel (except if feed override = 0).

Example:
```
N10 G1 X… Y… F500…
N50 X… Y… FD=700
```

The feedrate is accelerated to 700 mm/min in block N50. The path velocity can be increased or reduced according to the direction of rotation on the handwheel.

It is not possible to traverse in the opposite direction.

7.9 Feedrate with handwheel override, FD, FDA

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Note

With velocity override of path axes, you always control the path velocity with the handwheel of the 1st geometry axis.

Traversing range

The traversing range is limited by the starting position and the programmed end point.

Manual override in automatic mode

The manual override function in automatic mode for POS/A axes has two different effects that are analogous to Jog functions.

1. Path override: FDA [ax] = 0

The axis does not move. Handwheel pulses received for each IPO cycle are traversed direction-independent and accurate to the path. When the target position is reached, the axis brakes.

2. Velocity override FDA [ax] > 0

The axis moves to the target position at the programmed velocity. The destination is thus reached even without handwheel pulses.

Pulses received for each IPO cycle are converted to an accumulative change in the existing velocity.

Pulses in the traversing direction increase the speed; the limit is MAX_AX_VELO.

Pulses in the opposite direction reduce the speed. The minimum speed limit is 0.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.10 Percentage acceleration correction: ACC (Option)

Programming

ACC[axis]=…
ACC[SPI(spindle)]=… or ACC(S…)

Explanation of the command

ACC	Change in acceleration in percent for the specified path axis or change in speed for the specified spindle
SPI	Converts the spindle number into an axis identifier; the transfer parameter must contain a valid spindle number. The spindle identifiers SPI(...) and S... are identical in terms of function.
Axis	Channel axis name of path axis

Function

In critical program sections, it may be necessary to limit the acceleration to below the maximum values, e.g. to prevent mechanical vibrations from occurring.

Sequence

You can use the programmable acceleration override to change the acceleration for each path axis or spindle via a command in the NC program. The limit is effective for all types of interpolation. The values defined in the machine data determine the 100% acceleration.

Example N50 ACC[X]=80

Meaning Traverse the axis slide in the X direction with only 80% acceleration.

N60 ACC[SPI(1)]=50 or ACC[S1]=50

Meaning Accelerate or decelerate spindle 1 with only 50% of the maximum acceleration. The spindle identifiers SPI(...) and S... are identical in terms of function.

Value range 1…200%, integers

Deactivate ACC[axis]=100, program start, reset

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Additional notes

Please note that the maximum permissible values of the machine manufacturer can be exceeded with a higher acceleration rate.

SW 5.1 and higher
The defined acceleration can also be changed via synchronized actions.
See /FBSY/, Synchronized Actions

Example

```
N100 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140
```

The last programmed value is retained after RESET.
The current acceleration value can be polled using the system variables $AA_ACC[<Axis>].

7.11 Feedrate optimization for curved path sections, CFTCP, CFC, CFIN

Programming

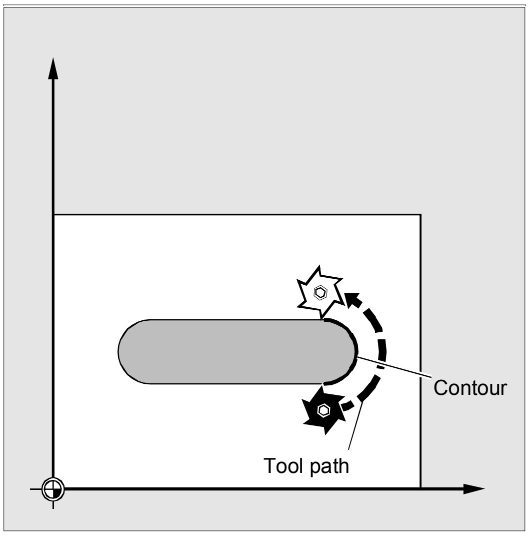
```
CFTCP
CFC
CFIN
```

Explanation of the commands

`CFTCP`	Constant feed at cutter center path
`CFC`	Constant feed at contour (tool edge)
`CFIN`	Constant feed at tool edge for concave contours only, otherwise on the cutter center path

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Function



The programmed feedrate initially refers to the cutter center path when the G41/G42 override is activated for the cutter radius (cf. Chapter 6).

When you mill a circle – the same applies to polynomial and spline interpolation – the extent of the change in feedrate at the cutter edge is such that it can have a considerable effect on the quality of the machined part.

Example: you are milling a small external radius with a large tool. The distance that the outer side of the cutter has to cover is much larger than the distance along the contour. You therefore work with a very small feed on the contour.

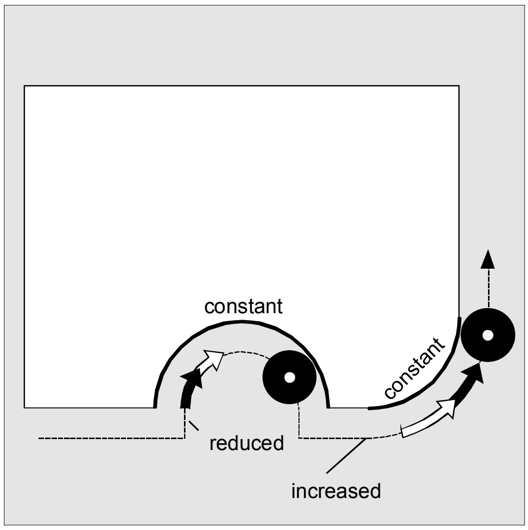
In order to avoid effects like this, you should regulate the feedrate for curved contours accordingly.

Sequence

Constant feedrate on center-point path, deactivate feedrate override, CFTCP
The control keeps the feedrate constant, feed overrides are deactivated.

Constant feedrate on contour, CFC
The feedrate is reduced for inside radii and increased for outside radii. This ensures a constant speed at the tool edge and thus at the contour.
This function is set as the default.

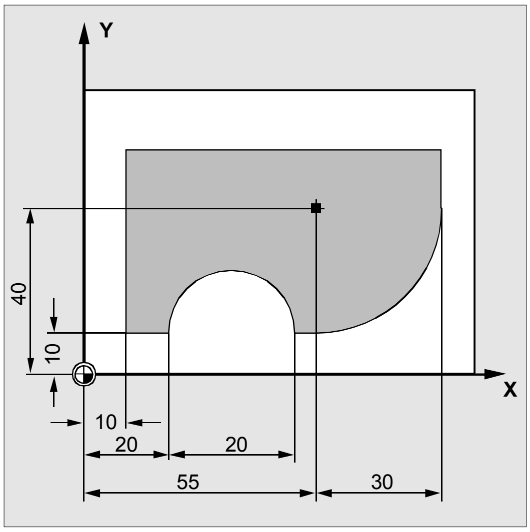
Constant feed only at inside radii, CFIN
The feedrate is reduced for inside radii but not increased for outside radii; the cutter center applies.



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

In this example, the contour is first machined with a CFC-compensated feed.
During finishing, the cutting base is additionally machined with CFIN. This prevents the cutting base from being damaged at outside radii by too high a feedrate.



N10	G17 G54 G64 T1 M6	
N20	S3000 M3 CFC F500 G41	
N30	G0 X-10	
N40	Y0 Z-10	Infeed to first cutting depth
N50	CONTOUR1	Subprogram call
N40	CFIN Z-25	Infeed to second cutting depth
N50	CONTOUR1	Subprogram call
N60	Y120	
N70	X200 M30	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.12 Spindle speed S, direction of spindle rotation M3, M4, M5

Programming

```
M3 or M4 or M5
M1=3 or M1=4 or M1=5
S…
Sn=…
SETMS(n) or SETMS
```

Explanation of the commands

`M1=3 M1=4 M1=5`	Spindle rotation clockwise/counterclockwise, spindle stop for spindle 1. Other spindles are defined according to M2=… M3=…
`M3`	Direction of spindle rotation clockwise for master spindle
`M4`	Direction of spindle rotation counterclockwise for master spindle
`M5`	Spindle stop for master spindle
`Sn=…`	Spindle speed in rpm for spindle n
`S…`	Spindle speed in rpm for the master spindle
`SETMS(n)`	Set spindle specified in n as master spindle
`SETMS`	Reset to the master spindle defined in machine data

Function

The above functions are used to

- switch on the spindle,
- specify the required direction of spindle rotation, and
- define the counterspindle or an actuated tool as the master spindle, e.g. on turning machines

The following programming commands are valid for the master spindle: G95, G96, G97, G33, G331.
(see also Chapter 1, "Main spindle, master spindle").

Machine manufacturer (MH7.2)

Definition as master spindle is also possible via machine data (default).

Sequence

Preset M commands, M3, M4, M5
In a block with axis commands, the above mentioned functions are activated *before* the axis movements commence (basic settings on the control).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Example:
```
N10 G1 F500 X70 Y20 S270 M3
N100 G0 Z150 M5
```

N10: the spindle accelerates to 270rpm, then the movements are performed on X and Y.
N100: spindle stop before the retraction movement on Z.

A machine data can be set to determine whether the axis motions are delayed until the spindle has run up and reached setpoint speedor stopped, or whether they are executed immediately after the programmed switching operations.

Spindle speed S
The speed specified with S... or S0=... applies to the master spindle. You specify the corresponding number for additional spindles: =…, S2=…

Three S values can be programmed per NC block.

Working with multiple spindles
Up to five spindles – master spindle plus four additional spindles, two spindles on the SINUMERIK FM-NC– can be configured in the same channel.

One of the spindles is defined in machine data as the **master spindle**. Special functions apply to this spindle, such as thread cutting, tapping, revolutional feed, dwell time.

The numbers must be specified with the speed and the direction of rotation/spindle stop for the other spindles, e.g. for a second spindle and actuated tool.

Example:
```
N10 S300 M3 S2=780 M2=4
```

Master spindle 300rpm, clockwise rotation
2nd spindle 780rpm, counterclockwise rotation

840D NCU 571	840D NCU 572 NCU 573	810D	840Di

Programmable switchover of master spindle, SETMS(n)

You can define any spindle as the master spindle with a command in the NC program.

Example:

`N10 SETMS(2)`	; SETMS must be programmed in a separate block
Spindle 2 is now the master spindle.	

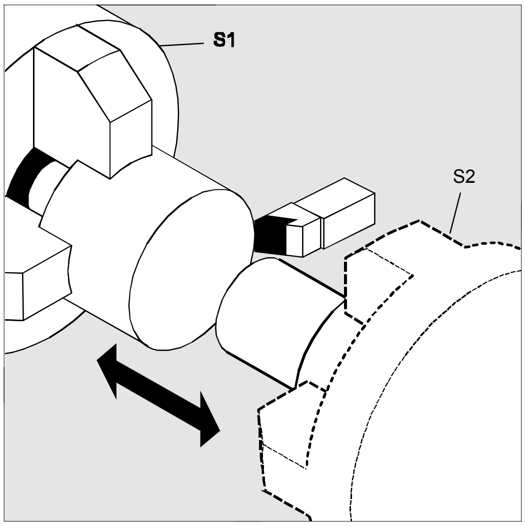
The speed specified with S and M3, M4, M5 now apply.

Deactivate

By issuing SETMS without spindle parameter you can switchback to the master spindle defined in the machine data.

Programming example

S1 is the master spindle, S2 is the second workspindle. The part is to be machined from two sides. To do this, it is necessary to divide the operations into steps. After parting, the synchronous device (S2) takes up the workpiece for machining on the parted side. To do this, this spindle S2 is defined as the master spindle to which G95 then applies.



`N10 S300 M3`	Speed and direction of rotation for drive spindle = preset master spindle
`N20…N90`	Machining of right side of workpiece
`N100 SETMS(2)`	S2 is now master spindle
`N110 S400 G95 F…`	Speed for new master spindle
`N120…N150`	Machining of left side of workpiece
`N160 SETMS`	Switch back to master spindle S1

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.13 Constant cutting rate, G96, G97, LIMS

Programming

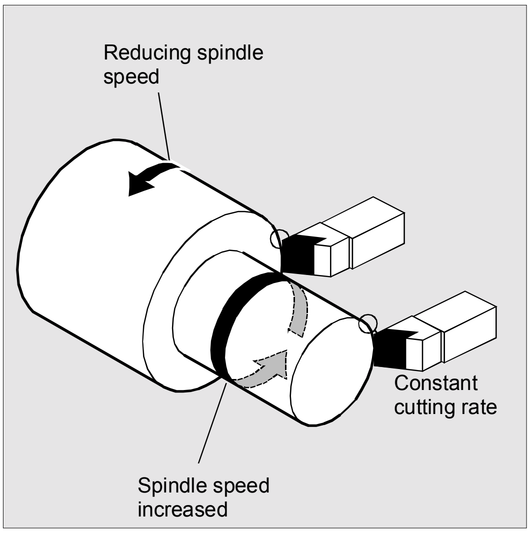
```
G96 S…
G97
LIMS=…
```

Explanation of the commands

`G96`	Activate constant cutting rate
`S`	Cutting rate in m/min, always applies to master spindle
`G97`	Deactivate constant cutting rate
`LIMS`	Speed limitation when G96 is active (applies to master spindle)

Function

When G96 is active, the spindle speed – depending on the respective workpiece diameter – is automatically modified in order that the cutting rate S in m/min remains constant at the tool edge.
This increases the uniformity and thus the surface quality of turned parts.



Sequence

Value range for cutting rate S
The precision can be set in the machine data.
The range for the cutting rate can be between **0.1m/min ... 9999 9999.9m/min**.

For G70/G700: cutting rate in feet/min.

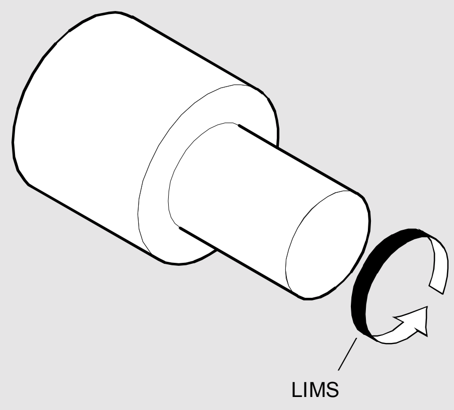
Adjust feed F
When G96 is active, G95 feed is automatically activated in mm/rev.

If G95 was not already active, you must specify a new feedrate F when you call G96 (e.g. convert F value from mm/min to mm/rev).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Upper speed limit LIMS

If you machine a workpiece that varies greatly in diameter, it is advisable to specify a speed limit for the spindle. This prevents excessively high speeds with small diameters.
LIMS is active with G96 and G97.



Example:
```
N10 SETMS(3)
N20 G96 S100 LIMS=2500
```
Limit speed to 2500rpm

The speed limit programmed with G26 or defined in machine data cannot be exceeded with LIMS.

Rapid traverse

With rapid traverse G0, there is no change in speed.
Exception: If the contour is approached in rapid traverse and the next NC block contains a G1, G2, G3 ... path command, the speed is adjusted in the G0 approach block for the next path command.

Deactivate constant cutting rate, G97

After G97, the control interprets an S word again as a spindle speed in rpm.
If you do not specify a new spindle speed, the last speed set with G96 is retained.

Additional notes

- The G96 function can also be deactivated with G94 or G95.
  In this case, the last programmed speed S is used for further machining operations.
- In SW 4.2 and higher, G97 can also be programmed without preceding it by G96. The function then has the same effect as G95; LIMS can also be programmed.

The transverse axis must be defined in machine data.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.14 Constant grinding wheel peripheral speed, GWPSON, GWPSOF

Programming

```
GWPSON(T No.)
GWPSOF(T No.)
S…
S1…
```

Explanation of the commands and parameters

GWPSON(T No.)	Select constant grinding wheel peripheral speed GWPS It is only necessary to specify the T number if the tool with this T number is not active.
GWPSOF(T No.)	Deselect GWPS; the T number only need be specified if the tool with this T number is not active
S… S1…	Program GWPS; value for peripheral speed in m/s or ft/s S… SUG for master spindle; S1… SUG for spindle 1

Function

With the function "Constant grinding wheel peripheral speed" (=GWPS) you can set the grinding wheel speed such that, taking account of the current radius, the grinding wheel peripheral speed remains constant.

The GWPS can only be selected for grinding tools (types 400–499).

Additional notes

To be able to activate the "Constant peripheral speed" function, the tool-specific grinding data $TC_TPG1, $TC_TPG8 and $TC_TPG9 must be set accordingly.
When the GWPS function is active, even online offset values (= wear parameters; cf. Sect. 6 PUTFTOC, PUTFTOCF) are taken into account with regard to speed variations!

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Select GWPS GWPSON, program GWPS

After activation of GWPSON, each subsequent S value for this spindle is interpreted as a grinding wheel peripheral speed.

Selection of grinding wheel peripheral speed with GWPSON does not cause the automatic activation of tool length compensation or tool monitoring.

The GWPS can be active for several spindles on a channel with different tool numbers.

If GWPS is to be selected for a new tool on a spindle where GWPS is already active, the active GWPS must first be deselected with GWPSOF.

Deactivate GWPS GWPSOF

When the GWPS is deselected with GWPSOF, the last speed determined is retained as set speed.

GWPS programming is reset at the end of the parts program or on RESET.

Query active GWPS $P_GWPS[spindle no.]

This system variable can be used to query from the parts program whether the GWPS is active for a specific spindle.

TRUE GWPS is **active**.

FALSE GWPS is **not active**.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

A constant grinding wheel peripheral speed is to be used for grinding tools T1 and T5.
T1 is the active tool.

Programming

`N20 T1 D1`	Select T1 and D1
`N25 S1=1000 M1=3`	1000rpm for spindle 1
`N30 S2=1500 M2=3`	1500rpm for spindle 2
…	
`N40 GWPSON`	GWPS selection for active tool T1
`N45 S1 = 60`	Set GWPS for active tool to 60m/s
…	
`N50 GWPSON(5)`	Select GWPS for tool 5 (2nd spindle)
`N55 S2 = 40`	Set GWPS for spindle 2 to 40m/s
…	
`N60 GWPSOF`	Deactivate GWPS for active tool
`N65 GWPSOF(5)`	Deactivate GWPS for tool 5 (spindle 2)
…	

840D
NCU 572
NCU 573

840Di

7.15 Constant workpiece speed for centerless grinding: CLGON, CLGOF

Programming

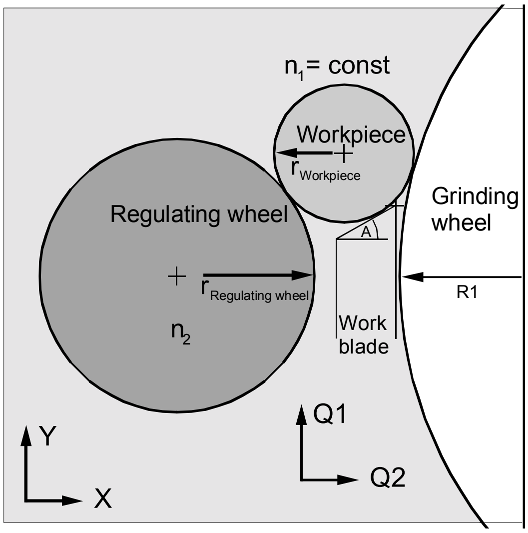
```
CLGON(set)
CLGOF
```

Explanation of the commands

`CLGON(set)`	Activate "Constant workpiece speed for centerless grinding" function; specify speed setpoint (set) for workpiece in rpm
`CLGOF`	Deactivate function

Function

When the "Constant workpiece speed for centerless grinding" function is active, the speed of the machined part is kept constant. The speed of the regulating wheel reduces as the diameter of the machined part decreases.



Sequence

Preconditions for CLGON

The axes of the regulating and grinding wheels and the work blade must be in a position that enables grinding of the workpiece from the initial to the finishing dimension.

CLGON acts only if the spindle of the regulating wheel is running in speed mode. An actual position sensor is not required.

G functions G94, G95, G96 and G97 can be active at the same time as CLGON. These functions have no effect on the spindle of the regulating wheel.

If the regulating wheel is running is a master spindle, G96 and CLGON cancel each other.

Additional notes

The following are stored in the channel-specific machine data (`$MC_TRACLG...`)

- The spindle numbers of the regulating and grinding wheels
- Geometrical parameters (axis numbers, direction vector of the work blade, etc.)
- Response on reset and end of program

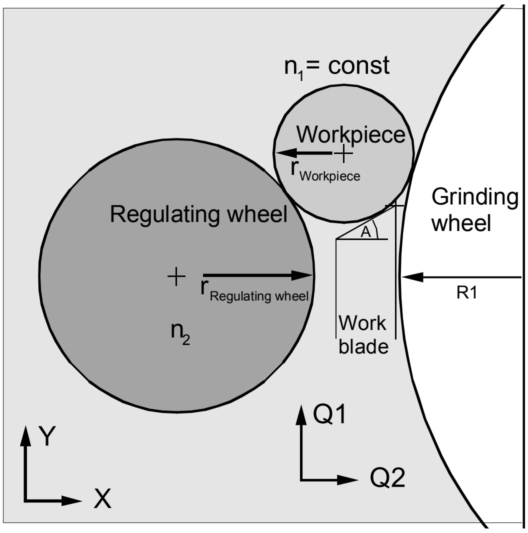
840D
NCU 572
NCU 573

840Di

Calculation of the regulating wheel speed

The speed of the regulating wheel is calculated from the set speed of the workpiece:

$S_{Regulating\ wheel} = r_{Workpiece}/r_{Regulating\ wheel} \cdot S_{prog}$



The radius of the workpiece $r_{Workpiece}$ is calculated as the radius of the circle touching the grinding wheel, regulating wheel and work blade.

Offset data for CLGON

The radii of the grinding and regulating wheels are taken from the current offset data for T1, D1 (grinding wheel) and T2, D1 (regulating wheel). Changes to the online tool offset (PUTFTOCF, FTOCON, FTOCOF) are taken into account.

Response on transition of traversing blocks

CLGON is only active in traversing blocks without G0 (traverse with path feed).

If a transition takes place from a G0 block to a traversing block without G0, the speed of the regulating wheel during the G0 block is set to the desired starting speed of the next block.

If a G0 block follows a traversing block without G0, the speed is frozen at the end of the block before G0. This does not apply if the G0 block is followed by a motion block without G0 in which a new set speed is programmed.

Gear stages

The gear stages must be selected in such a manner that the regulating wheel can cover the entire speed range required.

Monitoring

Speed monitoring defined with G25 and G26 is active.

The range of the work blade, in which the calculated tangent of the workpiece lies, is monitored. It is defined in machine data.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.16 Programmable spindle speed limitation, G25, G26

Programming

```
G25 S… S1=… S2=…
G26 S… S1=… S2=…
```

Explanation of the commands

`G25`	Lower spindle speed limitation
`G26`	Upper spindle speed limitation
`S S1=… S2=…`	Minimum or maximum speed

Function

You can use a command in the NC program to change the minimum and maximum spindle speeds defined in the machine data and setting data.

Sequence

It is possible to program spindle speed limitations for all spindles on the channel.

Example:

```
N10 G26 S1400 S2=350 S3=600
```

Upper speed limitation for master spindle, spindle 2 and spindle 3.

Value range

Value assignment for the spindle speed can be between **0.1rpm ... 9999 9999.9rpm**.

A spindle speed limitation programmed with G25 or G26 overwrites the speed limitations in the setting data and thus remains stored after the end of the program.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

7.17 Several feedrates in one block: F.., FMA..

Programming

`F2=... to F7=...`	Several path movements in one block
`ST=...`	
`SR=...`	
`FMA[2,x]=... to FMA[7,x]=...`	Several axial movements in one block
`STA=...`	
`SRA=...`	

Explanation of the commands

`F2=... to F7=...`	In addition to the path feed, you can program up to 6 further feedrates in the block; non modal
`ST=...`	Dwell time (for technology grinding: sparking out time); non modal
`SR=...`	Return path; non modal
`FMA[2,x]=... to FMA[7,x]=...`	In addition to the path feed, you can program up to 6 further feedrates per axis in the block; non modal
`STA=...`	Axis-spec. dwell time (for grinding technology: sparking out time); non modal
`SRA=...`	Axis-spec. return path; non modal

Function

The function "Several feedrates in one block" can be used independent of external analog and/or digital inputs to activate

- 6 different feedrates of an NC block,
- 1 dwell time and
- 1 return path

in synchronism with the movement.
The HW inputs signals are combined in one input byte, see description in:
/FB/ A2, Various Interface Signals.

Sequence

Programming the path motion
The path feedrate is programmed under the address A, which remains valid provided that no input signal is present. The numerical extension is indicated by the bit number of the input, whose change activates the feedrate:

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

e.g.

```
F7=1000     ; 7 corresponds to input bit 7.
F2=20       ; 2 corresponds to input bit 2.
ST=1        ; dwell time (s) input bit 1
SR=0.5      ; return path (mm) input bit 0
```

Programming the axial motion

The axial path feedrate is programmed under the address FA, which remains valid provided that no input signal is present.

FMA[7,x]=... to FMA[2,x]=... can also be used to program up to 6 further feedrates per axis in the block. The first expression in the square brackets indicate the bit number of the input; the second the axis for which the feedrate is to apply

e.g. `FMA[3,y]=1000` ; axial feedrate with the value 1000 for Y axis, 3 corresponds to input bit 3.

Dwell time and return path are programmed under the additional addresses

STA[x]=... Dwell time (s) bit 1 and

SRA[x]=... Return path (mm) bit 0.

Additional notes

- The axial feedrate/path feedrate (F value) corresponds to 100% feedrate. The function "Several feedrates in one block" can be used to implement feedrates which are less than or equal to the axial feedrate/path feedrate.
- If feedrates, swell time or return path are programmed for an axis on account of an external input, this axis must in this block must not be programmed as POSA axis (positioning axis over multiple blocks).
- If input bit 1 is activated for the dwell time or bit 0 for the return path, the distance to go for the path axes or the relevant single axes is deleted and the dwell time or return started.
- The unit for the return path is relative to the currently valid unit of measurement (mm or inch).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

- Look-ahead is also active for several feedrates in one block. In this way, you can use the look-ahead function to restrict the current feedrate.

Programming

`N20 T1 D1 F500 G0 X100`	Start position
`N25 G1 X105 F=20 F7=5 F3=2.5 F2=0.5 ST=1.5 SR= 0.5;` Standard feedrate with F, roughing with F7, finish cut with F3, smooth-finishing with F2, dwell 1.5s, return path 0.5mm	
`N30 ...`	
...	

7.18 Blockwise feedrate: FB... (as of SW 5.3)

Programming

`FB=...` Feed motion only in one block

Explanation of the commands

`FB=...`	Instead of the modal feedrate active in the previous block, you can program a separate feedrate for this block; in the block that follows, the previously active modal feedrate applies.

Function

You can use the function "Blockwise feedrate" to define a separate feedrate for a single block.

Sequence

The address FB is used to define the feedrate only for the current block. After this block, the previously active modal feedrate is active.

The feedrate is interpreted according to the active feedrate type:

- G94: feedrate in mm/min or °/min
- G95: feedrate in mm/rev. or inch/rev.
- G96: constant cutting speed

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

References: Description of Functions, V1 Feedrates

Additional notes

- The feedrate programmed value of FB=<value> must be greater than zero.
- If no traversing motion is programmed in the block (e.g. computation block), the FB has no effect.
- If no explicit feedrate is programmed for chamfer/rounding, the value of FB also applies for a chamfer/rounding contour element in this block.
- Feedrate interpolations FLIN, FCUB, etc. are also possible without restriction.
- Simultaneous programming of FB and FD (handwheel traversing with handwheel override) or F (modal path feed) is **not** possible.

Programming

`N10 G0 X0 Y0 G17 F100 G94;`	Start position
`N20 G1 X10;`	Feedrate 100 mm/min
`N30 X20 FB=80;`	Feedrate 80 mm/min
`N40 X30;`	Feedrate is 100 mm/min again
`N50 ...`	
...	

■

7.18 Blockwise feedrate: FB... (as of SW 5.3)

840D NCU 571

840D NCU 572 NCU 573

810D

840Di

Tool Offsets

840D
NCU 571

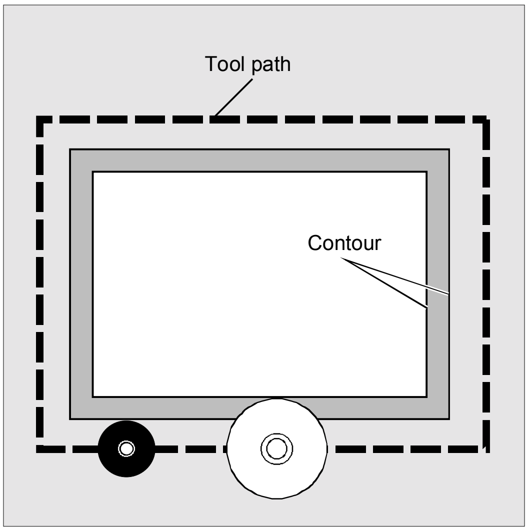
840D
NCU 572
NCU 573

810D

840Di

8.1 General information

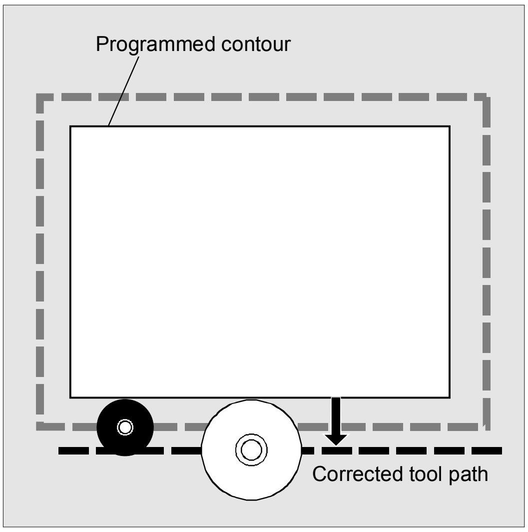
Why use tool offsets?



When writing a program, it is not necessary to specify the cutter diameter, the tool point direction of the turning tool (left/right-handed turning tools) or tool length.
You simply program the workpiece dimensions according to the dimensions in the production drawing.

When machining a workpiece, the tool paths are controlled according to the tool geometry such that the programmed contour can be machined using any tool.

The control corrects the traverse path



You enter the tool data separately in the tool table on the control.
All you need to do is call the required tool with its offset data in the program.

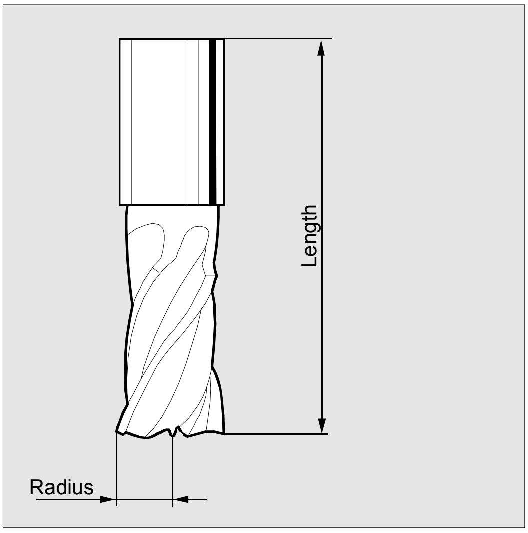
During program execution, the control fetches the offset data from the tool files and corrects the tool path individually for different tools.

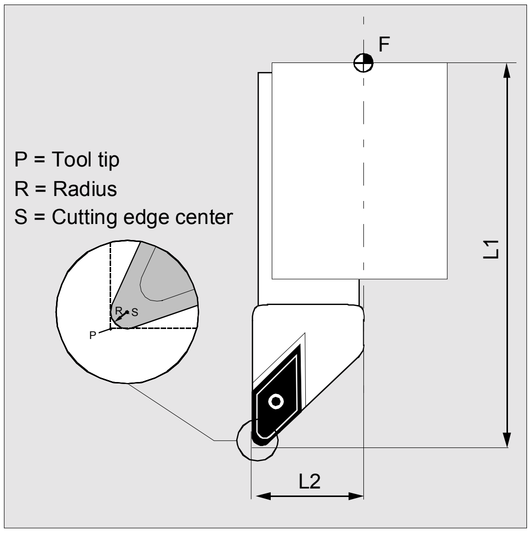
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Which tool offsets are stored in the control's compensation memory?

In the compensation memory enter the following:

- Geometrical dimensions: Length, radius.
  They consist of several components (geometry, wear). The control computes the components to a certain dimension (e.g. overall length 1, total radius). The respective overall dimension becomes effective when the compensation memory is activated.
  These values are calculated in the axes according to the tool type and the current plane G17, G18, G19.
- Tool type
  The type determines which geometry axes are needed and how they are calculated (drill or milling tool or turning tool).
- Tool point direction





Tool parameters

The following section "List of tool types" describes the individual tool parameters in the display. The relevant tool parameters must be entered in the input fields with "DP...". Any unneeded tool parameters must be set to "zero".

Caution

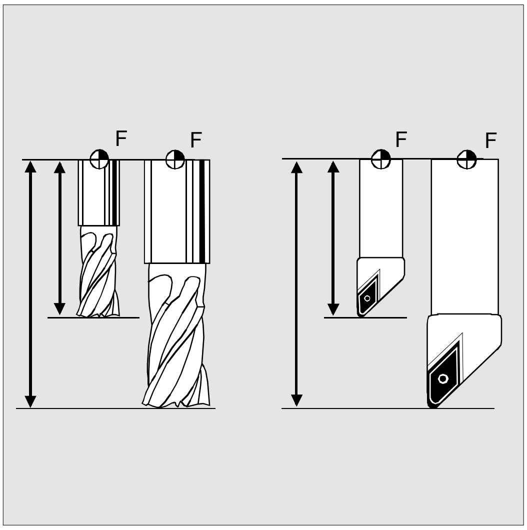
Values that have been entered once in the compensation memory are included in the processing for each tool numbered.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Tool length compensation

This value compensates for the differences in length between the tools used.

The tool length is the distance between the toolholder reference point and the tip of the tool. This length is measured and entered in the control together with definable wear values.
From this data, the control calculates the traversing movements in the infeed direction.



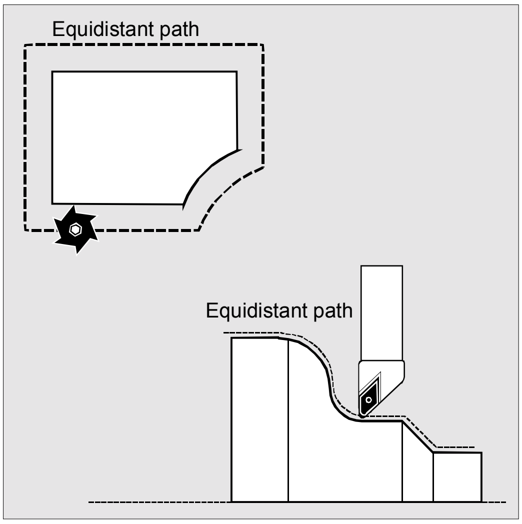
Additional notes

The compensation value of the tool length depends on the spatial orientation of the tool. See the section on tool orientation and tool length compensation for more information.

Tool radius compensation
The contour and tool path are not identical. The cutter or tool nose radius center must travel on a path that is equidistant from the contour.

To do this, the programmed tool center point path is displaced by an amount that depends on the radius and the direction of machining and such that the tool nose travels exactly along the desired contour.

The control fetches the required radii during program execution and calculates the tool path from these values.



The tool radius compensation acts correspondingly to the default CUTD or CUT2DF. You will find more information later in this chapter.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

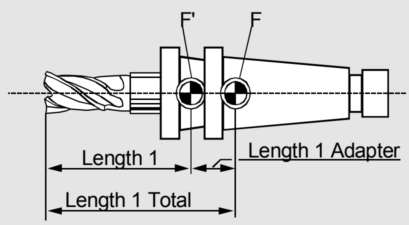
8.2 List of tool types

Coding of tool types for milling tools

Group with type 1xy (milling tool):

100	Milling tool according to CLDATA
110	Ball end mill (cylindrical die mill)
111	Ball end mill (conical die mill)
120	End mill (without fillet)
121	End mill (with fillet)
130	Angle head cutter (without fillet)
131	Angle head cutter (with fillet)
140	Facing tool
145	Thread cutter
150	Side mill
151	Saw
155	Bevel cutter (without corner rounding)
156	Bevel cutter (with corner rounding)
160	Drill and thread milling cutter

Entries in tool parameters	
DP1	1xy
DP3	Length 1-Geometry
DP6	Radius -Geometry
DP21	Length -Adapter



F-Reference point adapter (with tool inserted =toolholder reference point)

Wear values as required

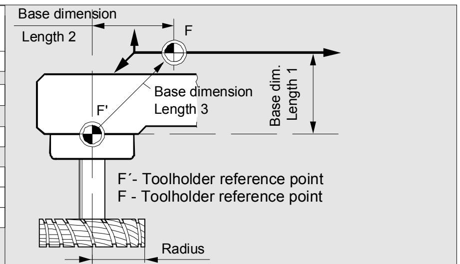
Set remaining values to 0

Effect	
G17:	Length 1 in Z Radius in X/Y
G18:	Length 1 in Y Radius in Z/X
G19:	Length 1 in X Radius in Y/Z

F´-Toolholder reference point

In SW 5 and higher, a fixed assignment is possible for G17, G18, G19 e.g. length1=X, length2=Z, length3=Y (see /FB1/ W1 Tool compens.)

Entries in tool parameters	
DP1	1xy
DP3	Length 1 - Geometry
DP6	Radius - Geometry
DP21	Length 1 - Base
DP22	Length 2 - Base
DP23	Length 3 - Base



Wear values as required

Set remaining values to 0

Effect		
G17:	Length 1 in Z Length 2 in Y Length 3 in X Radius in Y/Z	Z, Y, X
G18:	Length 1 in Y Length 2 in X Length 3 in Z Radius in X/Y	Y, X, Z
G19:	Length 1 in X Length 2 in Z Length 3 in Y Radius in Z/X	X, Z, Y

In SW 5 and higher, a fixed assignment is possible for G17, G18, G19
e.g. length 1 = X, length 2 = Y, length 3 = Z (see /FB1/W1 Tool compensation)

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Coding of tool types for drills

Group type 2xy (drills):

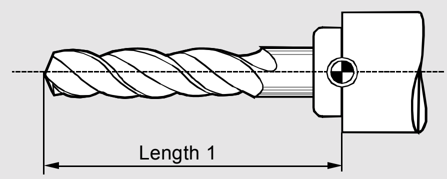
200 Twist drill
205 Solid drill
210 Boring rod
220 Center drill
230 Countersink
231 Counterbore
240 Tapping regular thread
241 Tapping fine thread
242 Tapping Whitworth thread
250 Reamer

Entries in tool parameters	
DP1	2xy
DP3	Length 1

Wear values as required

Set remaining values to 0

Effect	
G17:	Length 1 in Z
G18:	Length 1 in Y
G19:	Length 1 in X



F - toolholder reference point

Coding of tool types for grinding tools

Group type 4xy (grinding tools):

400 Surface grinding wheel
401 Surface grinding wheel with monitoring
403 Surface grinding wheel with monitoring without tool base dimension for grinding wheel peripheral speed GWPS
410 Facing wheel
411 Facing wheel with monitoring
413 Facing wheel with monitoring without tool base dimension for grinding wheel peripheral speed GWPS
490 Dresser

Entries in tool parameters	
DP1	403
DP2	Position*)
DP3	Length 1
DP4	Length 2
DP6	Radius

*) Tool nose position

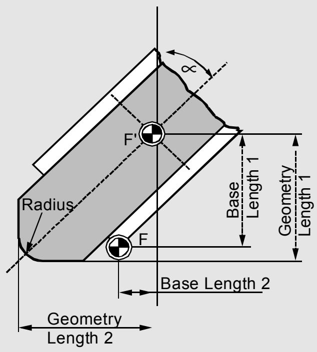
Wear values as required

Set remaining values to 0

Effect	
G17:	Length 1 in Y Length 2 in X Radius in X/Y
G18:	Length 1 in X Length 2 in Z Radius in Z/X
G19:	Length 1 in Z Length 2 in Y Radius in Y/Z

TPG1	Spindle number
TPG2	Chaining rule
TPG3	Minimum wheel radius
TPG4	Minimum wheel width
TPG5	Current wheel width
TPG6	Maximum speed
TPG7	Max. peripheral speed
TPG8	Angle of the inclined wheel
TPG9	Parameter no. f. radius calculation

F - Toolholder reference point

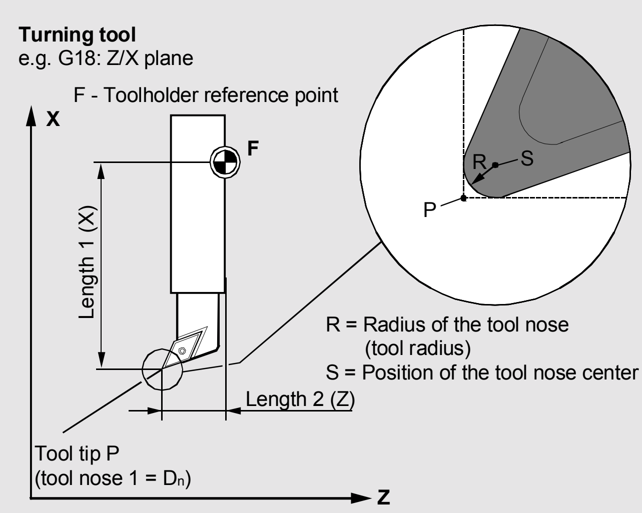


840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

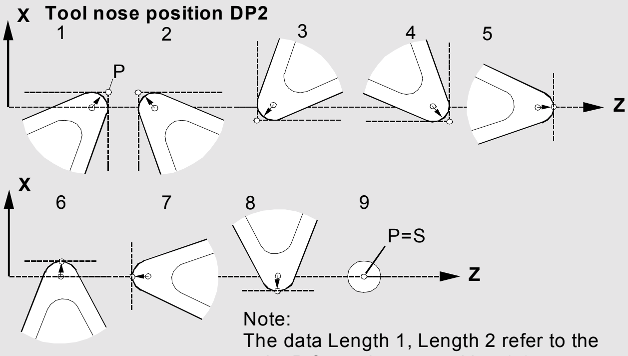
Breakdown of tool types for turning tools

Group type 5xy (turning tools):

500 Roughing tool
510 Finishing tool
520 Grooving tool
530 Parting tool
540 Threading tool
550 Button tool / form cutting tool
580 Probe with parameter tool point direction



The tool parameter DP2 specifies the length of the tool nose.
Position value 1 to 9 possible.



Note:
The data Length 1, Length 2 refer to the point P for tool nose position 1-8;
but S for position 9 (S=P)

Entries in tool parameters	
DP1	5xy
DP2	1...9
DP3	Length 1
DP4	Length 2
DP6	Radius

Wear values as required

Set remaining values to 0.

Effect	
G17	Length 1 in Y Length 2 in X
G18	Length 1 in X Length 2 in Z
G19	Length 1 in Z Length 2 in Y

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

- Chain rule
  The tool length offsets "Geometry", "Wear" and "Base dimension" can be chained for the left and right wheel correction in each case, i.e. if the length offsets for the left tool edge are altered, the values for the right edge are automatically entered and vice versa. Please refer to Description of Functions /FB II/, W4 "Grinding".

Explanation of tool types

Group type 7xy (special tools)

700 Slotting saw

710 3D probe

730 Fixed stop

Slotting saw

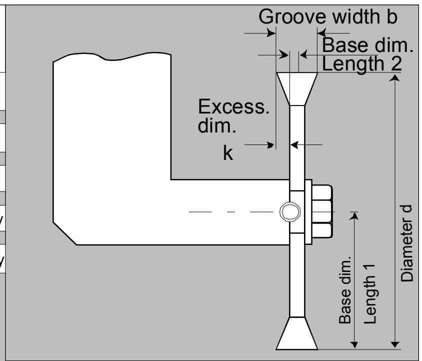
Group with type:

700 Slotting saw

Entries in tool parameters	
DP3	Length 1 - Base dim.
DP4	Length 2 - Base dim.
DP6	Diameter - Geometry
DP7	Groove width-Geometry
DP8	Excess. dim.-Geometry

Wear values as required

Set remaining values to 0



Effect		
G17:	Half diameter (L1) in X Excess dim. in (L2) Y Saw blade in (R) X/Y	Plane selection 1st-2nd axis (X-Y)
G18:	Half diameter (L1) in Y Excess dim. in (L2) X Saw blade in (R) Z/X	Plane selection 1st-2nd axis (X-Z)
G19:	Half diameter (L1) in Z Excess dim. in (L2) Z Saw blade in (R) Y/Z	Plane selection 1st-2nd axis (Y-Z)

Additional notes

The parameters for the tool types are described in:
References: FB, W1 Tool Compensation and in help displays of the controller

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.3 Tool selection/tool call T

8.3.1 Tool change with M06 (mill)

Programming

```
Tx or
T=x or
Ty=X
T0
M06
```

Explanation of the parameters

`Tx or T=x or Ty=x`	Tool selection with T No.
`x`	x stands for T No.: 0-32000
`T0`	Tool deselection
`M06`	Tool change, then tool T… and tool offset D are active
	Number of tools: 600, as of SW 5: 1200 (depending on the machine manufacturer's configuration)

Function

Tool selection takes place when the T word is programmed.

1. Tool selection without TOOLMAN
   - Free selection of D No. (flat D No.) relative to cutting edges

T...	[8-digit]

1D	2D	3D	• • •	D32000

   - Tabular D No.: D1 ... D8

T1	1D	2D	3D	• • •	D8
T2	1D				
T3	1D				
T6	1D	2D	3D		
T9	1D	2D			
	1D		3D		
•					
•					
T...	1D	2D			

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

2. Tool selection with tool management
   - Free selection of D No. (flat D No.) relative to cutting edges
   - Fixed assignment of D No. to cutting edge

The tool is not active until M06 (incl. corresponding D No.).

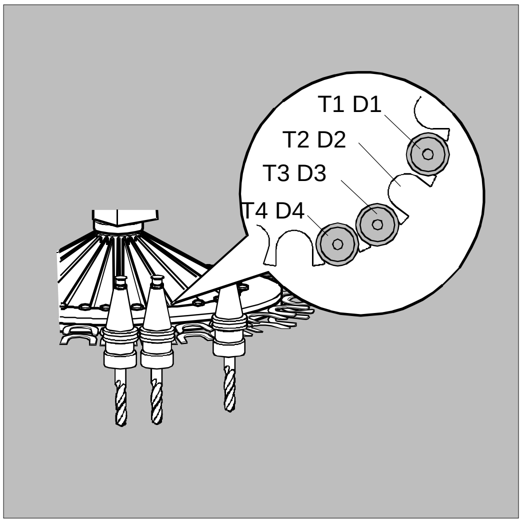
Machine manufacturer (MH 8.1)

The effect of the T number call is defined in machine data. See machine manufacturer's configuration.

Explanation

SW 4 and higher

The free selection of the D No. "flat D number structure" is applied when tool management is implemented outside the NCK. In this case, the D numbers are created with the corresponding tool compensation blocks without assignment to tools.
T can continue to be programmed in the parts program. However, this T has no reference to the programmed D number.
Example:
Circular magazine with 12 locations and 12 single-edge tools.



Machine manufacturer (MH 8.5)

T can or cannot be programmed in the parts program, depending on the setting in MD 18102.

Sequence

Creating a new D number

Creating a new D number with the associated tool compensation blocks is performed exactly as for the normal D number via tool parameters $TC_DP1 to $TC_DP25. The T number need not be entered any more.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Machine manufacturer (MH 8.6)

The type of D number management is defined in the machine data. There are two settings available for programming D numbers in the "flat D number structure":

- Flat D number structure with direct programming
- Flat D number structure with indirect programming (SW5 and higher).

8.3.2 Tool change with T command (rotate)

Programming

`Tx` or `T=x`
or `Ty=X`
`T0`

Explanation of the parameters

`Tx or T=x or Ty=x`	Tool selection with T No. including tool change (active tool), tool offset is active
`x`	x stands for T No.: 0-32000
`T0`	Tool deselection
	Number of tools: 600, as of SW 5: 1200 (depending on the machine manufacturer's configuration)

Function

A direct tool change takes place when the T word is programmed.

1. Tool selection without TOOLMAN
   - Free selection of D No. (flat D No.) relative to cutting edges
   - Tabular D No.: D1 ... D8

2. Tool selection with tool management
   - Free selection of D No. (flat D No.) relative to cutting edges
   - Fixed assignment of D No. to the cutting edges

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Machine manufacturer (MH 8.1)

The effect of the T number call is defined in machine data. See machine manufacturer's configuration.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.4 Tool offset D

Programming

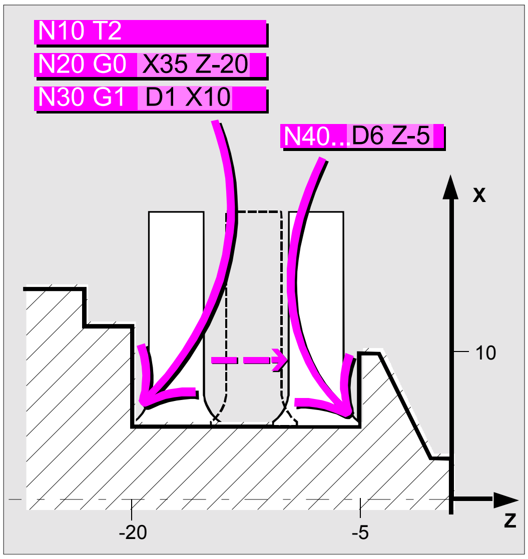
```
D...
D0
```

Explanation of the parameters

`Dx`	Tool offset number: without T 1–8 or with T (as of SW5) 1–12
`x`	x stands for the D No.: 0-32000
`D0`	Deselect tool offset, no offsets active

Function

It is possible to assign between 1 and 8 (12) tool noses with different tool compensation blocks to a specific tool. This allows you to define various cutting edges for **one** tool, which you can call as required in the NC program. Different offset values could be used, for example, for the left and right cutting edge of a grooving tool. When D is called, the tool length compensation for a specific tool nose is activated. When D0 is programmed, offsets for the tool are ineffective. If no D word is programmed, the default setting from the machine data is valid for tool change. Tool length compensations take immediate effect if the D number is programmed.
A tool radius offset must also be activated by G41/G42.



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Machine manufacturer (MH 8.10)

Default setting by machine manufacturer, e.g.: D1, i.e. without D programming, D1 is activated/selected when the tool is changed (M06).
The tools are activated with T programming (see machine manufacturer's specifications).
The compensation is performed with the first programmed traversing of the respective tool length compensation axis.

The required D number must always be programmed before the tool length compensation can be selected. The tool length compensation is also effective if the compensation is set in the machine data.

Working without tool offsets, D0
D0 is the default setting after start-up of the control. If you do not enter a D number you work without a tool offset.

The modified values do not become active until the T or D number is next programmed.

Programming example
(turning: tool change with T command)

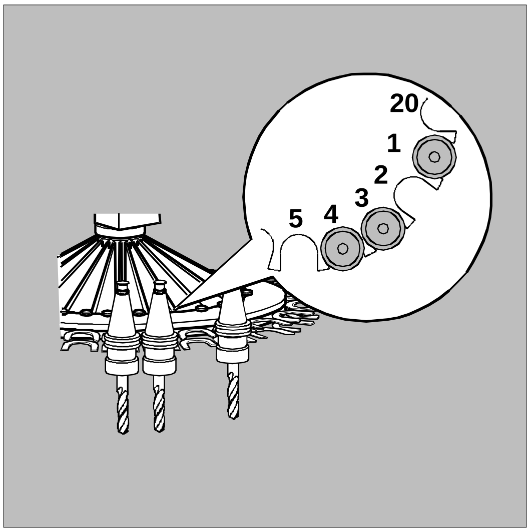
`N10 T1 D1`	Tool T1 is replaced and activated with associated D1
`N11 G0 X... Z...`	The length offsets are traversed
`N50 T4 D2`	Load tool T4, D2 from T4 is active
`...`	
`N70 G0 Z... D1`	Other cutting edge D1 is activated for tool T4

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.5 Tool selection T with tool management

Example

A magazine has locations 1 to 20:
Location 1 is taken up by drilling tool, duplo no.=1, T15, disabled
Location 2 is not occupied
Location 3 is taken up by drilling tool, duplo no.=2, T10, enabled
Location 4 is taken up by drilling tool, duplo no.=3, T1, active
Locations 5 to 20 are not occupied



1. Programming of T1 or T=1:
   Location number 1 of the magazine associated with the toolholder is selected.
2. Identifier "Drill" of tool in location is determined.
   The selection procedure is completed.
3. It is followed by the tool change procedure:
   On completion of tool search strategy "Take the first available tool from group", T10 is loaded because T15 is disabled.
4. On completion of tool search strategy "Take the first tool with status "active" from group", T1 is loaded.

8.5.1 Turning machine with circular magazine

Programming

The following procedure usually applies:

T = slot or T = identifier
T triggers the tool change

`D...` Tool offset number: 1...32000 (max., see machine manufacturer's specifications)
D0: no offset active!

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

The following procedure usually applies:

T = slot,
T triggers the tool change

D = offsets 1 to n (n ≤ 32000)

If the relative D No. structure with internal reference to the associated tools is used, replacement tool management and monitoring function are possible.

Note:

When calling the tool, the

- tool offset values stored under a D number must be activated.
- the appropriate working plane (system setting: G18) must be programmed. This ensures that the length compensation is assigned to the correct axis.

If the selected magazine location is not occupied in a selected magazine location, the tool command has the same effect as T0. Selection of the unoccupied magazine location can be used for positioning the empty location.

Machine manufacturer (MH 8.2)

Tool management: See machine manufacturer's configuration.

8.5.2 Milling machine with chain magazine

Sequence

The following procedure usually applies:

T = "Ident" or T = No. or T=Duplo no.,
M06 triggers the tool change

D = offset 1 to nth edge no.
(n ≤ 8, SW 5:12 and higher)

Selection:

- With integrated tool management (inside NC) relative D no. structure with internal reference to the associated tools (e.g. replacement tool management and monitoring function possible)

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

- Without integrated tool management (outside NC) flat D no. structure without internal reference to associated tools

Machine manufacturer (MH 8.3)

Tool management: See machine manufacturer's configuration.

Tool magazine

The T number preselects the tool, e.g. positioning the magazine to the tool change position. The actual tool change is triggered by M6. The M number for the tool change is set via the machine data. (See also the section on miscellaneous functions M). Only then do the new tool offsets apply.

Note:

When calling the tool, the

- Tool offset values stored under a D number must be activated.
- The appropriate working plane (system setting: G17) must be programmed. This ensures that the length compensation is assigned to the correct axis.

If the selected magazine location is not occupied in a selected magazine location, the tool command has the same effect as T0. Selection of the unoccupied magazine location can be used for positioning the empty location.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.6 Tool offset call D with tool management

Machine manufacturer (MH 8.4)

See machine manufacturer's configuration.

8.6.1 Turning machine with circular magazine

Programming

The following procedure usually applies:

`T` = slot or `T` = identifier or T=Duplo no.,
T triggers the tool change

`D...` Tool offset number: 1...32000 (max., see machine manufacturer's specifications)
D0: no offset active!

Direct, absolute programming

Programming is performed with the D number structure. The compensation blocks to be used are called directly via their D number.
Assignment of the D number to a specific tool does not take place in the NC kernel.

Machine manufacturer (MH 8.7)

Direct programming is defined by MD.

Programming example

`$MC_TOOL_CHANGE_MODE=0`	MD20270 CUTTING_EDGE_DEFAULT = 1
`...`	
`D92`	Traverse with tool offsets from D92
`...`	
`T17`	Select T17, traverse with tool offsets from D92
`...`	
`D16`	Traverse with tool offsets from D16
`...`	
`D32000`	Traverse with tool offsets from D32000
`...`	
`T29000500`	Select T29000500, traverse with tool offsets from D32000
`...`	
`D1`	Traverse with tool offsets from D1

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.6.2 Milling machine with chain magazine

Sequence

The following procedure usually applies:
T= identifier or T= identifier or
T= Duplo no.,
M06 triggers the tool change
D = offset 1 to nth edge no.
(n ≤ 8 or 12, SW 5 and higher)
Selection:

- with integrated tool management (inside NC) relative D no. structure with internal reference to the associated tools (e.g. replacement tool management and monitoring function possible)
- without integrated tool management (outside NC) flat D no. structure without internal reference to associated tools

Machine manufacturer (MH 8.9)

Tool management: See machine manufacturer's specifications

Function

It is possible to assign between 1 and 8 (12) tool noses with different tool compensation blocks to a specific tool. When D is called, the tool length compensation for a specific tool nose is activated. When D0 is programmed, offsets for the tool are ineffective. If no D word is programmed, the default setting from the machine data is valid for tool change.
Tool length compensations take immediate effect if the D number is programmed.
A tool radius offset must also be activated by G41/G42.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.7 Make active tool offset operative immediately

Function

MD $MM_ACTIVATE_SEL_USER_DATA can be used to define that the active tool offset can be activated immediately if the parts program is in "stop" mode.

/FB/, Description of Functions, Fundamentals, K2 Axes, Coordinate Systems...

Danger

The offset is backed out the next time the parts program is started.

840D
NCU 571

840D
NCU 572
NCU 573

810D

840Di

8.8 Tool radius compensation, G40, G41, G42

Programming

```
G40
G41
G42
OFFN=
```

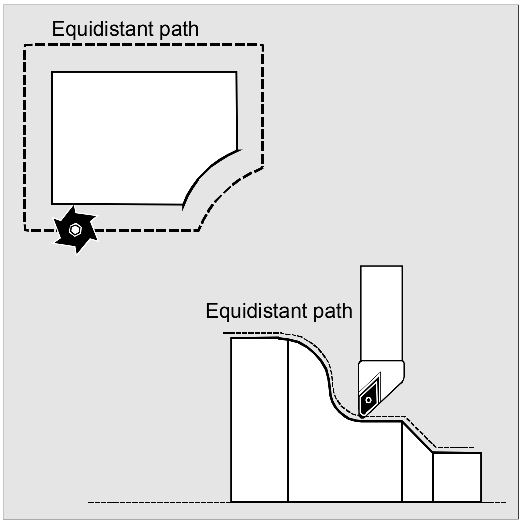
Explanation of the commands

G40	Deactivate tool radius compensation
G41	Activate tool radius compensation; tool operates in machining direction to the **left** of the contour.
G42	Activate tool radius compensation, tool operates in machining direction to the **right** of the contour.
OFFN=	Allowance on the programmed contour (normal contour offset)

Function

When tool radius compensation is active, the control automatically calculates the equidistant tool paths for different tools.

You can generate equidistant paths with OFFN, e.g. for rough-finishing.

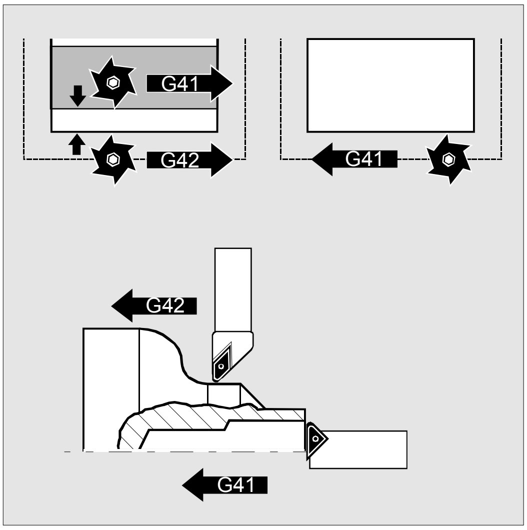


8.8 Tool radius compensation, G40, G41, G42

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Sequence

The control requires the following information in order to calculate the tool paths:



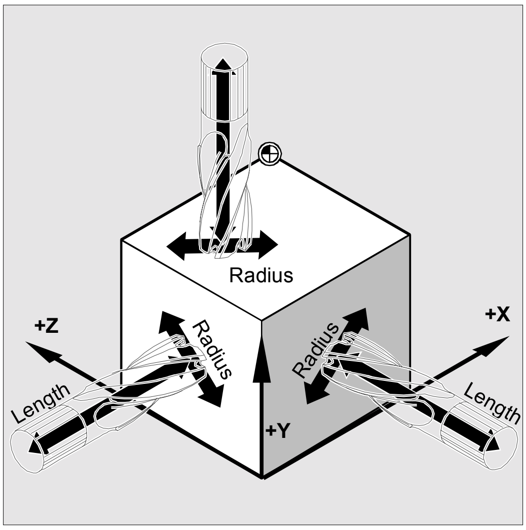
1. Tool number T/edge number D
Where appropriate, a tool offset number D is also required. The distance between the tool path and the workpiece contour is calculated from the cutter and tool edge radii and the tool point direction parameters.
With flat D number structure it is only necessary to program the D number.

2. Direction of machining G41, G42
From this information, the control detects the direction in which the tool path is to be displaced.

Note:
A negative offset value is the same as a change of offset side (G41, G42).

3. Working plane G17 toG19
From this information, the control detects the plane and therefore the axis directions for compensation.

Milling:



Example: Milling cutters
`N10 G17 G41 …`
The tool radius compensation is performed in the X/Y plane, the tool length compensation is performed in the Z direction.

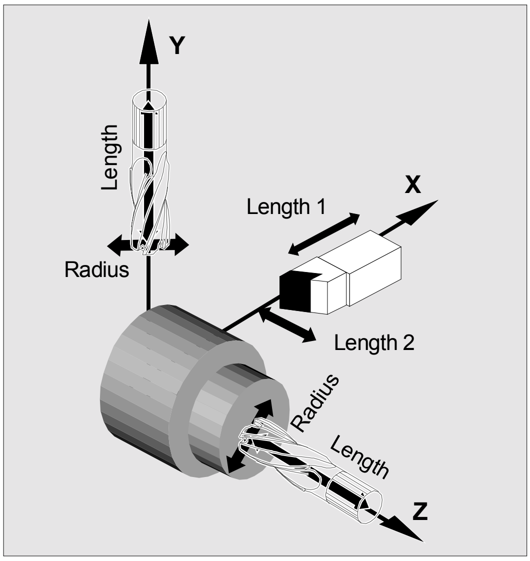
Note:
On 2-axis machines, the tool radius compensation is only possible in "real" planes, in general with G18 (see tool length compensation table).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Tool length compensation

The wear parameter assigned to the diameter axis on tool selection can be defined as the diameter value (MD). This assignment is not automatically altered when the plane is subsequently changed. To do this, the tool must be selected again after the plane has been changed.

Turning:



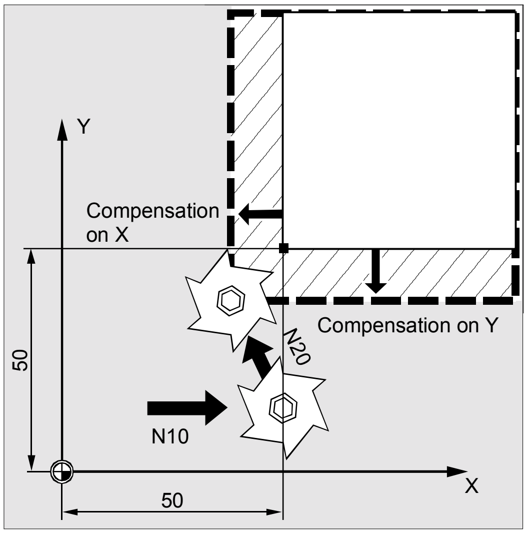
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Activation/deactivation of tool radius compensation

A travel command must be programmed with G0 or G1 in an NC block with G40, G41 or G42. This travel command must specify at least one axis in the selected working plane.

If you only specify one axis on activation, the last position on the second axis is added automatically and traversed with **both** axes.

Milling:

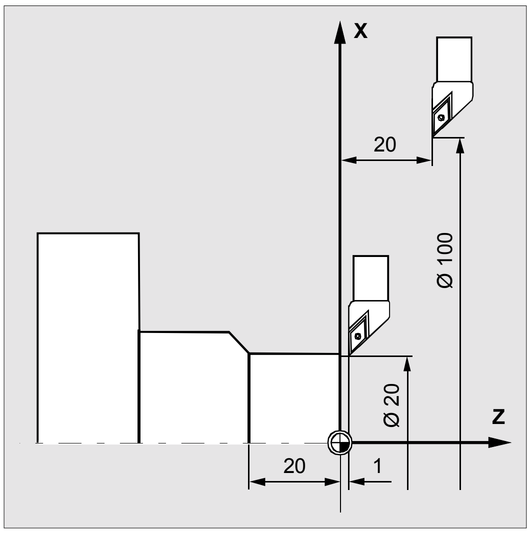


Example:

```
N10 G0 X50 T1 D1
N20 G1 G41 Y50 F200
N30 Y100
```

Only tool length compensation is activated in block N10. X50 is approached without compensation.
In block N20, the radius compensation is activated, point X50/Y50 is approached with compensation.

Turning:



Example:

```
N20 T1 D1
N30 G0 X100 Z20
N40 G42 X20 Z1
N50 G1 Z-20 F0.2
```

Only tool length compensation is activated in block N20. X100 Z20 is approached without compensation in block N30.
In block N40, the radius compensation is activated, point X20/Z1 is approached with compensation.

Using NORM and KONT you can determine the tool path for activation/deactivation of compensation mode (see Section 8.10 Contour approach and retraction, NORM, KONT, G450, G451).

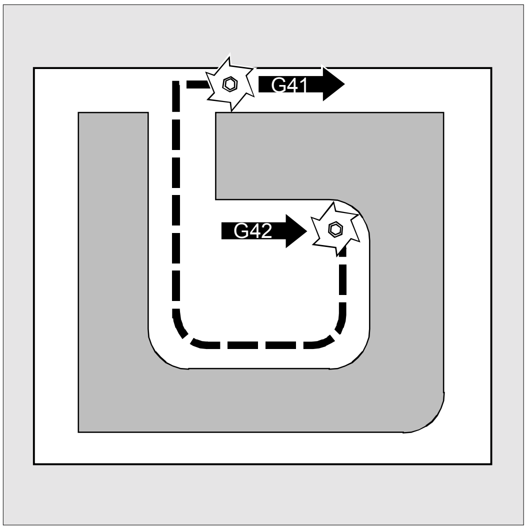
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Changing the direction of compensation

G41/G42, G42/G41 can be programmed without an intermediate G40.

Changing the working plane

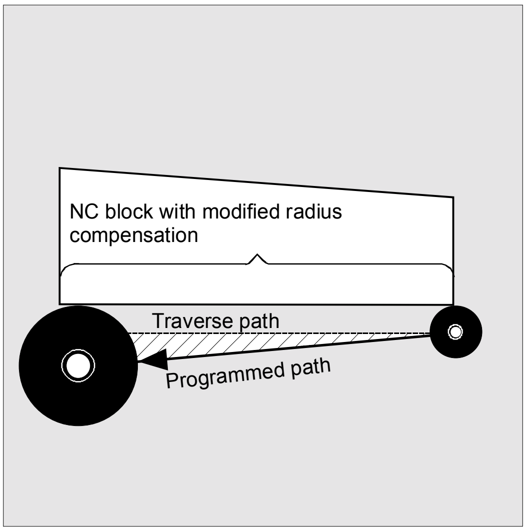
It is not possible to change the working plane G17 to G19 when G41/G42 is active.



Changing the offset number D

The offset number D can be changed in compensation mode.

A modified tool radius is active with effect from the block in which the new D number is programmed.



The radius change or compensation movement is performed across the entire block and only reaches the new equidistance at the programmed end point.

With linear movements, the tool travels along an inclined path between the starting point and end point; with circular interpolation spiral movements are produced.

Changing the tool radius

This can be achieved, for example, using system variables. The execution is the same as for changes in the D number.

The modified values only take effect the next time T or D is programmed. The change only applies with effect from the next block.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

During compensation mode

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

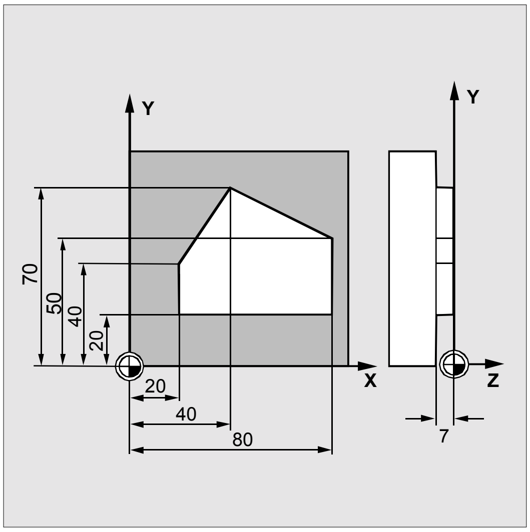
Machine manufacturer (MH 8.14)

The number of consecutive blocks or M commands can be set in machine data 20250 (see machine manufacturer).
A block with a path distance of zero also counts as an interruption!

Programming example

The "conventional" approach:
Call tool, load tool, activate working plane and tool radius compensation.

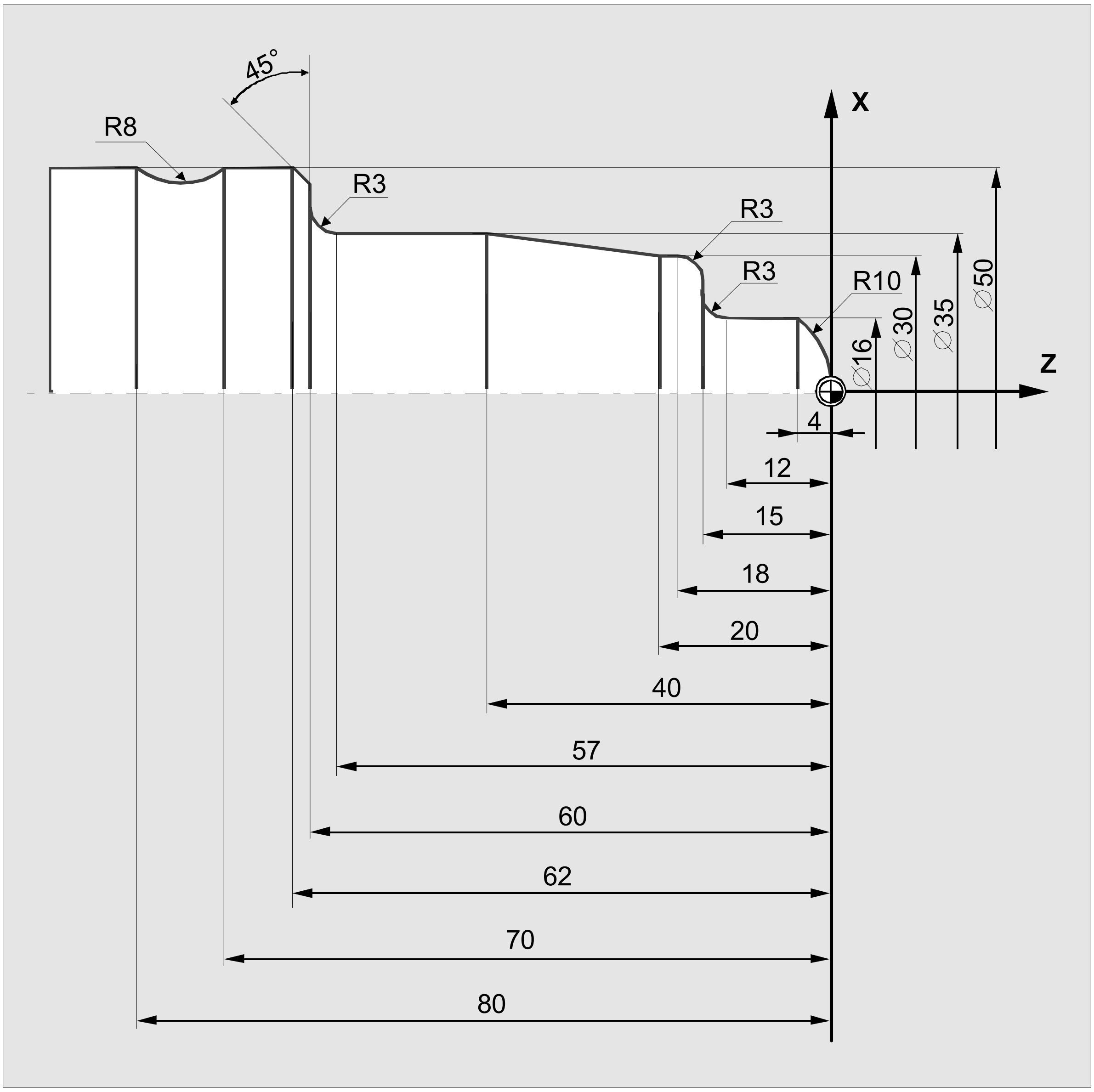
Milling:



`N10 G0 Z100`	Retract to tool change point
`N20 G17 T1 M6`	Tool change
`N30 G0 X0 Y0 Z1 M3 S300 D1`	Call tool offset values, select length compensation
`N40 Z-7 F500`	Tool infeed
`N50 G41 X20 Y20`	Approach tool radius compensation, tool operates on left-hand side of contour
`N60 Y40`	Mill contour
`N70 X40 Y70`	
`N80 X80 Y50`	
`N90 Y20`	
`N100 X20`	
`N110 G40 G0 Z100 M30`	Retract tool, end of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Turning:



%_N_1001_MPF		;Program name
N5	G0 G53 X280 Z380 D0	;Start point
N10	TRANS X0 Z250	;Zero offset
N15	LIMS=4000	;Speed limitation (G96)
N20	G96 S250 M3	;Select constant feed
N25	G90 T1 D1 M8	;Select tool and offset
N30	G0 G42 X-1.5 Z1	;Activate tool with tool radius compensation
N35	G1 X0 Z0 F0.25	
N40	G3 X16 Z-4 I0 K-10	;Rotate radius 10
N45	G1 Z-12	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

N50	G2 X22 Z-15 CR=3	;Rotate radius 3
N55	G1 X24	
N60	G3 X30 Z-18 I0 K-3	;Rotate radius 3
N65	G1 Z-20	
N70	X35 Z-40	
N75	Z-57	
N80	G2 X41 Z-60 CR=3	;Rotate radius 3
N85	G1 X46	
N90	X52 Z-63	
N95	G0 G40 G97 X100 Z50 M9	;Deselect tool radius compensation and approach tool change location
N100	T2 D2	;Call up tool and select offset
N105	G96 S210 M3	;Select constant cutting speed
N110	G0 G42 X50 Z-60 M8	;Activate tool with tool radius compensation
N115	G1 Z-70 F0.12	;Rotate diameter 50
N120	G2 X50 Z-80 I6.245 K-5	;Rotate radius 8
N125	G0 G40 X100 Z50 M9	;Retract tool and deselect tool radius compensation
N130	G0 G53 X280 Z380 D0 M5	;Move to tool change location
N135	M30	;End of program

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.9 Approach and retract from contour, NORM, KONT, G450, G451

Programming

```
NORM
KONTG450
KONT G451
```

Explanation of the parameters

`NORM`	The tool travels directly in a straight line and is positioned perpendicular to the contour point
`KONT`	The tool traverses the contour point according to the programmed corner behavior G450 or G451

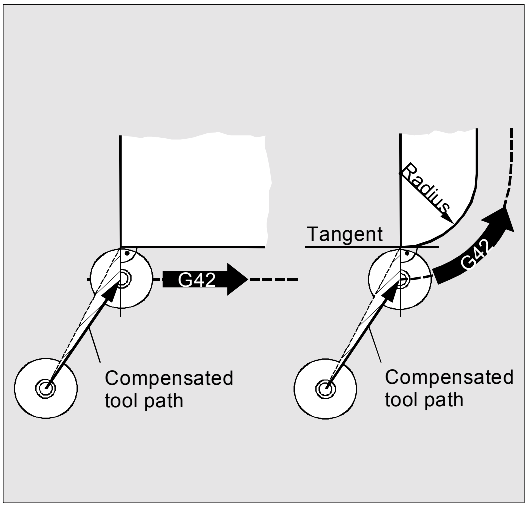
Function

You can use these functions to adapt the approach and retraction paths, for example, according to the desired contour or shape of the blanks.

Sequence

Direct approach to perpendicular position, G41, G42, NORM
The tool travels in a straight line directly to the contour and is positioned perpendicular to the path tangent at the starting point.

Selection of the approach point
When NORM is active, the tool travels directly to the compensated starting position irrespective of the approach angle programmed for the travel movement (see diagram).

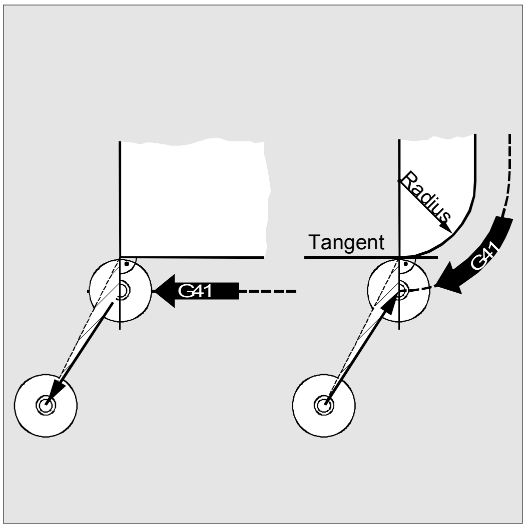


Machine manufacturer (MH 8.15)

Initial status see machine manufacturer's specifications.

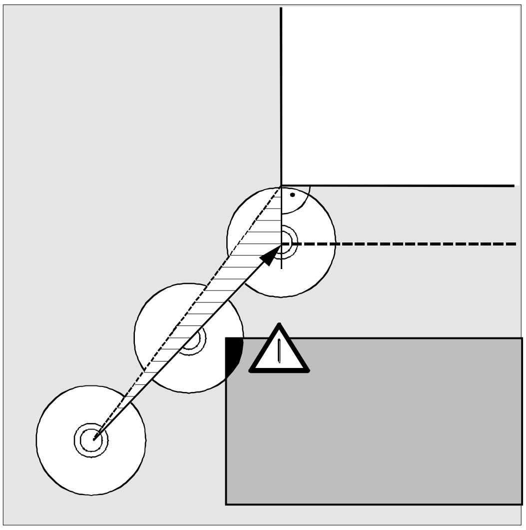
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Deactivate compensation mode, G40, NORM
The tool is positioned perpendicular to the last compensated path end point and then travels directly in a straight line to the next uncompensated position, e.g. to the tool change location.



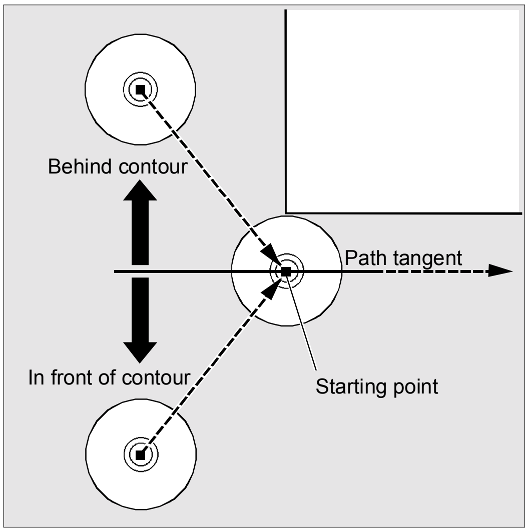
Choosing the retraction point
When NORM is active, the tool travels directly to the uncompensated position irrespective of the approach angle programmed for the travel movement (see diagram).

The following applies to approach and retraction movements:
You should make allowance for the modified angle of travel when programming in order to avoid collisions.



Travel round contour at starting point, G41, G42, KONT
Two cases are distinguished here:

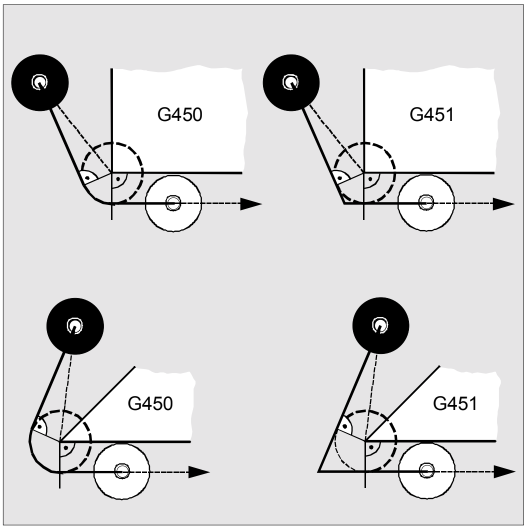
Starting point lies in front of the contour
The approach strategy is the same as with NORM. The path tangent at the starting point serves as a dividing line between the **front** and **rear** of the contour.



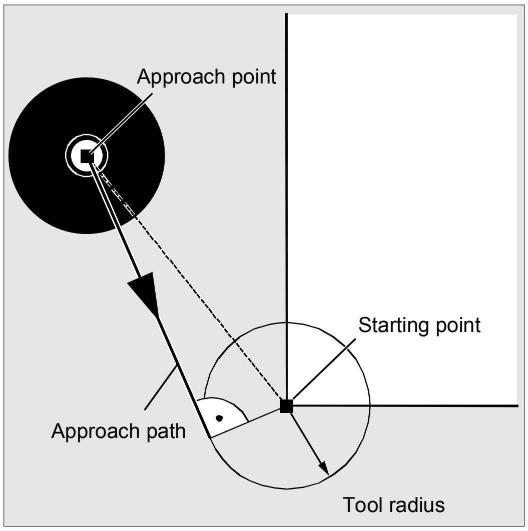
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Starting point lies behind the contour
The tool travels round the starting point either along a circular path or over the intersection of the equidistant paths depending on the programmed corner behavior G450/G451

The commands G450/G451 apply to the transition from the current block to the next block.



Generation of the approach path
In both cases (G450/G451), the following approach path is generated:
A straight line is drawn from the uncompensated approach point. This line is a tangent to a circle with circle radius = tool radius. The center point of the circle is on the starting point.



Deactivate compensation mode, G40, KONT
If the retraction point is located in front of the contour, the same retraction movement as for NORM applies.

If the retraction point is located behind the contour, the retraction movement is the reverse of the approach movement.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.10 Compensation at outside corners, G450, G451

Programming

```
G450 DISC=…
G451
```

Explanation of the parameters

`G450`	Transition circle: the tool travels around workpiece corners on a circular path with tool radius
`DISC=`	Flexible programming of the approach and retraction instruction. In steps of 1 from DISC=0 circle to DISC=100 intersection
`G451`	Intersection, the tool backs off from the workpiece corner

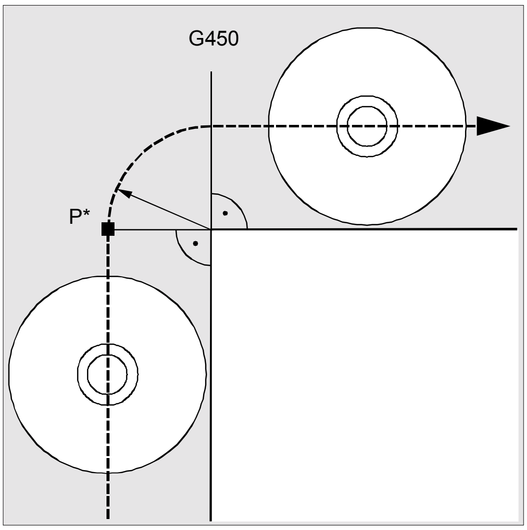
Function

G450/G451 defines the following:

On the one hand, the approach path for active KONT and the approach point behind the contour (see previous page).

On the other hand, the corrected tool path when traveling around outside corners.

Corner behavior, transition circle, G41, G42, G450



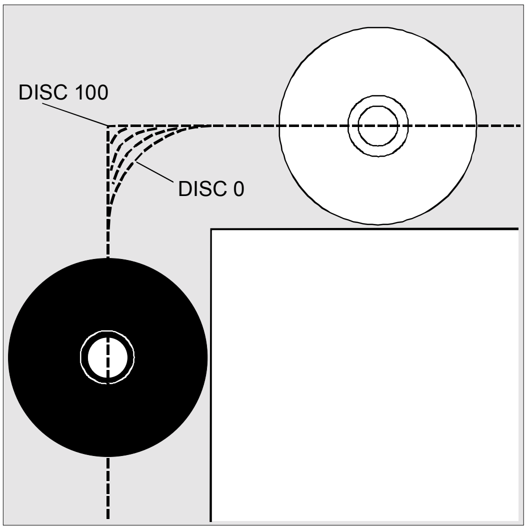
The tool center point travels around the workpiece corner across an arc with tool radius.

At intermediate point P*, the control executes instructions such as infeed movements or switching functions. These instructions are programmed in blocks inserted between the two blocks forming the corner.
The transition circle belongs to the next travel command with respect to the data.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Corner behavior, selectable transitions G41, G42, G450 DISC=…

DISC distorts the transition circle, thus creating sharp contour corners.



The values have the following meanings:
DISC=0 transition circle
DISC=100 intersection of the equidistant paths (theoretical value)
DISC is programmed in steps of 1.

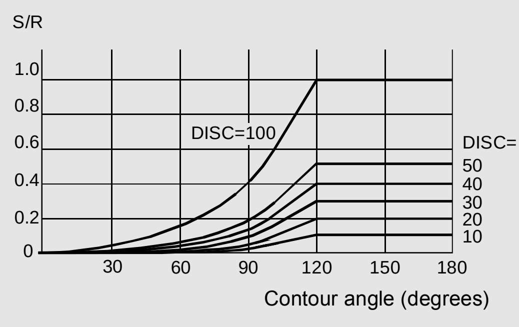
When DISC values greater than 0 are specified, intermediate circles are shown with a magnified height – the result is transition ellipses or parabolas or hyperbolas.

An upper limit can be defined in machine data – generally DISC=50.

DISC=… is effective only when G450 is called, but can be programmed in a preceding block without G450. Both commands are modal.

Path action, depending on DISC values and contour angle

Depending on the angle of the contour that is traversed, with acute contour angles and high DISC values the tool is lifted off the contour at the corners. With angles of 120° and more, the contour is traversed evenly (see adjacent table).

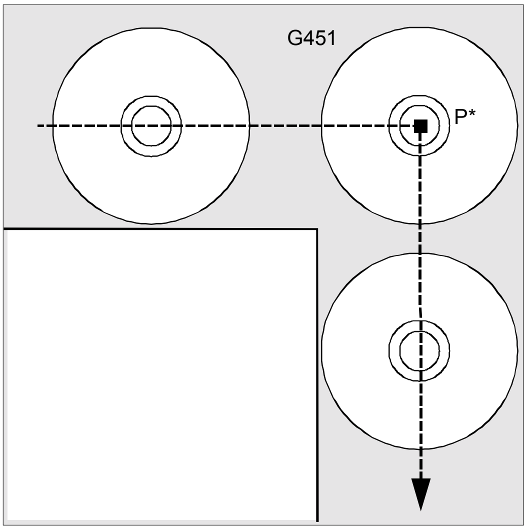


R - Tool radius S - Traversed overshoot
S/R - Scaled overshoot (with reference to the tool radius)

8.10 Compensation at outside corners, G450, G451

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

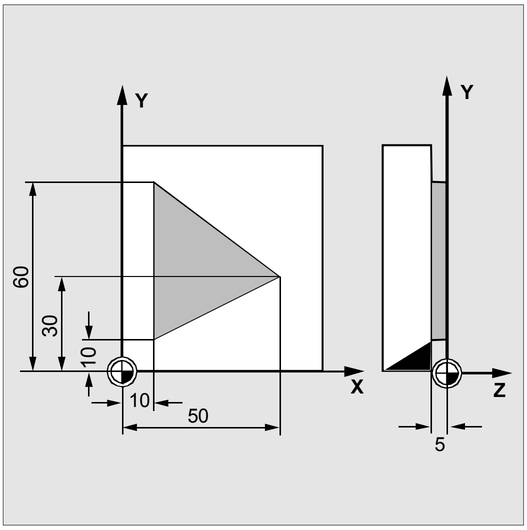
Corner behavior, intersection, G41, G42, G451



The tool approaches the intersection of the two equidistant which lie in the distance between the tool radius and the programmed contour. G451 applies only to circles and straight lines.
At intermediate point P*, the control executes instructions such as infeed movements or switching functions. These instructions are programmed in blocks inserted between the two blocks forming the corner.

Superfluous non-cutting tool paths can result from liftoff movements at acute contour angles.
A parameter can be used in the machine data to define automatic switchover to transition circle in such cases.

Programming example



In this example, a transition radius is inserted for all outside corners (progr. in block N30). This avoids the necessity of stopping the tool on the change of direction and making no cut.

`N10 G17 T1 G0 X35 Y0 Z0 F500`	Start conditions
`N20 G1 Z-5`	Tool infeed
`N30 G41 KONT G450 X10 Y10`	Activate compensation mode

`N40 Y60`
`N50 X50 Y30`
`N60 X10 Y10`
	Cut contour

`N80 G40 X-20 Y50`	Deactivate compensation mode, retract on transition circle
`N90 G0 Y100`	
`N100 X200 M30`	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.11 Smooth approach and retraction

Programming

```
G140 to G143, G147, G148
G247, G248, G347, G348G340, G341
DISR=..., DISCL=...FAD=...
```

Explanation of the parameters

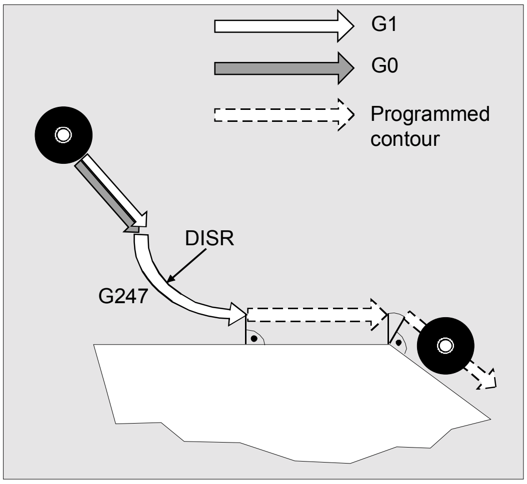
`G140`	Approach and retraction direction independent of the current compensation side (basic setting)
`G141`	Approach from the left or retraction to the left
`G142`	Approach from the right or retraction to the right
`G143`	Approach and retraction direction depends on the relative position of the start and end point with respect to the tangent direction
`G147`	Approach with a straight line
`G148`	Retraction with a straight line
`G247`	Approach with a quadrant
`G248`	Retraction with a quadrant
`G347`	Approach with a semicircle
`G348`	Retraction with a semicircle
`G340`	Approach and retraction in space (basic setting)
`G341`	Approach and retraction in the plane
`DISR`	• Approach and retraction with straight line (G147/G148) Distance from the mill edge to the start point of the contour • Approach and retraction with circles (G247, G347/G248, G348) Radius of the tool center point path Caution: In the case of REPOS with a semicircle, DISR is the diameter of the circle
`DISCL`	`DISCL=...` Distance from the end point of the fast infeed motion to the machining plane `DISCL=AC(...)` Specifies the absolute position of the end point of the fast infeed motion
`FAD`	Speed of the slow infeed motion `FAD=...` The programmed value acts in accordance with the G code of group 15 (feedrate; G93, G94 etc.) `FAD=PM(...)` The programmed value is interpreted as a linear feedrate (like G94) independently of the active G code of group 15 `FAD=PR(...)` The programmed value is interpreted as a revolutional feedrate (like G95) independently of the active G code of group 15

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Function

The smooth approach and retraction (SAR) function is used to achieve a tangential approach to the start point of a contour, regardless of the position of the start point.

The function is mainly used in conjunction with the tool radius offset, but is not mandatory.



Sequence

The approach and retraction motion consists of a maximum of 4 sub-movements

- Start point of the motion P_0
- Intermediate points P_1, P_2 and P_3
- End point P_4

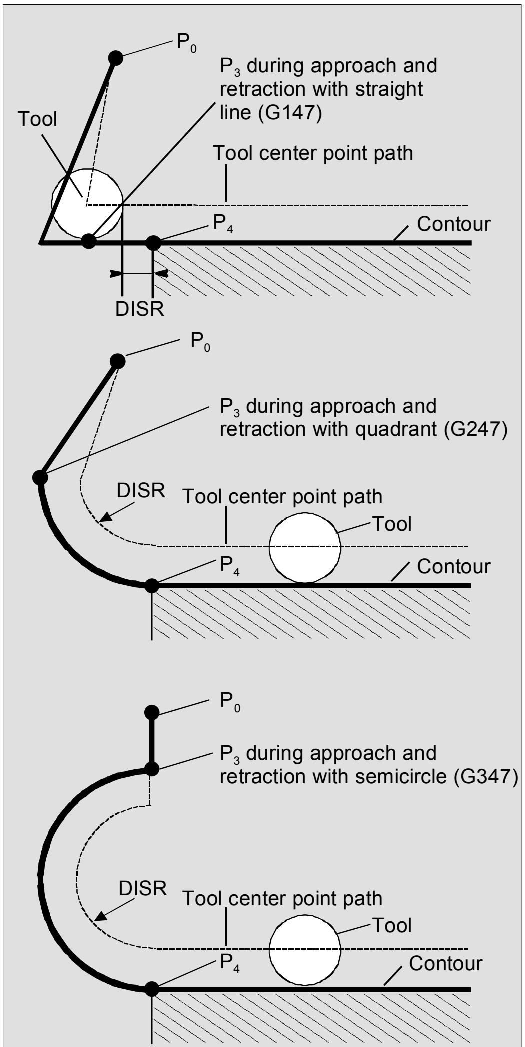
The points P_0, P_3 and P_4 are always defined. The intermediate points P_1 and P_2 may be omitted as appropriate to the parameter settings and geometrical conditions.

Selecting the approach and retraction contour
Using the appropriate G command, approach/retraction is possible with a straight line (G147, G148), a quadrant (G247, G248) or a semicircle (G347, G348).

Selecting the approach and retraction direction
Use the tool radius compensation (G140, basic setting) to determine the approach and retraction direction.

With positive tool radius:
G41 active → approach from left
G42 active → approach from right

G141, G142 and G143 provide further approach options.



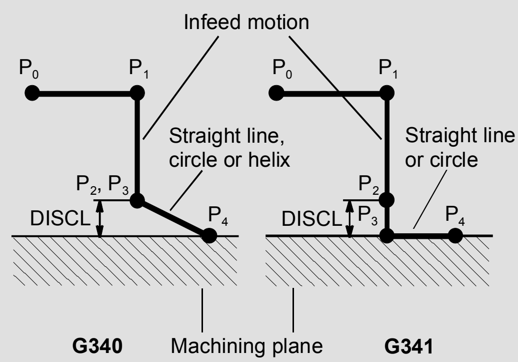
Approach and retraction motions, represented with intermediate point P_3 (tool radius compensation is simultaneously active)

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

The G codes are only significant when the approach contour is a quadrant or a semicircle.

Motion steps between start point and end point (G340 and G341)

The characteristic approach to P_0 to P_4 is shown in adjacent the figure.



Approach motion depending on G340/G341

In cases which include the position of the active plane G17 to G19 (circular plane, helical axis, infeed motion perpendicular to the active plane), any active rotating FRAME is taken into account.

Length of the approach straight line or radius for approach circles (DISR) (see figure for sequence)

- Approach/retract with straight line
  DISR specifies the distance of the cutter edge from the starting point of the contour, i.e. the length of the straight line when TRC is active is the sum of the tool radius and the programmed value of DISR. The tool radius is only considered if it is positive.
  The resultant line length must be positive, i.e. negative values for DISR are allowed provided that the absolute value of DISR is less than the tool radius.
- Approach/retraction with circles
  DISR indicates the radius of the tool center point path. If TRC is activated, a circle is produced with a radius that results in the tool center point path with the programmed radius.

Distance between the point and the machining plane (DISCL) (see figure for sequence)

If the position of the point P_2 is to be specified as an absolute value on the axis perpendicular to the circular plane, the value must be programmed in the form DISCL=AC(...).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

The following applies for DISCL=0:

- With G340: The entire approach motion consists of only two blocks (P_1, P_2 and P_3 are combined). The approach contour is obtained from P_1 to P_4.
- With G341: The entire approach motion consists of three blocks (P_2 and P_3 are combined). If P_0 and P_4 lie in the same plane, only two blocks are produced (infeed motion of P_1 to P_3 is omitted).

The point defined by DISCL is monitored to ensure that it is located between P_1 and P_3 , i.e. the sign must be identical for the component perpendicular to the machining plane in all motions which possess such a component.

On detection of a direction reversal, a tolerance defined by the machine data SAR_CLEARANCE_TOLERANCE is permitted.

Programming the end point P_4 for approach or P_0 for retraction

The end point is generally programmed with X... Y... Z....

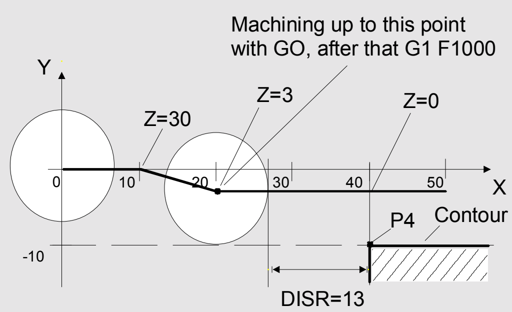
- Programming during approach
  - P_4 in the SAR block
  - P_4 is defined by the end point of the next traversing block

  Further blocks can be inserted between the SAR block and the next traversing block without moving the geometry axes.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Example:

```
$TC_DP1[1,1]=120   ;Milling tool T1/D1
$TC_DP6[1,1]=7     ;Tool with 7mm radius
N10 G90 G0 X0 Y0 Z30 D1 T1
N20 X10
N30 G41 G147 DISCL=3 DISR=13 Z=0 F1000
N40 G1 X40 Y-10
N50 G1 X50
...
...
```



N30/N40 can be replaced by:

1.

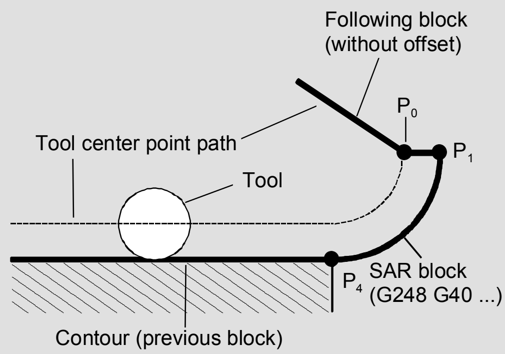
```
N30 G41 G147 DISCL=3 DISR=13 X40 Y-10
     Z0 F1000
```

or

2.

```
N30 G41 G147 DISCL=3 DISR=13 F1000
N40 G1 X40 Y-10 Z0
```

- Programming during retraction
  - For a SAR block without programmed geometry axis, the contour ends in P_2. The position in the axes that form the machining plane are obtained from the retraction contour. The axes components perpendicular to this are defined by DISCL. If DISCL=0, the entire motion is in the plane.
  - If only the axis perpendicular to the machining plane is programmed, the contour ends in P_1. The position of the other axes is obtained as described previously. If the SAR block is also the deactivation block of the TRC, an additional path from P_1 to P_0 is added so that there is no motion at the end of the contour on deactivation of the TRC.
  - If there is only one machining plane programmed, the missing 2nd axis is added modally taken from its last position in the previous block.



Retraction with SAR with TRC simultaneously deactivated

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Approach and retraction velocities

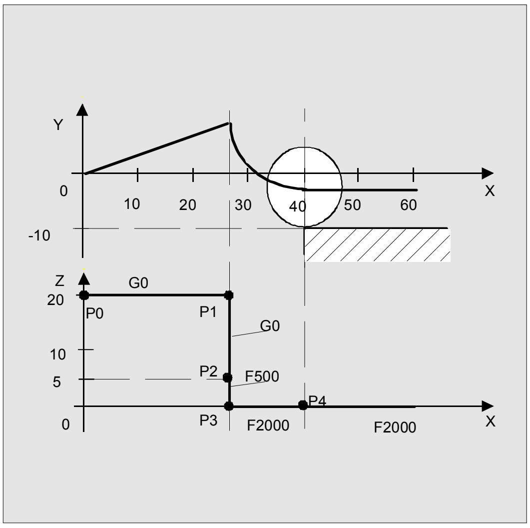
- Speed of the previous block (G0):
  All motions from P_0 up to P_2 are executed at this speed, i.e. the motion parallel to the machining plane and the part of the infeed motion up to the safety clearance.
- Programming with FAD:
  Specify the feedrate for
  - G341: infeed motion perpendicular to the machining plane from P_2 to P_3
  - G340: from point P_2 or P_3 to P_4

  If FAD is not programmed, this part of the contour is also traversed at the modally active speed of the previous block, if no F word is programmed in the SAR block.
- Programmed feedrate F:
  This feedrate is active from P_3 or P_2, if FAD is not programmed. If no F word is programmed in the SAR block, the speed of the previous block is active.

Example:

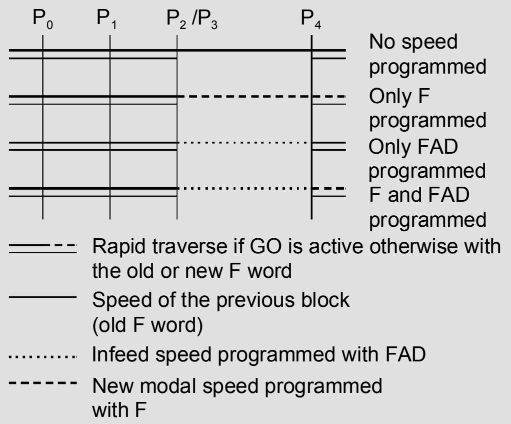
```
$TC_DP1[1,1]=120   ;Milling tool T1/D1
$TC_DP6[1,1]=7     ;Tool with 7mm radius

N10 G90 G0 X0 Y0 Z20 D1 T1
N20 G41 G341 G247 DISCL=AC(5) DISR=13
    FAD 500 X40 Y-10 Z=0 F200
N30 X50
N40 X60
...
```



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

During retraction, the rolls of the modally active feedrate from the previous block and the programmed feedrate value in the SAR block are changed round, i.e. the actual retraction contour is traversed with the old feedrate value and a new speed programmed with the F word applies from P_2 up to P_0.

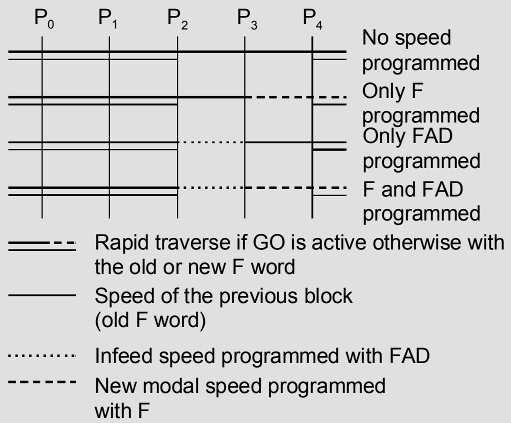


Speeds in the SAR sub-blocks during approach with G340

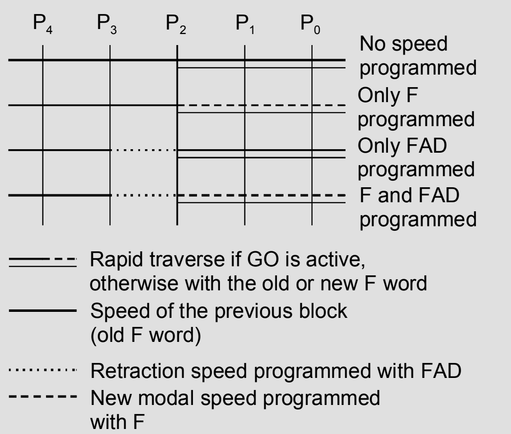
Reading positions

The points P_3 and P_4 can be read as system variables in the WCS during approach.

- $P_APR: read P_3 (start point)
- $P_AEP: read P_4 (contour start point)
- $P_APDV: read whether $P_APR and $P_AEP contain valid data



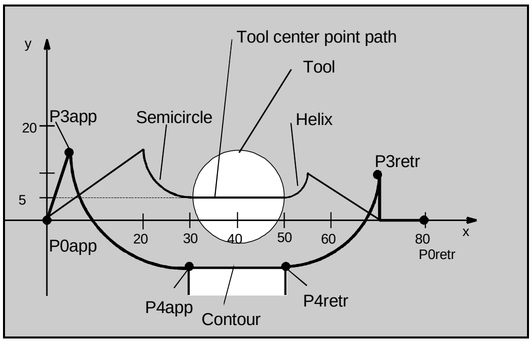
Speeds in the SAR sub-blocks during approach with G341



Speeds in the SAR sub-blocks during retraction

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example



- Smooth approach (block N20 activated)
- Approach motion with quadrant (G247)
- Approach direction not programmed, G140 is operative, i.e. TRC is active (G41)
- Contour offset OFFN=5 (N10)
- Current tool radius=10; thus the effective offset radius for TRC=15, the radius of the SAR contour=25, so that the radius of the tool center point path is then DISR=10
- The end point of the circle is obtained from N30, since only the Z position is programmed in N20
- Infeed motion
  – from Z20 to Z7 (DISCL=AC(7)) in rapid traverse
  – then to Z0 with FAD=200
  – approach circle in X-Y plane and following blocks with F1500 (for this speed to be active in the following blocks, the active G0 in N30 must be overwritten with G1; otherwise, the contour would be machined further with G0.)
- Smooth retraction (block N60 active)
- Retraction motion with quadrant (G248) and helix (G340)
- FAD not programmed, since irrelevant for G340
- Z=2 in the start point; Z=8 in the end point, since DISCL=6
- When DISR=5, the radius of SAR contour=20; that of the tool center point path=5
- Retraction motions from Z8 to Z20 and the motion parallel to X-Y plane to X70 Y0.

`$TC_DP1[1,1]=120`	Tool definition T1/D1	
`$TC_DP6[1,1]=10`	Radius	
`N10   G0 X0 Y0 Z20 G64 D1 T1 OFFN = 5`		(P_{0app})
`N20   G41 G247 G341 Z0 DISCL = AC(7)` `      DISR = 10 F1500 FAD=200`	Approach	(P_{3app})
`N30   G1 X30 Y-10`		(P_{4app})
`N40   X40 Z2`		
`N50   X50`		(P_{4retr})
`N60   G248 G340 X70 Y0 Z20 DISCL = 6` `      DISR = 5 G40 F10000`	Retraction	(P_{3retr})
`N70   X80 Y0`		(P_{0retr})
`N80   M30`		

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.11.1 Extension approach and retract: G461/G462 (SW 5 and higher)

Programming

```
G460
G461
G462
```

Explanation

G460	As before (activation of collision monitoring for approach and retraction block)
G461	Insertion of a circle in the TRC block, if no intersection point is possible, whose center point is at the end point of the uncorrected block and whose radius is equal to the tool radius.
G462	Insertion of a straight line in the TRC block if no intersection point is possible; the block is extended by its end tangent (default setting)

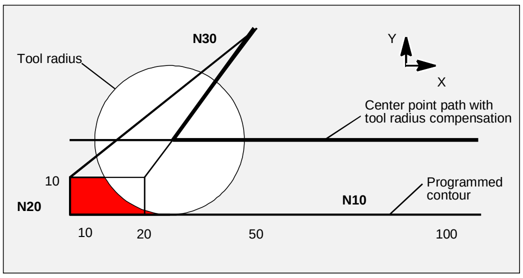
Function

In certain special geometrical situations, extended approach and retraction strategies, compared with the previous implementation, are required in order to activate or deactivate tool radius compensation (see figure below).

The following example describes only the situation for deactivation of tool radius compensation. The behavior for approach is the same.

Example:

```
G42 D1 T1       ; Tool radius 20mm
...
G1 X110 Y0
N10 X0
N20 Y10
N30 G40 X50 Y50
```



Retraction behavior with G460 (identical to behavior with SW 4.x and lower)

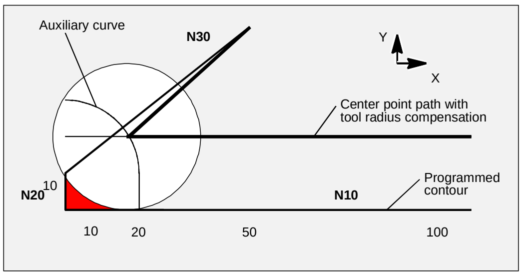
The last block with active tool radius compensation (N20) is so short that an intersection no longer exists between the offset curve and the preceding block (or a previous block) for the current tool radius.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

An attempt is therefore made to find an intersection between the offset curves of the following block and the preceding block, i.e. in the example between N10 and N30. The curve used for the retraction block is not a real offset curve, but is a straight line from the offset point at the end point of block N20 to the programmed end point of N30. If an intersection is found, it is approached. In this case, the shaded area in the figure is not machined, although this would be possible with the tool used.

G461

If no intersection is possible between the last TRC block and a preceding block, the offset curve of this block is extended with a circle whose center point lies at the end point of the uncorrected block and whose radius is equal to the tool radius.
The control attempts to cut this circle with one of the preceding blocks.



Retraction behavior with G461
(see example at the end of this section)

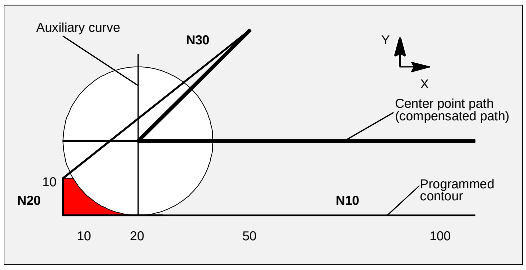
If CDOF is active (see Section 8.12), the search is aborted when an intersection is found, i.e. the system does not check whether further intersections with previous blocks exist.
If CDON is active, the search continues for further intersections after the first intersection is found.
An intersection point which is found in this way is the new end point of a preceding block and the start point of the deactivation block. The inserted circle is used exclusively to calculate the intersection and does not produce a traversing movement.

If no intersection is found, alarm 10751 (collision danger) is output.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

G462

If no intersection is possible between the last TRC block and a preceding block, a straight line is inserted, on retraction with G462 (initial setting), at the end point of the last block with tool radius compensation (the block is extended by its end tangent).
The search for the intersection is then identical to the procedure for G461.



Retraction behavior with G462
(see example at the end of this section)

With G462, the corner generated by N10 and N20 in the example program is not machined to the full extent actually possible with the tool used. However, this behavior may be necessary if the part contour (as distinct from the programmed contour), to the left of N20 in the example, is not permitted to be violated even with y values greater than 10mm.
If KONT is active (travel round contour at start or end point), the behavior differs according to whether the end point is in front of or behind the contour.

End point in front of contour

If the end point is in front of the contour, the retraction behavior is the same as with NORM. This property does not change even if the last contour block for G451 is extended with a straight line or a circle. Additional circumnavigation strategies to avoid a contour violation in the vicinity of the contour end point are therefore not required.

End point behind contour

If the end point is behind the contour, a circle or straight line is always inserted depending on G450 / G451. In this case, G460-462 has no effect.
If, in this situation, the last traversing block has no intersection with a preceding block, an intersection with the inserted contour element or with the linear section from the end point of the circumnavigation circle to the programmed end point can result.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

If the inserted contour element is a circle (G450), and it intersects with the preceding block, this is the same as the intersection which would be produced with NORM and G461. In general, however, a remaining section of the circle still has to be traversed. An intersection calculation is no longer required for the linear section of the retraction block.
In the second case (if no intersection is found between the inserted contour element and the preceding blocks), the intersection between the retraction straight line and a preceding block is approached.
Therefore, when G461 or G462 is active, a behavior different to G460 can only arise if NORM is active or if the behavior with KONT is identical to NORM due to the geometrical conditions.

Additional notes

The approach behavior is symmetrical to the retraction behavior.
The approach/retraction behavior is determined by the state of the G command in the approach/retraction block. The approach behavior can therefore be set independently of the retraction behavior.

Programming example

G461 for approach

`N10 $TC_DP1[1,1]=120`	`; Tool type mill`
`N20 $TC_DP6[1,1]=10`	`; Radius`
`N30 X0 Y0 F10000 T1 D1`	
`N40 Y20`	
`N50 G42 X50 Y5 G461`	
`N60 Y0 F600`	
`N70 X30`	
`N80 X20 Y-5`	
`N90 X0 Y0 G40`	
`N100 M30`	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.12 Collision monitoring, CDON, CDOF

Programming

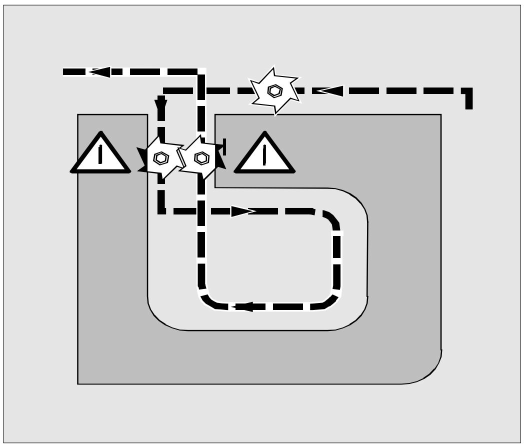
```
CDON
CDOF
```

Explanation of the commands

CDON	Activate bottleneck detection
CDOF	Deactivate bottleneck detection

Function

When CDON (Collision Detection ON) and tool radius compensation are active, the control monitors the tool paths with Look Ahead contour calculation. This Look Ahead function allows possible collisions to be detected in advance and permits the control to actively avoid them.



When collision detection is off (CDOF), a search is made at inside corners in the previous traversing block (and if necessary in blocks further back) for a common intersection point for the current block. If no intersection is found with this method, an error is generated.

CDOF helps prevent the incorrect detection of bottlenecks, e.g. due to missing information which is not available in the NC program.

Machine manufacturer (MH 8.16)

The number of NC blocks monitored can be defined in the machine data (see machine manufacturer).

Sequence

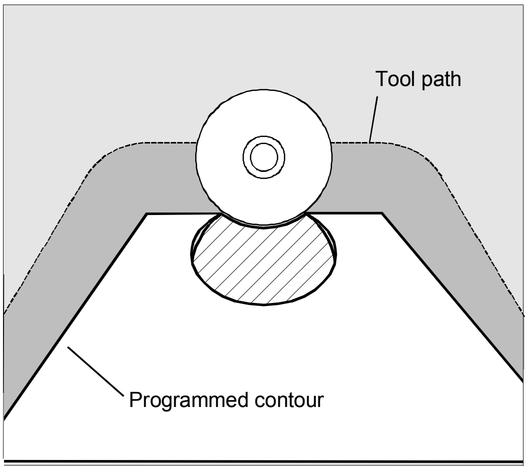
The following are some examples of critical machining situations which can be detected by the control and compensated for by modifying the tool paths.

In order to prevent program stops, you should always select the tool with the widest radius from all of the tools used when testing the program.

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

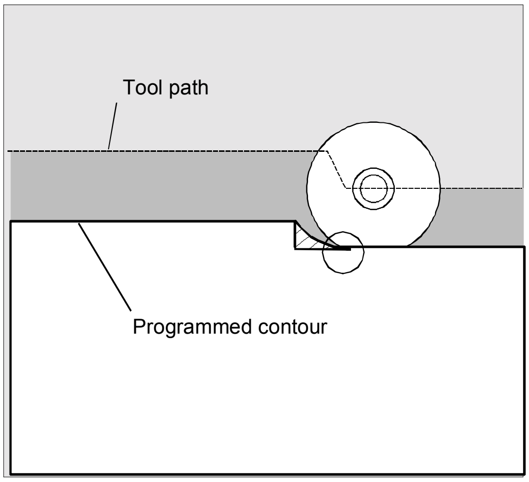
Bottleneck detection

Since the tool radius selected is too wide to machine this inside contour, the "bottleneck" is bypassed.
An alarm is output.



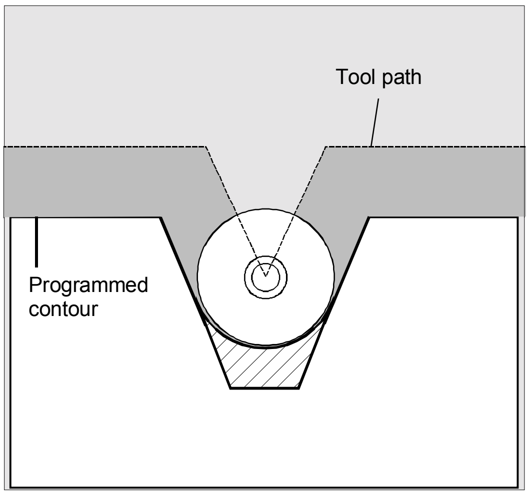
Contour path shorter than tool radius

The tool travels round the workpiece corner on a transition circle and then continues to follow the programmed contour exactly.



Tool radius too wide for inside machining

In such cases, machining of the contours is performed only as far as is possible without causing damage to the contour.



840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.13 2 1/2 D tool offset, CUT2D, CUT2DF

Programming

CUT2D
CUT2DF

Explanation

CUT2D	Activate 2 1/2 D radius compensation (default)
CUT2DF	Activate 2 1/2 D radius compensation, tool radius compensation relative to the current frame or to inclined planes

Function

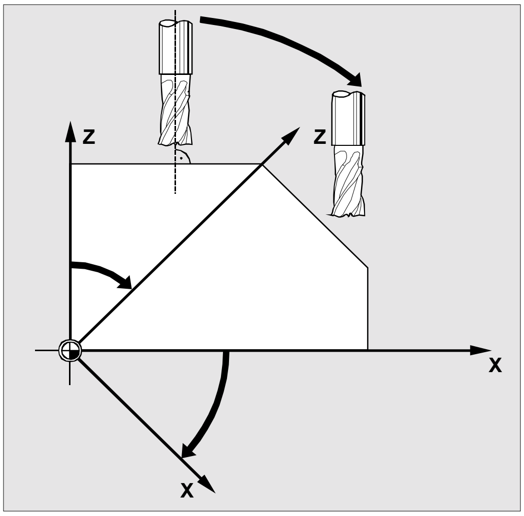
With CUT2D or CUT2DF you define how the tool radius compensation is to act or to be interpreted when machining in inclined planes.

Sequence

Tool length compensation
Tool length compensation is generally calculated according to the non-rotated working plane fixed in space.

Tool radius compensation, CUT2D
As for many applications, tool length compensation and tool radius compensation are calculated in the **fixed** working plane specified with G17 to G19.

Example for G17 (X/Y plane):
Tool radius compensation is active in the non-rotated X/Y plane, tool length compensation in the Z direction.



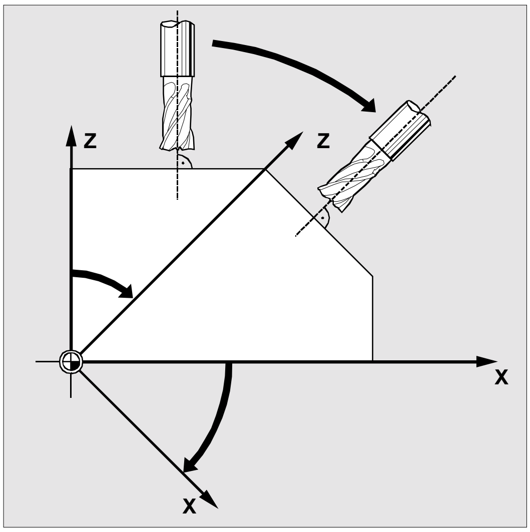
8.13 2 1/2 D tool offset, CUT2D, CUT2DF

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

For machining on inclined surfaces, the tool compensation values have to be defined accordingly, or be calculated using the functions for "Tool length compensation for orientable tools". For more information on this calculation method, see Section "Tool orientation and tool length compensation".

CUT2D is used when the orientation of the tool cannot be changed and the workpiece is rotated for machining on inclined surfaces.
CUT2D is generally the standard setting and does not therefore have to be specified explicitly.

Tool radius compensation, CUT2DF
In this case, it is possible to arrange the tool orientation perpendicular to the inclined working plane on the machine.



If a frame containing a rotation is programmed, the compensation plane is also rotated with CUT2DF. The tool radius compensation is calculated in the rotated machining plane.

The tool length compensation continues to be active relative to the non-rotated working plane.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.14 Tool length offset for orientable tools: TCARR, TCOABS, TCOFR

Programming

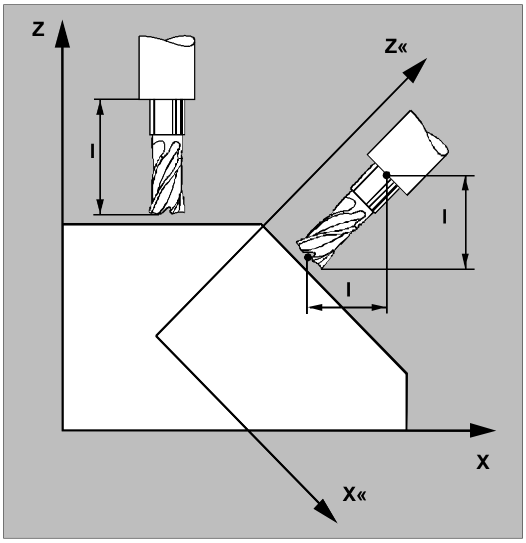
```
TCARR=[m]
TCOABS
TCOFR
```

Explanation

TCARR=[m]	Request toolholder with the number "m"
TCOABS	Determine tool length components from the orientation of the current toolholder.
TCOFR	Determine tool length components from the orientation of the active frame.

Function

When the spatial orientation of the tool changes, its tool length components also change.
After a reset, e.g. through manual setting or change of the toolholder with a fixed spatial orientation, the tool length components also have to be determined again.
This is performed using the TCOABS and TCOFR path commands.



Sequence

Request toolholder TCARR
With TCARR the toolholder number m is requested with its geometry data (offset memory).
With m=0, the active toolholder is deselected.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Additional notes

The geometry data of the toolholder only become active after a tool is called. The selected tool remains active after a toolholder change has taken place.

The current geometry data for the toolholder can also be defined in the parts program via the corresponding system variables.

For a definition of toolholder kinematics with system variables, see
References: /PGA/, "Programming Guide Advanced"
Section "Toolholder kinematics"

Determine tool length compensation from the orientation of the toolholder, TCOABS

TCOABS calculates the tool length compensation from the current orientation angles of the toolholder; stored in system variables $TC_CARR13 and $TC_CARR14.

In order to make a new calculation of the tool length compensation when frames are changed, the tool has to be selected again.

Additional notes

The tool orientation must be manually adapted to the active frame.

When the tool length compensation is calculated, the angle of rotation of the toolholder is calculated in an intermediate step. With toolholders with two rotary axes, there are generally two sets of rotation angles which can be used to adapt the tool orientation to the active frame; therefore, the rotation angle values stored in the system variables must at least correspond approximately to the mechanically set rotation angles.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Additional notes

It is not possible for the control to check whether the rotation angles calculated by means of the frame orientation are settable on the machine.

If the rotary axes of the toolholder are arranged such that the tool orientation calculated by means of the frame orientation cannot be reached, then an alarm is output.
If there is a switchover between TCOFR and TCABS, the tool length compensation is calculated again.

The combination of tool precision compensation and the functions for tool length compensation on movable toolholders is not permissible. If both functions are called simultaneously, an error message is issued.

The **TOFRAME** function allows a frame to be defined on the basis of the direction of orientation of the selected toolholder. For more detailed information please refer to the Programming Guide Fundamentals in Section "Frames".

When orientation transformation is active (3, 4 or 5-axis transformation), it is possible to select a toolholder with an orientation deviating from the zero position without causing output of an alarm.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.15 Grinding-specific tool monitoring in parts program TMON, TMOF

Assignment of tool-specific parameters

Further tool-specific parameters can be set up in the machine data and assigned by the user.

Parameter	Meaning	Data type
Tool-specific parameters		
`$TC_TPG1`	Spindle number	Integer
`$TC_TPG2`	Chaining rule The parameters are automatically kept identical for the left and right side of the wheel.	Integer
`$TC_TPG3`	Minimum wheel radius	Real
`$TC_TPG4`	Minimum wheel width	Real
`$TC_TPG5`	Current wheel width	Real
`$TC_TPG6`	Maximum speed	Real
`$TC_TPG7`	Maximum peripheral speed	Real
`$TC_TPG8`	Angle of inclined wheel	Real
`$TC_TPG9`	Parameter number for radius calculation	Integer

Programming

```
TMON (T No.)
TMOF (T No.)
```

Explanation of the commands

`TMON (T No.)`	Activate tool monitoring	It is only necessary to specify the T number if the tool with this number is not active.
`TMOF (T No.)`	Deselect tool monitoring T No. = 0: Deactivate monitoring for all tools	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Function

The command TMON is used to activate geometry and speed monitoring for grinding tools (types 400–499) in the NC parts program. Monitoring remains active until it is deactivated in the parts program with TMOF.

Additional notes

You can only activate tool monitoring if the toolspecific grinding data $TC_TPG1 to $TC_TPG9 (see Programming Guide Advanced) are set.
According to the machine data settings, tool monitoring for the grinding tools (types 400–499) can be automatically activated when the tool selection is activated.
Only **one** monitoring routine can be active at any one time for each spindle.

Geometry monitoring
The current wheel radius and the current width are monitored.

The set speed is monitored against the speed limitation cyclically with allowance for the spindle override.
The speed limit is the smaller value resulting from a comparison of the maximum speed with the speed calculated from the maximum wheel peripheral speed and the current wheel radius.

Working without a T or D number
In the machine data, a default T and D number can be set; they no longer have to be programmed and become active after Power On / Reset.
Example:
All machining is performed with the same grinding wheel.
Machine data can be set to keep the current tool active after Reset;
see /PGA/ Programming Guide "Advanced".

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Machine manufacturer (MH 8.11)

See machine manufacturer's specifications.

8.16 Additive offsets (SW 5 and higher)

Additive offsets are essentially process offsets which can be programmed during machining. They refer to the geometrical data of a tool edge and are thus components of the tool edge data.
The data of an additive offset are addressed via a DL number (DL: location-dependent; offsets relative to relevant location) and enter tool offset via parameter display in the Parameter area.

8.16.1 Select offset (by DL number)

Programming

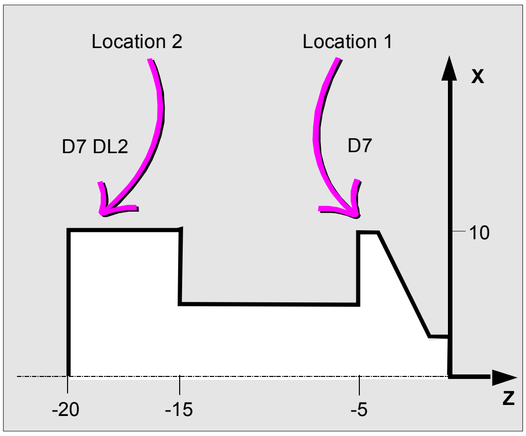
`DL=x` additive offset selection, x = 1 to 6

Explanation

- Up to 6 additive offsets can be activated per D number
  (saved under the respective DL numbers).
- You can distinguish between setup and wear values.
- DL=1 is activated when a D number is called.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Function



Setup value:
The setup value is defined optionally by the machine manufacturer by MD.
Same tool edge:
The same tool edge is used for 2 bearing seats (see example). Compensation can be made for a location-dependent measurement error occurring as a result of machining forces, etc.
Fine offset:
Location-dependent allowances can be made for over/under-dimensioning.

Machine manufacturer (MH 8.13)

Machine data are used to activate and define the number of additive offsets.

See machine manufacturer's specifications.

Programming example

`N110 T7 D7`	The tool turret is positioned at location 7. D7 and DL=1 are activated and traversed in the next block
`N120 G0 X10 Z1`	N120 retracted
`N130 G1 Z-6`	
`N140 G0 DL=2 Z-14`	DL=2 is activated in addition to D7 and traversed in the next block
`N150 G1 Z-21`	
`N160 G0 X200 Z200`	Approach tool change position
`...`	

8.16.2 Define wear and setup values

Wear and setup values can be read and written via system parameters and the corresponding OPI services.
The logic is based on the logic of the corresponding system parameters for tools and tool edges.

8.16 Additive offsets (SW 5 and higher)

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming

`$TC_SCPxy[t,d]` Wear values
`$TC_ECPxy[t,d]` Setup values

Explanation of the parameters

`$TC_SCPxy`	Wear values are assigned to the corresponding geometry parameters via xy, where x is the number of the wear value and y is the reference to the geometry parameter.
`$TC_ECPxy`	Setup values are assigned to the corresponding geometry parameters via xy, where x is the number of the setup value and y is the reference to the geometry parameter.
`t`	T number of the tool
`d`	D number of the tool edge

Function

System parameters $TC_DP3 - $TC_DP11 describe the tool geometry. In addition to the parameters for the physical wear ($TC_DP12 - $TC_DP20), up to six wear values ($TC_SCP1y - $TC_SCP6y) and up to six setup values ($TC_ECP1y - $TC_ECP6y) can be assigned to each geometry parameter.

Example:

Parameters: $TC_DP**3** (Length 1, for rotary tools)
Wear values: $TC_SCP1**3** - $TC_SCP6**3**
Setup values: $TC_ECP1**3** - $TC_ECP6**3**

$TC_SCP4**3** [t, d] = 1.0

The wear value of length 1 is set to the value 1.0 for the tool edge (D number d) of the tool (t).

Note

The defined wear and setup values are added to the geometry parameters and the other offset parameters (D numbers).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

8.16.3 Delete additive offsets (DELDL)

Programming

```
status = DELDL[t,d]
```

Explanation of the parameters

`DELDL[t,d]`	All additive offsets of the tool edge with D number d of tool t are deleted.
`DELDL[t]`	All additive offsets of all tool edges of tool t are deleted.
`DELDL`	All additive offsets of the tool edges of all tools of the TO-unit are deleted (for the channel in which the command is programmed).
`status`	0: Deletion was successful. –1: Deletion could not be performed (if the parameters describe only one tool edge), or deletion was not complete (if the parameters describe several tool edges).

Function

DELDL is used to delete the additive offsets for the tool edge of a tool (in order to release memory). Both the defined wear values and the setup values are deleted.

Additional notes

The wear and setup values of active tools cannot be deleted (similar to deletion of D offsets or tool data).

840D
NCU 572
NCU 573

840Di

8.17 Tool offset – special features (SW 5 and higher)

Function

Setting data SD 42900 – SD 42940 can be used to control the evaluation of the sign for tool length and wear.
The same applies to the behavior of the wear components when mirroring geometry axes or changing the machining plane.

Where reference is made below to wear values, this means in each case the sum of the actual wear values ($TC_DP12 to $TC_DP20) and the total offsets with the wear ($SCPX3 to $SCPX11) and setup values ($ECPX3 to $ECPX11).
You will find more information on resulting offsets in /FBW/, Description of Functions, Tool Management.

See also:

- /PGA/, Programming Guide Advanced, Chapter 8
- /PG/, Programming Guide Fundamentals, Chapter 8
- Description of Functions, Basic Machine (Part 1), Tool Compensation (W1)

Required setting data

SD42900 MIRROR_TOOL_LENGTH	Mirroring of tool length components and components of the tool base dimension
SD42910 MIRROR_TOOL_WEAR	Mirroring of wear values of the tool length components
SD42920 WEAR_SIGN_CUTPOS	Sign evaluation of the wear components depending on the tool point direction
SD42930 WEAR_SIGN	Inverts the sign of the wear dimensions
SD42940 TOOL_LENGTH_CONST	Assignment of tool length components to geometry axes

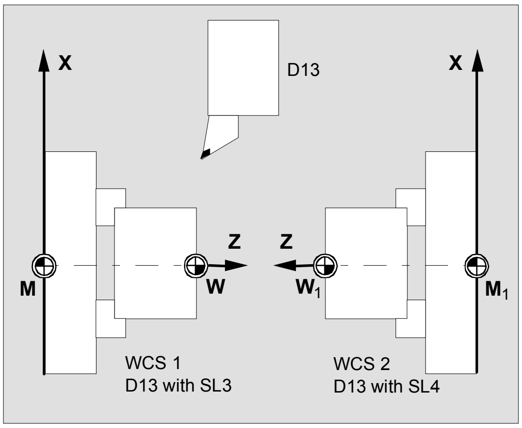
840D
NCU 572
NCU 573

840Di

8.17.1 Mirroring of tool lengths

SD 42900 MIRROR_TOOL_LENGTH

Setting data **not equal to** zero:
The tool length components ($TC_DP3, $TC_DP4 and $TC_DP5) and the components of the tool base dimensions ($TC_DP21, $TC_DP22 and $TC_DP23), whose associated axes are mirrored, are also mirrored – through sign inversion.
The wear values are **not** mirrored. If these are also to be mirrored, setting data $SC_MIRROR_TOOL_WEAR must be enabled.



SD 42910 MIRROR_TOOL_WEAR

Setting data **not equal to** zero:
The wear values of the tool length components, whose associated axes are mirrored, are also mirrored by sign inversion.

8.17.2 Wear sign evaluation

SD 42920 WEAR_SIGN_CUTPOS

Setting data **not equal to** zero:
In the case of tools with a relevant tool point direction (turning and grinding tools – tool types 400–599), the sign evaluation of the wear components depends on the tool point direction in the machining plane. This setting data has no effect on tool types which do not have a relevant tool point direction.
In the following table, the dimensions whose sign is inverted by SD 42920 (not equal to 0) are marked with an X:

Tool point direction	Length 1	Length 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

840D
NCU 572
NCU 573

840Di

Additional notes

The sign settings of SD 42920 and 42910 are independent. For example, if the sign of a dimensional parameter is changed by both setting data, the resulting sign is unchanged.

SD 42930 WEAR_SIGN
Setting data **not equal to** zero:
Inverts the sign of all wear dimensions. It acts both on the tool length and on other quantities such as tool radius, rounding radius, etc.
If a positive wear dimension is entered, the tool becomes "shorter" and "thinner".
Example: See following section "Activation of modified setting data".

8.17.3 Tool length and plane change

SD 42940 TOOL_LENGTH_CONST
Setting data **not equal to** zero:
The assignment between the tool length components (length, wear and tool base dimension) and the geometry axes is not modified when the machining plane is changed (G17–19).

The following table shows the assignment between the tool length components and the geometry axes for turning and grinding tools (tool types 400 to 599):

Contents	Length 1	Length 2	Length 3
17	Y	X	Z
18***)**	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

***)** Each value not equal to 0 which is not equal to one of the six listed values is evaluated like value 18.

840D
NCU 572
NCU 573

840Di

The following table shows the assignment between the tool length components and the geometry axes for all other tools (tool types < 400 and > 599):

Machining plane	Length 1	Length 2	Length 3
17*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Each value not equal to 0 which is not equal to one of the six listed values is evaluated as value 17.

Additional notes

For representation in tables, it is assumed that geometry axes 1 to 3 are named X, Y, Z. The axis order and not the axis identifier determines the assignment between an offset and an axis.

Activation of modified setting data

Function

When the setting data described above are modified, the tool components are not reevaluated until the next time a tool edge is selected. If a tool is already active and the data of this tool are to be reevaluated, the tool must be selected again.

The same applies in the event that the resulting tool length is modified due to a change in the mirroring status of an axis. The tool must be selected again after the mirror command, in order to activate the modified tool length components.

840D
NCU 572
NCU 573

840Di

Orientable toolholders and new setting data

Function

Setting data SD 42900–42940 have no effect on the components of an active orientable toolholder. However, the calculation with an orientable toolholder always allows for a tool with its total resulting length (tool length + wear + tool base dimension). All modifications initiated by the setting data are included in the calculation of the resulting total length; i.e. vectors of the orientable toolholder are independent of the machining plane.

Additional notes

When orientable toolholders are used, it is frequently practical to define all tools for a non-mirrored basic system, even those which are only used for mirrored machining. When machining with mirrored axes, the toolholder is then rotated such that the actual position of the tool is described correctly. All tool length components then automatically act in the correct direction, dispensing with the need for control of individual component evaluation via setting data, depending on the mirroring status of individual axes.

The "orientable toolholder" functionality can also be useful on machines which have no physical means of turning tools but which feature permanently installed tools with different orientations. In this case, tools can be uniformly dimensioned in a basic orientation; the dimensions relevant to the machining operation are acquired by rotating a virtual toolholder.

840D
NCU 572
NCU 573

840Di

8.18 Tools with a relevant tool point direction (SW 5 and higher)

Function

Up to SW 4.x

In the case of tools with a relevant tool point direction (turning and grinding tools – tool types 400–599; see Subsection 8.17.2), a change from G40 to G41/G42 or vice-versa is treated as a tool change. If a transformation is active (e.g. TRANSMIT), this leads to a preprocessor stop (decoding stop) and hence possibly to deviations from the intended part contour.

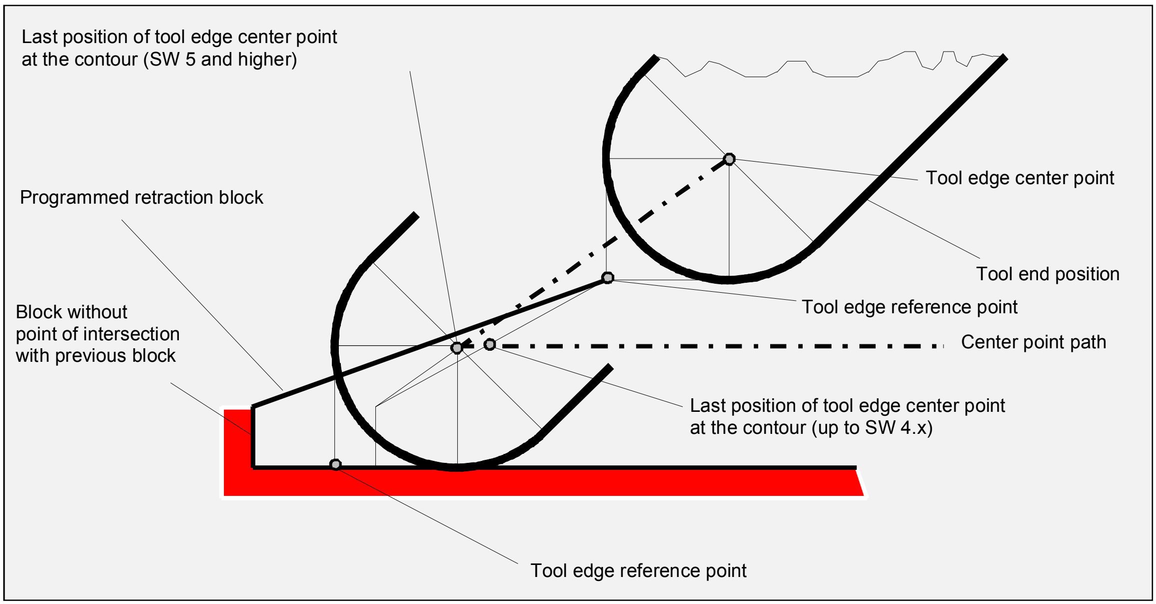
SW 5 and higher

The following changes have been made:

1. A change from G40 to G41/G42 and vice-versa is no longer treated as a tool change. A preprocessor stop therefore no longer occurs with Transmit.
2. The straight line between the tool edge center points at the block start and block end is used to calculate intersection points with the approach and retraction block. The difference between the tool edge reference point and the tool edge center point is superimposed on this movement.
   During approach and retraction with KONT (tool travels around the contour point; see Section 8.9), superimposing takes place in the linear subblock of the approach or retraction movement. The geometrical relationships are therefore identical for tools with and without a relevant tool point direction. Differences from the previous behavior occur only in relatively rare situations where the approach or retraction block does not intersect with an adjacent motion block, see diagram below.

840D
NCU 572
NCU 573

840Di



3. In circle blocks and in motion blocks containing rational polynomials with a denominator degree > 4, it is not permitted to change a tool with active tool radius compensation in cases where the distance between the tool edge center point and the tool edge reference point changes. With other types of interpolation, it is now possible to change when a transformation is active (e.g. Transmit).

4. For tool radius compensation with variable tool orientation, the transformation from the tool edge reference point to the tool edge center point can no longer be performed by means of a simple zero offset. Tools with a relevant tool point direction are therefore not permitted for 3D peripheral milling (an alarm is output).

Additional notes

This subject has no relevance with respect to face milling operations, since only defined tool types without relevant tool point direction have been permitted for these to date anyway. (Tools with a tool type that is not expressly permissible are treated like a ball end mill with the specified radius). A tool point direction parameter is ignored).

■

Miscellaneous functions

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

9.1 Auxiliary function outputs

Function

The auxiliary function output sends information to the PLC indicating when the NC program needs the PLC to perform specific switching operations on the machine tool. The auxiliary functions are output, together with their parameters, to the PLC interface. The values and signals must be processed by the PLC user program.

Function outputs
The following functions can be downloaded to the programmable controller:

- Select tool T
- Tool offset D, DL (SW 5.2 and higher)
- Feed F / FA
- Spindle speed S
- H functions
- M functions.

For the above-mentioned functions it is possible to define whether they are to be transferred during the machining sequence, and which reactions are to be activated.

For each function group or individual function, machine data define whether the output is initiated

- before the traversing movement,
- with the traversing movement or
- after the traversing movement.

The PLC can be programmed to acknowledge auxiliary function outputs in various ways.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming

```
Letter[address extension]=Value
```

Explanation

The letters which can be used for auxiliary functions are:
M, S, H, T, D, DL, F.
In the following table you will find information about the meaning and value ranges for the address extension and the value in auxiliary function outputs. The maximum number of auxiliary functions of the same type per block is also specified.

DL in SW 5.2 and higher

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Overview of auxiliary functions, programming							
Func-tion	**Address extension** (integer)		**Value**			**Explanation**	**Number per block**
	Meaning	**Range**	**Range**	**Type**	**Meaning**		
M	-	implicit 0	0–99	INT	Function	The address extension is 0 for the value range between 00 and 99. M0, M1, M2, M17, M30 must be used without an address extension.	5
	Spindle no.	1–12	1–99		Function	M3, M4, M5, M19, M70 with address extension spindle No., for example, M5 for spindle 2: M2=5. Master spindle is used if no spindle is specified.	
	Any	0–99	100– (max. INT value)		Function	User M function	
S	Spindle no.	1–12	0–±3.4028 ex 38	REAL	Spindle speed	Without spindle no. for master spindle	3
H	Any	0–99	±(max. INT value) ±3.4028 ex 38	INT (SW 5) REAL	Any	Functions have no effect in the NCK; only to be implemented on the PLC	3
T	Spindle no. (with active tool manage-ment)	1–12	0–32000 (or tool names with active tool manage-ment)	INT	Tool selection	Tool names are not passed to the PLC interface.	1
D			0–9	INT	Tool offset selection	D0 deselection, D1 default	1
DL	Location-dependent offset	1–6	±3.4028 ex 38	REAL	See tool fine offset selection /FBW/	Refers to previously selected D number	1
F	Path feedrate	0	0.001– 999 999.999	REAL	Path feedrates		6
(FA)	Axis No.	1–31	0.001– 999 999.999		Axis feedrates		

Number of functions output per NC block
Up to ten function outputs can be programmed in one NC block.
Auxiliary functions can also be output from the action component of **synchronized actions**. See /FBSY/.

The highest number for a type specified in the table must not be exceeded.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Grouping
The above-mentioned functions can be grouped together. Group assignment is predefined for some M commands. The acknowledgment behavior can be defined by the grouping.

Acknowledgments
High-speed function outputs, QU
Functions which have not been programmed as high-speed outputs can be defined as high-speed outputs for individual outputs with the keyword QU. Program execution continues without waiting for the acknowledgment of the miscellaneous function (the program waits for the transport acknowledgment).

This helps avoid unnecessary stoppages and interruptions to traversing movements.

Machine manufacturer (MH9.1)

The appropriate MD must be set for "high-speed function outputs".
(see /FB/, H2, FB Auxiliary Function Output).

Programming

```
M=QU(…)
H=QU(…)
```

Examples:

```
N10 H=QU(735)                    ;High-speed output for H735
```

```
N10 G1 F300 X10 Y20 G64
N20 X8 Y90 M=QU(7)
```

M7 was programmed as a high-speed output, so continuous-path mode (G64) is not interrupted.

You should only use this function in individual cases, because it can affect the time synchronization as a result of interaction with other function outputs.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Function outputs for travel commands

Time is needed to transfer information and wait for a corresponding response, and this has an impact on the travel movements.

High-speed acknowledgment without block change delay

In SW 5 and higher, the block change behavior can be controlled by machine data. When the "without block change delay" setting is selected, the system response with respect to high-speed auxiliary functions is as follows:

Auxiliary function output	Behavior
Before the movement	The block transition between blocks with high-speed auxiliary functions takes place **with no** interruption and **with no** reduction in speed. The auxiliary function output takes place in the first interpolation cycle of the block. The following block is executed with no acknowledgment delay.
During the movement	The block transition between blocks with high-speed auxiliary functions takes place **with no** interruption and **with no** reduction in speed. The auxiliary function output takes place during the block. The following block is executed with no acknowledgment delay.
After the movement	The movement stops at the end of the block. The auxiliary function output takes place at the end of the block. The following block is executed with no acknowledgment delay.

Function outputs in continuous-path mode

Function outputs before the traversing movements interrupt continuous-path mode (G64/G641) and generate an exact stop for the previous block.

Function outputs after the traversing movements interrupt continuous-path mode (G64/G641) and generate an exact stop for the current block.

A wait for an outstanding acknowledgment signal from the PLC can also cause an interruption to continuous-path mode, e.g. M command sequences in blocks with extremely short path lengths.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

9.1.1 M functions

Programming

`M...` Possible values 0 to 9999 9999, integers
(Max. INT value in SW 5 and higher)

Function

M functions initiate, for example, switching operations such as "Coolant ON/OFF" and other operations on the machine. Permanent functions have already been assigned to some of the M functions by the control manufacturer (see following list).

List of predefined M functions

`M0*`	Programmed stop
`M1*`	Optional stop
`M2*`	End of main program with return to beginning of program
`M30*`	End of program, same effect as M2
`M17*`	End of subprogram
`M3`	Spindle clockwise
`M4`	Spindle counterclockwise
`M5`	Spindle stop
`M6`	Tool change (default setting)
`M70`	Spindle is switched to axis mode
`M40`	Automatic gear change
`M41`	Gear step 1
`M42`	Gear step 2
`M43`	Gear step 3
`M44`	Gear step 4
`M45`	Gear step 5

*Extended address notation cannot be used for the functions marked with *.*

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Machine manufacturer (MH9.2)

All free M function numbers can be assigned by the machine manufacturer, e.g. with switching functions for controlling clamping fixtures or for activating/ deactivating other machine functions, etc.

Machine manufacturer (MH9.3)

See machine manufacturer's specifications

The commands M0, M1, M2, M17 and M30 are always initiated ***after*** *the traversing movement.*

Predefined M commands

Certain important M functions for program execution are supplied as standard with the control:

Programmed stop, M0

Machining stops in the NC block with M0. You can now, for example, remove swarf, remeasure, etc.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Optional stop, M1

M1 can be set with

- MMC/dialog "Program Control"
- or the VDI interface.

End of program, M2, M17, M30

A program is terminated with M2, M17 or M30 and reset to the beginning of the program. If the main program is called from another program (as a subroutine), M2/M30 has the same effect as M17 and vice versa, i.e. M17 has the same effect in the main program as M2/M30.

Spindle functions, M3, M4, M5, M19, M70

The extended address notation with spindle number is used for all spindle functions.

Example:

M2=3 means CW spindle rotation for the second spindle. If no address extension is programmed, the function applies to the master spindle.

Programming example

`N10 S...`	
`N20 X... M3`	M function in the block with axis movement, spindle accelerates before the X axis movement
`N180 M789 M1767 M100 M102 M376`	Max. of 5 M functions in the block

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

9.1.2 H functions

Programming

```
N10 G0 X20 Y50 H3=–11.3
```

Function

H functions are used to transfer information to the PLC (programmable logic controller), in order to activate specific switching operations. H functions are REAL values.

Machine manufacturer (MH9.4)

The meaning of the functions is determined by the manufacturer.

Sequence

Number of functions per NC block
Up to three H functions can be programmed in one NC block.

■

Arithmetic Parameters and Program Jumps

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

10.1 Arithmetic parameters R

Programming

`Rn=...`

Explanation

R	Arithmetic parameter
n	Number of the arithmetic parameter, n= 0 to max. See machine data or machine manufacturer for max.; default setting: max = 0–99

Machine manufacturer (MH10.1)

The number of R parameters is set in the machine data, or see specifications of machine manufacturer.

Function

The arithmetic parameters are used, for example, if an NC program is not only to be valid for values assigned once, or if you need to calculate values. The required values can be set or calculated by the control during program execution. Another possibility consists of setting the arithmetic parameter values through operation. If values have been assigned to the arithmetic parameters, they can be assigned to other NC addresses in the program. The value of these addresses should be flexible.

Value assignments

You can assign values in the following range to the arithmetic parameters:

`±(0.000 0001 ... 9999 9999)`

(8 decimal places and sign and decimal point).

- The decimal point can be omitted for integer values.
- A positive leading sign can always be omitted.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Example:

```
R0=3.5678 R1=-37.3 R2=2 R3=-7
R4=-45678.1234
```

It is possible to assign an extended numerical range by using exponential notation:

Example:

$\pm (10^{-300} \ ... \ 10^{+300})$

The value of the exponent is written after the EX characters; maximum total number of characters: 10 (including leading signs and decimal point)

Value range of EX: –300 to +300

Example:

R0=-0.1EX-5 ;Meaning: R0 = –0.000 001

R1=1.874EX8 ;Meaning: R1 = 187 400 000

Note:

- There can be several assignments in one block; or assignments of expressions.
- Value assignment must be in a separate block.

Assignments to other addresses

The flexibility of an NC program lies in assigning these arithmetic parameters or expressions with arithmetic parameters to other NC addresses.

Values, expressions and arithmetic parameters can be assigned to all addresses;

Exception: address N, G and L.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

When assigning, write the character " = " after the address character.
It is also possible to have an assignment with a negative leading sign.
A separate block is required for assignments to axis addresses (traversing instructions).
Example:
`N10 G0 X=R2` ;Assignment to X axis

Arithmetic operations and functions
When operators/arithmetic functions are used, it is necessary to use the conventional mathematical notation. Machining priorities are set by parentheses. Otherwise, multiplication and division take precedence over addition and subtraction.
Degrees are used for the trigonometrical functions.

Programming example: R parameter

`N10 R1= R1+1`	The new R1 is calculated from the old R1 plus 1
`N20 R1=R2+R3    R4=R5-R6    R7=R8* R9 R10=R11/R12`	
`N30 R13=SIN(25.3)`	R13 equals sine of 25.3 degrees
`N40 R14=R1*R2+R3`	Multiplication or division takes precedence over addition or subtraction R14=(R1*R2)+R3
`N50 R14=R3+R2*R1`	Result, the same as block N40
`N60 R15=SQRT(R1*R1+R2*R2)`	Meaning: R15=square root of $R1^2+R2^2$

Programming example:
Assignment of axis values

```
N10 G1 G91 X=R1 Z=R2 F300
N20 Z=R3
N30 X=-R4
N40 Z=-R5
...
```

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

10.2 Unconditional program jumps

Programming

```
GOTOB&lt;Jump destination specification>
GOTOF<Jump destination specification>
GOTO/GOTOC < Jump destination variable>
```

Explanation

`GOTOB`	"Jump instruction" with backward destination (toward beginning of program)
`GOTOF`	Jump instruction with forward jump destination (towards program end)
`GOTO`	Jump instruction with destination search going forward than backward (first toward end of program then toward beginning of program)
`GOTOC`	Suppress Alarm 14080 "Destination not found". Jump instruction with destination search first forward then backward (first toward end of program and then toward beginning of program)
`<Jump destination specification>`	Destination parameters for label, block number, or string variable
`Label`	Destination for a jump command
`Label:`	Labeling of destination within the program
`Block number`	Destination as main block or subblock number (e.g. : 200, N300)
`String variable`	Variable of type string containing a label or block number.

Function

By default, main programs, subprograms, cycles and interrupt routines execute the blocks in the sequence in which they were programmed.
Program jumps can be used to modify this sequence.

Sequence

Jump destinations with user-defined names can be programmed in a routine. The command GOTOF or GOTOB can be used to branch to a jump destination from any other point within the same program. The program then resumes execution at the instruction immediately following the jump destination.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Destination not found

If the destination is not found, program execution is terminated with Alarm 14080 "Destination not found". Command GOTOC suppresses this alarm. Program execution is resumed at the line following the GOTOC command.

Destination backward

```
1. Jump with label
   Label_1:           ;Destination
   ....
   GOTOB Label_1
```

Jump forward

```
2. Jump with block number
   GOTOF N100
   ....
   N100               ;Destination
```

Indirect jumps

```
3. Jump to block number
   N5 R10=100
   N10 GOTOF "N"<<R10 ;Jump to the block
                      whose number is in R10
   N90
   N100               ;Destination
   N110
4. Jump to labels
   DEF STRING[20] DEST
   DEST = "Label2"    ;Jump with
   GOTOF DEST         variables destination
   Label1: T="Drill1"
   ....
   Label2: T="Drill2" ;Destination
```

Additional notes

The unconditional jump must be programmed in a separate block.

In programs with unconditional jumps, the end of program M2/M30 does not have to appear at the end of the program.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming examples

`N10 …`	
`N20 GOTOF LABEL_0`	;Jump forward to LABEL_0
`N30 …`	
`N40 LABEL_1: R1=R2+R3`	;Destination LABEL_1
`N50 …`	
`N60 LABEL_0:`	;Destination LABEL_0
`N70 …`	
`N80 GOTOB LABEL_1`	;Jump backward to LABEL_1
`N90 …`	

10.3 Conditional program jumps

Programming

`IF` expression `GOTOB <Jump destination specification>`

`IF` expression `GOTOF <Jump destination specification>`

`IF` expression `GOTO/GOTOC <Jump destination specification>`

Explanation of the commands

`IF`	Keyword for condition
`GOTOB`	"Jump instruction" with backward destination (toward beginning of program)
`GOTOF`	Jump instruction with forward jump destination (towards program end)
`GOTO`	Jump instruction with destination search going forward than backward (first toward end of program then toward beginning of program)
`GOTOC`	Suppress Alarm 14080 "Destination not found". Jump instruction with destination search first forward then backward (first toward end of program and then toward beginning of program)
`<destination>`	Destination parameters for label, block number, or string variable
`Label`	Destination for a jump command
`Label:`	Labeling of destination within the program
`Block number`	Destination as main block or subblock number (e.g. : 200, N300)
`String variable`	Variable of type string containing a label or block number.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Comparison and logical operands

==	Equal to
<>	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

For further information, see
/PGA/ Chapter 1 "Flexible NC programming"

Function

Jump conditions can be formulated using IF instructions. The jump to the programmed jump destination is only performed if the jump condition is fulfilled.

Sequence

The jump condition can be programmed with any comparison or logic operation (result: TRUE or FALSE). The program jump is executed if the result of the operation is TRUE.

The jump destination can only be a block with a label or block number that appears within the program.

Several conditional jumps can be formulated in the same block.

Programming example

`N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20`	Assignment of initial values
`N41 MA1: G0 X=R2*COS(R1)+R5 -> -> Y=R2*SIN(R1)+R6`	Calculation and assignment to axis address
`N42 R1=R1+R3 R4=R4-1`	Specification of variable
`N43 IF R4>0 GOTOB MA1`	Jump instruction with label
`N44 M30`	End of program

■

Subprograms and Repetition of Program Sections

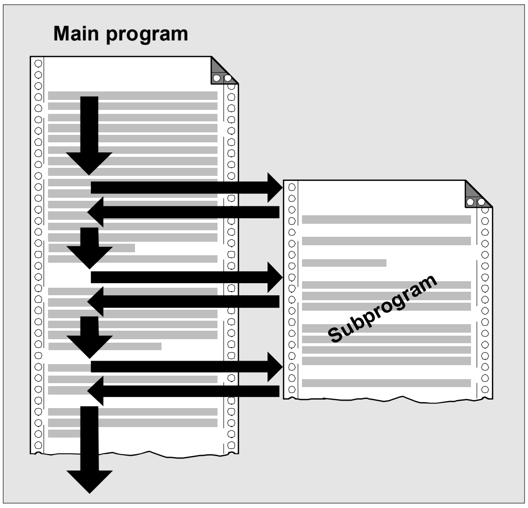
840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

11.1 Use of subprograms

What is a subprogram?

In principle, a subprogram has the same structure as a parts program. It consists of NC blocks with traversing and switching commands.

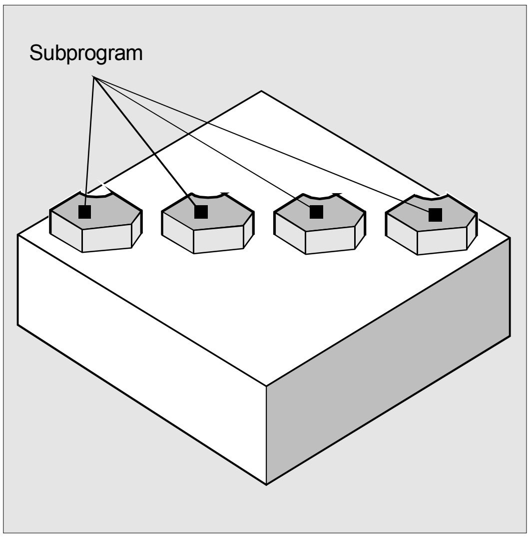
Basically, there is no difference between a main program and a subprogram. The subprogram contains either machining operations or sequences of operations that are to be performed several times.



Use of subprograms

Recurring machining sequences are programmed once in a subprogram. Examples include certain contour shapes, which occur repeatedly, and machining cycles.

The subprogram can be called and executed in any main program.



Structure of a subprogram

The structure of the subprogram is identical to that of the main program (see section on structure and content of an NC program).
Subprograms are **M17 end of program** provided.
This means a return to the program level where the subprogram was called from.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Explanation

It is possible to suppress this end of program **M17** in the machine data (e.g.: to achieve a better running time).

Additional notes

A program header with parameter definitions can also be programmed in the subprogram. You will find a more detailed description in the Programming Guide "Advanced".

End of subprogram with RET

The instruction **RET** can also be used in subprograms as a substitute for the backward jump with M17.
RET must be programmed in a separate block.
The RET statement is used when G64 continuous-path mode is **not** to be interrupted by a return. This is only possible if the subprogram has **no** SAVE attribute.

If M17 is programmed in a separate block, G64 is interrupted and an exact stop generated.
Remedy:
Do not write M17 in a subprogram block on its own, instead use it with, for example, a traverse path:
G1 X=YY M17 must be set in the machine data:
"No M17 from PLC".

Subprogram name

The subprogram is given a name allowing it to be selected from among the others. The name can be freely chosen during program creation, taking the following conventions into account:

- The first two characters must be letters.
- The others can be letters, numerals or underscore characters
- A maximum of 31 characters can be used
- No separators are to be used (see the section on "Language elements of the programming language")

840D NCU 571	840D NCU 572 NCU 573	810D	840Di

The same rules apply as to the names of main programs.

Example:
`N10 POCKET1`
It is also possible to use the address word L... in subprograms. The value can have 7 decimal places (integers only).
Note: With address L, leading zeros are meaningful for differentiation.

Example:

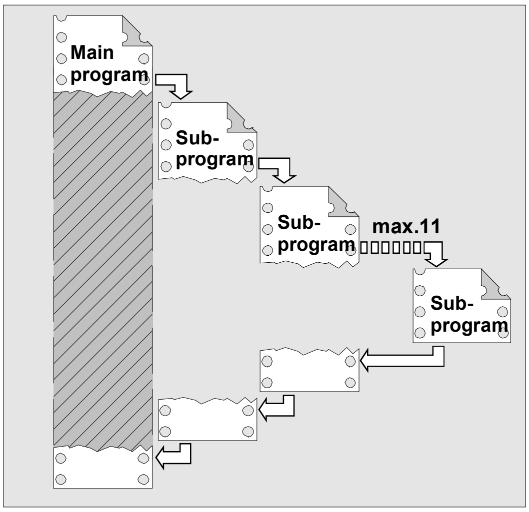
```
N10 L123
N20 L0123
N30 L00123
```

There are three different subprograms in this example.

Nesting depth

Subprograms can also be called from a subprogram, not only from a main program.
In total, up to **12 program levels** are available for this type of nested call, including the main program level.

This means:
Up to 11 nested subprogram calls can be issued from the main program.



Note:
If you are working with SIEMENS machining and measuring cycles, then three levels are required. If a cycle is to be called from a subprogram, this call cannot be issued from beyond level 9.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

11.2 Subroutine call

Subprogram call

You call the subprogram in the main program either with address L and the subprogram number or by specifying the subprogram name.

Example:

```
...
N120  L100
```

Subprogram call "L100.SPF":

```
N10 MSG (DIN subroutine")
N20 G1 G91...
...
N60 M17     ;End of subprogram
```

```
N160  M30
```

End of main program

Example with transfer of R parameter:

`N10   G0 X0 Y0 G90 T1`	Tool T1 in rapid traverse to the first position, absolute dimension
`N20   R10=10 R11=20`	Describe arithmetic parameters R10 and R11
`N30   RECTANGLE`	Call the rectangle subprogram "RECTANGLE.SPF" with transfer of R parameters:

```
N15 G1 X=R10 G91 F500
N25 Y=R11
N35 X=-R10
N45 Y=-R11
N55 M17    ;End of subprogram
```

`N40   G0 X50 Y50 G90`	Position tool at the next machining position
`N50   RECTANGLE`	Call the rectangle subprogram "RECTANGLE.SPF" with transfer of R parameters:
`N60   M30`	End of main program

11.2 Subroutine call

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

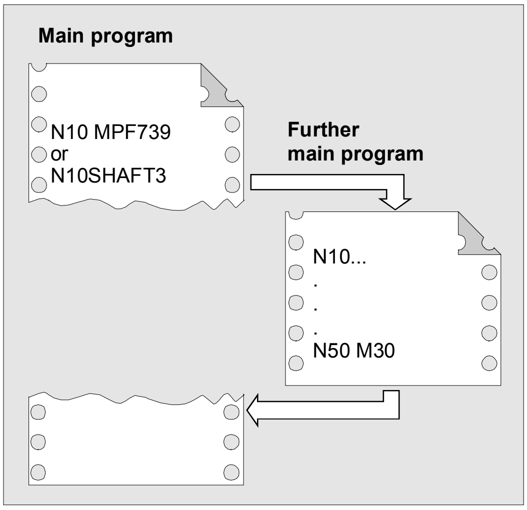
Call main program as subprogram

A main program can also be called as a subprogram. The end of program set in the main program M30 is evaluated as M17 in this case (end of program with return to calling program).

You program the call specifying the program name.

Example:

```
N10 MPF739 or
N10 SHAFT3
```



A subprogram can also be started as a main program.

Additional notes

Search strategy of the control:

1. Are there any *_MPF ?
2. Are there any *_SPF ?

This means: if the name of the subprogram to be called is identical to the name of the main program, the main program that issues the call is called again. This is generally an undesirable effect and must be avoided by assigning unique names to subprograms and main programs.

Call the subprograms with the INI file

Subprograms that do not require parameter assignment can be called from an initialization file:

Example:

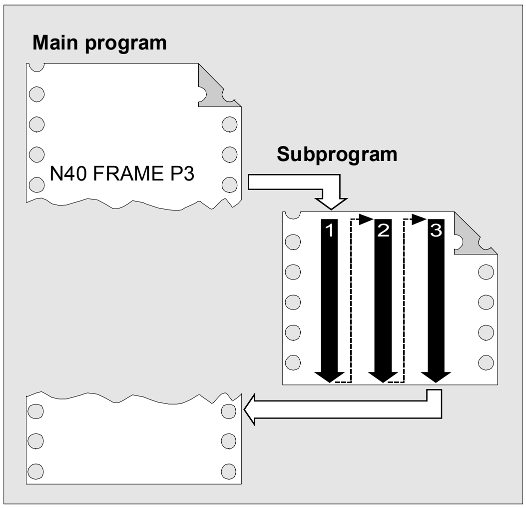
```
N10 MYINISUB1        ;Subprogram call
                     without parameters
```

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

11.3 Subprogram with program repetition

Program repetition, P

If a subprogram is to be executed several times in succession, the desired number of program repetitions can be entered at address P in the block with the subprogram call.



Example:

```
N40 FRAME P3
```

The subprogram FRAME must be executed 3 times in succession.

Value range

P: 1…9999

The following applies to every subprogram call:

The subprogram call must always be programmed in a separate NC block.

Subprogram call with program repetition and parameter transfer

Parameters are transferred only when the program is called, i.e. on the first run. The parameters remain unchanged for the remaining repetitions.

If you want to change the parameters during program repetitions, you must make the appropriate provision in the subprogram.

840D NCU 571	840D NCU 572 NCU 573	810D	840Di

11.4 Program section repetition (SW 4.3 and higher)

Function

As compared with subprogram technology, program section repetition allows the repetition of existing program sections in any combination.
The block or program sections to be repeated are identified by labels.

For more information on labels, please see:
References: /PG/, Programming Guide Fundamentals, Section 2.2
/PGA/, Programming Guide Advanced Section 1.11, 1.12

Explanation

LABEL:	Jump destination; the name of the jump destination is followed by a colon
REPEAT	Repeat (repeat several lines)
REPEATB	Repeat block (repeat one line only)

Programming

Repeat block

```
LABEL: xxx
yyy
REPEATB LABEL P=n
zzz
```

The program line identified by a label is repeated P=n times.
If P is not specified, the program section is repeated exactly once. After the last repetition, the program is continued at the line zzz following the REPEATB line.

The block identified by the label can appear before or after the REPEATB statement.
The search initially commences towards the start of the program.
If the label is not found in this direction, the search continues towards the end of the program.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

Repetition of positions

`N10 POSITION1: X10 Y20`	
`N20 POSITION2: CYCLE(0,,9,8)`	Position cycle
`N30 ...`	
`N40 REPEATB POSITION1 P=5`	Execute block N10 five times
`N50 REPEATB POSITION2`	Execute block N20 once
`N60 ...`	
`N70 M30`	

Programming

Repeat area starting at label

```
LABEL: xxx
yyy
REPEAT LABEL P=n
zzz
```

The program section between the label with any name and the REPEAT statement is repeated P=n times.
If the block with the label contains further statements, these are executed again on each repetition.
If P is not specified, the program section is repeated exactly once.
After the last repetition, the program is continued at the line zzz following the REPEAT line.

The label must appear before the REPEAT statement. The search is performed towards the start of the program only.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

5 squares with increasing width are to be produced

`N5 R10=15`	
`N10 Begin: R10=R10+1`	Width
`N20 Z=10-R10`	
`N30 G1 X=R10 F200`	
`N40 Y=R10`	
`N50 X=-R10`	
`N60 Y=-R10`	
`N70 Z=10+R10`	
`N80 REPEAT BEGIN P=4`	Execute area from N10 to N70 four times
`N90 Z10`	
`N100 M30`	

Programming

Repeat area between two labels

```
START_LABEL: xxx
ooo
END_LABEL: yyy
ppp
REPEAT START_LABEL END_LABEL P=n
zzz
```

The area between the two labels is repeated P=n times. User-defined names can be assigned to the labels.

The first line of the repetition contains the start label, the last line contains the end label. If the line containing the start or end label contains further statements, these are executed again on each pass. If P is not specified, the program section is repeated once. After the last repetition, the program is continued at the line zzz following the REPEAT line.

The program section to be repeated can appear before or after the REPEAT statement. The search initially commences towards the start of the program. If the start label is not found in this direction, the search resumes from the REPEAT statement towards the end of the program.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

It is not possible to nest the REPEAT statement with the two labels within parentheses. If the start label is found before the REPEAT statement and the end label is not reached before the REPEAT statement, the repetition is performed on the section between the start label and the REPEAT statement.

Programming example

Repeat program section from BEGIN to END

`N5 R10=15`	
`N10 Begin: R10=R10+1`	Width
`N20 Z=10-R10`	
`N30 G1 X=R10 F200`	
`N40 Y=R10`	
`N50 X=-R10`	
`N60 Y=-R10`	
`N70 END:Z=10`	
`N80 Z10`	
`N90 CYCLE(10,20,30)`	
`N100 REPEAT BEGIN END P=3`	Execute area from N10 to N70 three times
`N110 Z10`	
`N120 M30`	

Programming

Repeat an area between a label and the end label

```
LABEL: xxx
ooo
ENDLABEL: yyy
REPEAT LABEL P=n
zzz
```

ENDLABEL is a predefined label with a fixed name. ENDLABEL marks the end of a program section and can be used multiple times in the program.
The block marked by ENDLABEL can contain further statements.

11.4 Program section repetition (SW 4.3 and higher)

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

The area between a label and the following ENDLABEL is repeated P=n times. Any name can be used to define the start label.
If the block with the start label or ENDLABEL contains further statements, these are executed on each repetition.

If no ENDLABEL is found between the start label and the block with the REPEAT call, the loop ends before the REPEAT line. The construct therefore has the same effect as described above in "repeat area from label".
If P is not specified, the program section is repeated once.
After the last repetition, the program is continued at the line zzz following the REPEAT line.

Programming example

`N10 G1 F300 Z-10`	
`N20 **BEGIN1:**`	
`N30 X10`	
`N40 Y10`	
`N50 **BEGIN2:**`	
`N60 X20`	
`N70 Y30`	
`N80 **ENDLABEL:** Z10`	
`N90 X0 Y0 Z0`	
`N100 Z-10`	
`N110 **BEGIN3:** X20`	
`N120 Y30`	
`N130 **REPEAT BEGIN3** P=3`	Execute area from N110 to N120 three times
`N140 **REPEAT BEGIN2** P=2`	Execute area from N50 to N80 twice
`N150 M100`	
`N160 **REPEAT BEGIN1** P=2`	Execute area from N20 to N80 twice
`N170 Z10`	
`N180 X0 Y0`	
`N190 M30`	

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Preconditions

- Program section repetitions can be nested. Each call uses a subprogram level.
- If M17 or RET is programmed during processing of a program section repetition, the repetition is aborted. The program is resumed at the block following the REPEAT line.
- In the actual program display, the program section repetition is displayed as a separate subprogram level.
- If the level is canceled during the program section repetition, the program resumes at the point after the program section repetition call.

  Example:

```
N5 R10=15
N10 BEGIN: R10=R10+1          Width
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10                     Level cancellation
N50 X=-R10
N60 Y=-R10
N70 END: Z10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT BEGIN END P=3
N120 Z10                      Resume program processing
N130 M30
```

- Control structures and program section repetitions can be used in combination.
  There should be no overlap between the two, however.
  A program section repetition should appear within a control structure branch or a control structure should appear within a program section repetition.

11.4 Program section repetition (SW 4.3 and higher)

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

- If jumps and program section repetitions are mixed, the blocks are executed purely sequentially. For example, if a jump is performed from a program section repetition, processing continues until the programmed end of the program section is found.

  Example:

```
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X10
N40 Y10
N50 GOTOF BEGIN2
N60 ENDLABEL:
N70 BEGIN2:
N80 X20
N90 Y30
N100 ENDLABEL: Z10
N110 X0 Y0 Z0
N120 Z-10
N130 REPEAT BEGIN1 P=2
N140 Z10
N150 X0 Y0
N160 M30
```

Activation

Program section repetition is activated by programming.

The REPEAT instruction should be placed behind the traveling blocks.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programming example

Milling: Machine drill position with different technologies

`N10 CENTER DRILL()`	Load center drill
`N20 POS_1:`	Drill positions 1
`N30 X1 Y1`	
`N40 X2`	
`N50 Y2`	
`N60 X3 Y3`	
`N70 ENDLABEL:`	
`N80 POS_2:`	Drill positions 2
`N90 X10 Y5`	
`N100 X9 Y-5`	
`N110 X3 Y3`	
`N120 ENDLABEL:`	
`N130 DRILL()`	Change drill and drilling cycle
`N140 THREAD(6)`	Load tap M6 and threading cycle
`N150 REPEAT POS_1`	Repeat program section once from POS_1 up to ENDLABEL
`N160 DRILL()`	Change drill and drilling cycle
`N170 THREAD(8)`	Load tap M8 and threading cycle
`N180 REPEAT POS_2`	Repeat program section once from POS_2 up to ENDLABEL
`N190 M30`	

■

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Notes

Tables

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12.1 List of statements

Legend:

[1] Default setting at beginning of program (factory settings of the control, if nothing else programmed).

[2] The groups are numbered according to the table headed "List of G functions/preparatory functions" in Section 12.3

[3] Absolute end points: modal; incremental end points: non-modal; otherwise modal/non-modal (m, s) depending on syntax of G function.

[4] As arc centers, IPO parameters act incrementally. They can be programmed in absolute mode with AC. With other meanings (e.g. pitch), the address modification is ignored.

[5] The vocabulary word is not valid for SINUMERIK FM-NC/810D

[6] The vocabulary word is not valid for SINUMERIK FM-NC/810D/NCU571

[7] The vocabulary word is not valid for SINUMERIK 810D

[8] The OEM can add two extra interpolation types. The names can be changed by the OEM.

[9] The vocabulary word is only valid for SINUMERIK FM-NC

[10] Extended address notation cannot be used for these functions.

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
:	Block number - main block (see N)	0 ... 9999 9999 integers only, without signs	Special block identification - instead of N... ; this block should contain all statements for a complete subsequent machining section	e.g.. **:**20		
A	Axis	Real			m,s [3]	
A2 [5]	Tool orientation: Euler angles	Real			s	
A3 [5]	Tool orientation: Direction vector component	Real			s	
A4 [5]	Tool orientation for start of block	Real			s	
A5 [5]	Tool orientation for end of block; normal vector component	Real			s	
AC	Input of absolute dimensions	0, ..., 359.9999°		X=AC(100)	s	
ACC [5]	Axial acceleration	Real, w/o signs			m	
ACN	Absolute dimensions for rotary axes, approach position in negative direction			A=ACN(...) B=ACN(...) C=ACN(...)	s	
ACP	Absolute dimensions for rotary axes, approach position in positive direction			A=ACP(...) B=ACP(...) C=ACP(...)	s	
ADIS	Approximate distance for path functions G1, G2, G3, ...	Real, w/o signs			m	
ADISPOS	Approximate distance for rapid traverse G0	Real, w/o signs			m	
ALF	Angle tilt fast	Integer, w/o signs			m	

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
AMIRROR	Programmable mirroring (additive mirror)			AMIRROR X0 Y0 Z0 ; separate block	s	3
ANG	Contour angle				s	
AP	Angle polar	0, ..., ± 360°			m,s [3]	
AR	Aperture angle (angle circular)	0, ..., 360°			m,s [3]	
AROT	Programmable rotation (additive rotation)	Rotation around 1st geo. axis: -180° .. 180° 2nd geo. axis: -89.999° ... 90° 3rd geo. axis: -180° .. 180°		AROT X... Y... Z... ;Separate AROT RPL= block	s	3
AROTS	Programmable frame rotations with solid angles (additive rotation)			AROTS X... Y... AROTS Z... X... AROTS Y... Z... ;separate AROTS RPL= block	s	3
ASCALE	Programmable scaling (additive scale)			ASCALE X... Y... Z... ; separate block	s	3
ASPLINE	Akima spline				m	1
ATRANS	Additive programmable shift (additive translation)			ATRANS X... Y... Z... ; separate block	s	3
AX	Variable axis identifier	Real			m,s [3]	
AXCTSWE	Advance container axis			AXCTSWE(CT_i)		25
B	Axis	Real			m,s [3]	
B2 [5]	Tool orientation: Euler angles	Real			s	
B3 [5]	Tool orientation: Direction vector component	Real			s	
B4 [5]	Tool orientation for start of block	Real			s	
B5 [5]	Tool orientation for end of block; normal vector component	Real			s	
BAUTO	Definition of first spline segment by the following 3 points (begin not a knot)				m	19
BNAT [1]	Natural transition to first spline block (begin natural)				m	19
BRISK [1]	Brisk path acceleration				m	21
BRISKA	Switch on brisk path acceleration for the programmed axes					
BSPLINE	B spline				m	1
BTAN	Tangential transition to first spline block (begin tangential)				m	19
C	Axis	Real			m,s [3]	
C2 [5]	Tool orientation: Euler angles	Real			s	
C3[5]	Tool orientation: Direction vector component	Real			s	

12.1 List of statements

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
C4 [5]	Tool orientation for start of block	Real			s	
C5 [5]	Tool orientation for end of block; normal vector component	Real			s	
CDOF [1]	Collision detection OFF				m	23
CDON	Collision detection ON				m	23
CDOF2	Collision detection OFF		For CUT3DC only		m	23
CFC [1]	Constant feed at contour				m	16
CFTCP	Constant feed in tool edge reference point (center-point path)				m	16
CFIN	Constant feed at internal radius only, not at external radius				m	16
CHF SW 3.5 and higher CHR	Chamfer; value = length of chamfer Chamfer; value = width of chamfer in direction of movement (chamfer)	Real, w/o signs			S	
CHKDNO	Check for unique D numbers					
CIP	Circular interpolation through intermediate point			CIP X... Y... Z... I1=... J1=... K1=...	m	1
CLGOF	Const. workpiece speed for centerless grinding OFF					
CLGON	Const. workpiece speed for centerless grinding ON					
COMPOF [1,6]	Compressor OFF				m	30
COMPON [6]	Compressor ON				m	30
COMPCURV	Compressor ON: polynomials with constant curvature				m	30
COMPCAD	Compressor ON: Surface quality CAD program				m	30
CP	**C**ontinuous **p**ath; path motion				m	49
CPRECOF [1,6]	Programmable contour precision OFF				m	39
CPRECON [6]	Programmable contour precision ON				m	39
CR	Circle radius	Real, w/o signs			S	
CROTS	Programmable frame rotations with solid angles (rotation in the specified axes)			CROTS X... Y... CROTS Z... X... CROTS Y... Z... ;separate CROTS RPL= block	S	
CSPLINE	Cubic spline				m	1
CT	Circle with tangential transition			CT X... Y.... Z...	m	1
CUT2D [1]	2 ½D cutter compensation type 2-dimensional				m	22
CUT2DF	2 ½D cutter compensation type 2-dimensional frame; Tool compensation is effective in relation to the current frame (inclined plane)				m	22
CUT3DC [5]	3D cutter compensation type 3-dimensional circumference milling				m	22
CUT3DCC [5]	Cutter compensation type 3-dimensional circumference milling with limit surfaces				m	22

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
CUT3DCCD [5]	Cutter compensation type 3-dimensional circumference milling with limit surfaces with differential tool				m	22
CUT3DF [5]	3D cutter compensation type 3-dimensional face milling				m	22
CUT3DFF [5]	3D cutter compensation type 3-dimensional face milling with constant tool orientation dependent on the current frame				m	22
CUT3DFS [5]	3D cutter compensation type 3-dimensional face milling with constant tool orientation independent of the current frame				m	22
CUTCONOF [1]	Constant radius compensation OFF				m	40
CUTCONON	Constant radius compensation ON				m	40
D	Tool offset number	1, ..., 9 in SW 3.5 and higher 1, ... 32 000	contains offset data for a specific tool T... ; D0 → Offset values for a tool	D...		
DC	Absolute dimensions for rotary axes, approach position directly			A=DC(...) B=DC(...) C=DC(...) SPOS=DC(...)	s	
DIAMCYCOF	Radius programming for G90/G91: ON. The G-code of this group that was last active remains active for display		Radius programming last active G-code		m	29
DIAMOF [1]	Diameter programming: OFF		Radius programming for G90/G91		m	29
DIAMON	Diametral programming: ON		Diameter progr. for G90/G91		m	29
DIAM90	Diametral programming: for G90, radius progr. For G91				m	29
DILF	Length for lift fast				m	
DISC	Transition circle overshoot - radius compensation	0, ..., 100			m	
DISPR	Distance for repositioning	Real, w/o signs			S	
DISR	Distance for repositioning	Real, w/o signs			S	
DITE	Thread run-out path	Real			m	
DITS	Thread run-in path	Real			m	
DL	Total tool offset	INT			m	
DRFOF	Deactivate the handwheel offsets (DRF)				m	
DRIVE [9]	Velocity-dependent path acceleration				m	21
EAUTO	Definition of last spline section by the last 3 points (end not a knot)				m	20
ENAT [1]	Natural transition to next traversing block (end natural)				m	20
ETAN	Tangential transition to next traversing block at spline end (end tangential)				m	20

12.1 List of statements

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
F	Feed value (in conjunction with G4 the dwell time is also programmed in F)	0.001, ..., 99 999.999	Tool/workpiece path feedrate; unit of measurement in mm/min or mm/rev dependent on G94 or G95	F=100 G1 ...		
FA	Axial feed	0.001, ..., 999999.999 mm/min, degr./min; 0.001, ..., 39999.9999 inch/min		FA[X]=100	m	
FCUB [6]	Feedrate variable according to cubic spline (feed cubic)				m	37
FD	Path feed for handwheel override (feed DRF)	Real, w/o signs			S	
FDA	Axial feed for handwheel override (feed DRF axial)	Real, w/o signs			S	
FENDNORM	Corner deceleration OFF				m	57
FFWOF [1]	Feedforward control OFF (feed forward OFF)				m	24
FFWON	Feedforward control ON (feed forward ON)				m	24
FGREF	Reference radius				m	
FGROUP	Definition of axis/axes with path feed		F applies for axes specified under FGROUP	FGROUP (axis1, [axis2], ...)		
FIFOCTRL	Preprocessing memory control				m	4
FL	Speed limit for synchronized axes (feed limit)	Real, w/o signs	The unit set with G93, G94, G95 is applicable (max. rapid traverse)	FL [axis] =...	m	
FLIN [6]	Feed linear variable (feed linear)				m	37
FMA	Feed multiple axial	Real, w/o signs			m	
FNORM [1,6]	Feed normal to DIN 66025				m	37
FORI1	Feed for rotating orientation vector on large circle				m	
FORI2	Feed for rotation superimposed on rotated orientation vector				m	
FP	Fixed point: number of fixed point to be approached	Integer, w/o signs		G75 FP=1	S	
FPR	Identification for rotary axis	0.001 ... 999999.999		FPR (rotary axis)		
FPRAOF	Deactivate revolutional feedrate					
FPRAON	Activate revolutional feedrate					
FRC	Feed for radius and chamfer				s	
FRCM	Feed for radius and chamfer, modal				m	
FTOCOF [1,6]	Online fine tool offset OFF				m	33
FTOCON [6]	Online fine tool offset ON				m	33

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
FXS	Travel to fixed stop ON	Integer, w/o signs	1 = select, 0 = deselect		m	
FXST	Torque limit for travel to fixed stop (fixed stop torque)	%	Parameter optional		m	
FXSW	Monitoring window for travel to fixed stop (fixed stop window)	mm, inches or degrees	Parameter optional			
G functions						
G	G function (preparatory function) The G functions are divided into G groups. Only one G function from one group can be written in one block. A G function can either be modal (until canceled by another function from the same group), or non-modal (only effective for the block it is written in).	Only predefined, integer values		G...		
G0	Linear interpolation with rapid traverse (rapid traverse motion)		Motion	G0 X... Z...	m	1
G1[1]	Linear interpolation with feedrate (linear interpolation)		commands	G1 X... Z... F...	m	1
G2	Circular interpolation clockwise			G2 X... Z... I... K... F... ; center and end points G2 X... Z... CR=... F... ; radius and end point G2 AR=... I... K... F... ; arc angle and center point G2 AR=... X... Z... F... ; arc angle and end point	m	1
G3	Circular interpolation counterclockwise			G3 ... ; otherwise as for G2	m	1
G4	Predefined dwell time		Special motion	G4 F... ; Dwell time in s, or G4 S... ; Dwell time in spindle revolutions. ; separate block	s	2
G5	Oblique plunge-cut grinding		Oblique plunge-cutting		s	2
G7	Compensatory motion during oblique plunge-cut grinding		Start position		s	2
G9	Exact stop – deceleration				s	11
G17[1]	Selection of working plane X/Y		Infeed direction Z		m	6
G18	Selection of working plane Z/X		Infeed direction Y		m	6
G19	Selection of working plane Y/Z		Infeed direction X		m	6
G25	Lower working area limitation		Value assignments in	G25 X.. Y.. Z.. ; separate block	s	3
G26	Upper working area limitation		channel axes	G26 X.. Y.. Z.. ; separate block	s	3

12.1 List of statements

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
G33	Thread interpolation with constant pitch	0.001, ..., 2000.00 mm/rev	Motion command	G33 Z... K... SF=... ; cylinder thread G33 X... I... SF=... ; cross thread G33 Z... X... K... SF=... ; taper thread (in Z axis path greater than in X axis) G33 Z... X... I... SF=... ; taper thread (in X axis path greater than in Z axis)	m	1
G34	Linear degressive speed change [mm/rev^2]		Motion command	G34 X.. Y.. Z.. I.. J.. K.. F..	m	1
G35	Linear progressive speed change [mm/rev^2]		Motion command	G35 X.. Y.. Z.. I.. J.. K.. F..	m	1
G40 [1]	Tool radius compensation OFF				m	7
G41	Tool radius compensation to left of contour				m	7
G42	Tool radius compensation to right of contour				m	7
G53	Suppression of current zero offset (non-modal)		incl. programmed offsets		s	9
G54	1st settable zero offset				m	8
G55	2nd settable zero offset				m	8
G56	3rd settable zero offset				m	8
G57	4th settable zero offset				m	8
G58	Axial programmable zero offset, absolute				s	3
G59	Axial programmable zero offset, additive				s	3
G60 [1]	Exact stop - deceleration				m	10
G62	Corner deceleration at inside corners when tool radius offset is active (G41, G42)		Only in conjunction with continuous-path mode	G62 Z... G1	m	57
G63	Tapping with compensating chuck			G63 Z... G1	s	2
G64	Exact stop - continuous-path mode				m	10
G70	Dimension in inches (lengths)				m	13
G71 [1]	Metric dimension (lengths)				m	13
G74	Reference point approach			G74 X... Z...; separate block	s	2
G75	Fixed point approach		Machine axes	G75 FP=.. X1=... Z1=...; separate block	s	2
G90 [1]	Absolute dimensions			G90 X... Y... Z...(...) Y=AC(...) or X=AC Z=AC(...)	m s	14
G91	Incremental dimensions			G91 X... Y... Z... or X=IC(...) Y=IC(...) Z=IC(...)	m s	14
G93	Inverse-time feedrate rpm		Execution of a block: Time	G93 G01 X... F...	m	15
G94 [1]	Linear feedrate F in mm/min or inch/min and °/min				m	15
G95	Revolutional feedrate F in mm/rev or inches/rev				m	15
G96	Constant cutting speed (as for G95) ON			G96 S... LIMS=... F...	m	15
G97	Constant cutting speed (as for G95) OFF				m	15

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
G110	Polar programming relative to last programmed set position			G110 X.. Y.. Z..	s	3
G111	Polar programming relative to origin of current workpiece coordinate system			G110 X.. Y.. Z..	s	3
G112	Polar programming relative to last valid pole			G110 X.. Y.. Z..	s	3
G140 [1]	SAR approach direction defined by G41/G42				m	43
G141	SAR approach direction to left of contour				m	43
G142	SAR approach direction to right of contour				m	43
G143	SAR approach direction tangent-dependent				m	43
G147	Soft approach with straight line				s	2
G148	Soft retraction with straight line				s	2
G153	Suppress current frames including base frame		incl. system frame		s	9
G247	Soft approach with quadrant				s	2
G248	Soft retraction with quadrant				s	2
G290	Switch to SINUMERIK mode ON				m	47
G291	Switch to FANUC mode ON				m	47
G331	Tapping	±0.001, ...,	Motion		m	1
G332	Retraction (tapping)	2000.00 mm/rev	commands		m	1
G340 [1]	Spatial approach block (depth and in plane (helix))		Effective during soft approach/ retraction		m	44
G341	Initial infeed on perpendicular axis (z), then approach in plane		Effective during soft approach/ retraction		m	44
G347	Soft approach with semicircle				s	2
G348	Soft retraction with semicircle				s	2
G450 [1]	Transition circle		Tool radius comp.		m	18
G451	Intersection of equidistant paths		Resp. at corners		m	18
G460 [1]	Collision monitoring for approach and retraction block on				m	48
G461	Extend border block with arc if		no intersection in TRC block		m	48
G462	Extend border block with line if				m	48
G500 [1]	Deactivate all settable frames if G500 does not contain a value				m	8
G505 .... G599	5. ... 99. Settable zero offset				m	8
G601 [1]	Block change on exact stop fine		Only effective with		m	12
G602	Block change at stop exact coarse		active G60 or G9		m	12
G603	Block change at IPO - end of block		with program-		m	12
G641	Exact stop - continuous-path mode		mable transition	G641 ADIS=...	m	10
G642	Corner rounding with axial precision		rounding		m	10
G643	Block-internal corner rounding				m	10
G644	Corner rounding with specified axis dynamics				m	10
G621	Corner deceleration at all corners		Only in conjunc-tion with contin-uous-path mode	G621 ADIS=...	m	57

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
G700	Dimension in inches and inch/min (lengths + velocities + system variable)				m	13
G710 [1]	Metric dimension in mm and mm/min (lengths + velocities + system variable)				m	13
G810[1], ..., G819	G group reserved for the OEM					31
G820[1], ..., G829	G group reserved for the OEM					32
G931	Feedrate specified by travel time		Travel time		m	15
G942	Freeze linear feedrate and constant cutting rate or spindle speed				m	15
G952	Freeze revolutional feedrate and constant cutting rate or spindle speed				m	15
G961	Constant cutting speed (as for G94) ON			G961 S... LIMS=... F...	m	15
G962	Linear or revolutional feedrate and constant cutting rate				m	15
G971	Constant cutting speed (as for G94) OFF				m	15
G972	Freeze linear or revolutional feedrate and constant spindle speed				m	15
GOTOF	Jump forwards (towards the end of the program)					
GOTOB	Jump backwards (towards the start of the program)					
GWPSOF	Deselect constant grinding wheel peripheral speed (GWPS)			GWPSOF(T No.)	s	
GWPSON	Select constant grinding wheel peripheral speed (GWPS)			GWPSON (T No.)	s	
H...	Auxiliary function output to PLC	Real/INT Program: REAL : ±3.4028ex38 INT: -2147483648 +2147483648 Display: ±999 999 999.9999	Can be set by MD (machine manufacturer)	H100 or H2=100		
I [4]	Interpolation parameters	Real			s	
I1	Intermediate point coordinate	Real			s	
IC	Input of incremental dimensions	0, ..., ±99999.999°		X=IC(10)	s	
INCW	Travel on a circle involute in CW direction with interpolation of involute by G17/G18/G19	Real	End point: Center point: Radius with CR > 0: Angle of rotation in degrees between start and end vectors	INCW/INCCW X... Y... Z... INCW/INCCW I... J... K... INCW/INCCW CR=... AR...	m	1
INCCW	Travel on a circle involute in CCW direction with interpolation of involute by G17/G18/G19	Real		Direct programming: INCW/INCCW I... J... K... CR=... AR=...	m	1
ISD	Insertion depth	Real			m	
J [4]	Interpolation parameters	Real			s	
J1	Intermediate point coordinate	Real			s	

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
K [4]	Interpolation parameters	Real			s	
K1	Intermediate point coordinate	Real			s	
KONT	Travel round contour on tool offset				m	17
L	Subprogram number	Integer, up to 7 places		L10	s	
LEAD [5]	Lead angle	Real			m	
LFOF [1]	Interruption of thread cutting OFF				m	41
LFON	Interruption of thread cutting ON				m	41
LFPOS	Axial retraction to a position				m	46
LFTXT [1]	Tangential tool direction on retraction				m	46
LFWP	Non-tangential tool direction on retraction				m	46
LIMS	Spindle speed limitation (Limit Spindle Speed) with G96	0.001 ... 99 999.999			m	
M...	Switching operations	INT Display: 0, ..., 999 999 999 Program: 0;...; 2147483647	Up to 5 unassigned M functions can be assigned by the machine manufacturer			
M0 [10]	Programmed stop					
M1 [10]	Optional stop					
M2 [10]	End of main program with return to beginning of program					
M3	Direction of spindle rotation clockwise for master spindle					
M4	Direction of spindle rotation counterclockwise for master spindle					
M5	Spindle stop for master spindle					
M6	Tool change					
M17 [10]	End of subprogram					
M19	For SSL accumulated spindle programming					
M30 [10]	End of program, same effect as M2					
M40	Automatic gear change					
M41... M45	Gear stage 1, ..., 5					
M70	Transition to axis mode					
MEAC	Continuous measurement without deleting distance-to-go	Integer, w/o signs			S	
MEAS	Measure with touch-trigger probe	Integer, w/o signs			S	
MEASA	Measurement with deletion of distance-to-go				s	
MEAW	Measure with touch-trigger probe without deleting distance-to-go	Integer, w/o signs			S	
MEAWA	Measurement without deleting distance-to-go				s	

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
MIRROR	Programmable mirror			MIRROR X0 Y0 Z0 ; separate block	s	3
MOV						
MSG	Programmable messages			MSG("message")	m	
N	Block number - subblock	0, ..., 9999 9999 integers only, w/o signs	Can be used to identify blocks by means of a number; written at beginning of block	e.g. N20		
NORM [1]	Standard setting in starting point and end point with tool offset				m	17
OEMIPO1 [6,8]	OEM interpolation 1				m	1
OEMIPO2 [6,8]	OEM interpolation 2				m	1
OFFN	Allowance on the programmed contour			OFFN=5		
OMA1 [6]	OEM address 1	Real			m	
OMA2 [6]	OEM address 2	Real			m	
OMA3 [6]	OEM address 3	Real			m	
OMA4 [6]	OEM address 4	Real			m	
OMA5 [6]	OEM address 5	Real			m	
OFFN	Offset - normal	Real			m	
ORIC [1,6]	Orientation changes at outside corners are superimposed on the circle block to be inserted (orientation change continuously)				m	27
ORID [6]	Orientation changes are performed before the circle block (orientation change discontinuously)				m	27
ORIAXPOS	Orientation angle via virtual orientation axes with rotary axis positions				m	50
ORIEULER	Orientation angle via Euler angle				m	50
ORIAXES	Linear interpolation of machine axes or orientation axes		Final orientation: Vector specification A3, B3, C3 or Euler/RPY angle A2, B2, C2 Additional inputs: Rotational vectors A6, B6, C6 Arc angle of taper in degrees 0 < SLOT <180 degrees Intermediate vectors: A7, B7, C7 2nd contact point of tool: XH, YH, ZH,	Parameter settings as follows: Direction vectors normalized A6=0 B6=0 C6=1 Arc angle implemented as travel angle with SLOT=... SLOT=+... at ≤ 180 degrees SLOT= -... at ≥ 180 degrees Intermediate orientation normalized A7=0 B7=0 C7=1	m	51
ORICONCW	Interpolation on a circular peripheral surface in CW direction				m	51
ORICONCCW	Interpolation on a circular peripheral surface in CCW direction				m	51
ORICONIO	Interpolation on a circular peripheral surface with intermediate orientation setting				m	51
ORICONTO	Interpolation on a circular peripheral surface in tangential transition				m	51
ORICURVE	Interpolation of orientation with specification of motion of two contact points of tool				m	51
ORIPLANE	Interpolation in a plane (corresponds to ORIVECT) large-radius circular interpolation				m	51
ORIPATH	Tool orientation trajectory referred to path		Transformation package handling, see /FB/, TE4		m	51
ORIROTA	Angle of rotation in relation to absolute direction of rotation				m	54

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
ORIROTR	Angle of rotation relative to plane between start and end orientations				m	54
ORIROTT	Angle of rotation relative to change in orientation vector				m	54
ORIRPY	Orientation angle via RPY angle				m	50
ORIS [5]	Orientation modification (orientation smoothing factor)	Real	Referring to the path		m	
ORIVECT	Large-radius circular interpolation (identical to ORIPLANE)				m	51
ORIVIRT1	Orientation angle via virtual orientation axes (definition 1)				m	50
ORIVIRT2	Orientation angle via virtual orientation axes (definition 1)				m	50
ORIMKS [6]	Tool orientation in the machine coordinate system				m	25
ORIWKS [1,6]	Tool orientation in the workpiece coordinate system				m	25
OS	Oscillation on/off	Integer, w/o signs				
OSC [6]	Continuous tool orientation smoothing				m	34
OSCILL	Axis assignment for oscillation - activate oscillation		Axis: 1–3 infeed axes		m	
OSCTRL	Oscillation control options	Integer, w/o signs			m	
OSE	Oscillating: End point				m	
OSNSC	Oscillating: Number of spark-out cycles number spark out cycles)				m	
OSOF [1,6]	Tool orientation smoothing OFF				m	34
OSP1	Oscillating: Left reversal point (oscillating: position 1)	Real			m	
OSP2	Oscillating: Right reversal point (oscillating: position 2)	Real			m	
OSS [6]	Tool orientation smoothing at end of block				m	34
OSSE [6]	Tool orientation smoothing at start and end of block				m	34
OST1	Oscillating: Stop at left reversal point	Real			m	
OST2	Oscillating: Stop at right reversal point	Real			m	
OVR	Speed override	1, ..., 200%			m	
OVRA	Axial speed override	1, ..., 200%			m	
P	Number of subprogram passes	1 ... 9999, integers w/o signs		e.g. L781 P... ; separate block		
PAROTOF	Deactivate workpiece-related frame rotation				m	52
PAROT	Align workpiece coordinate system on workpiece				m	52
PDELAY-OF [6]	Punch with delay OFF				m	36
PDELAY-ON [1,6]	Punch with delay ON				m	36
PL	Parameter interval length	Real, w/o signs			S	
POLY [5]	Polynomial interpolation				m	1
PON [6]	Punch ON				m	35

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
PONS [6]	Punch ON in IPO cycle (punch ON slow)				m	35
POS	Position axis			POS[X]=20		
POSA	Position axis across block boundary			POSA[Y]=20		
POLF	LIFTFAST position				m	
PRESETON	Sets the actual value for programmed axes		An axis identifier is programmed with the corresponding value in the next parameter. Up to 8 axes possible	PRESETON(X,10,Y,4.5)		
PTP	**P**oint **t**o **p**oint		Synchronized axis		m	49
PUTFTOC	PutFineToolCorrection: Fine tool correction for continuous dressing					
PUTFTOCF	PutFineToolCorrectionFunctionDependent: Fine tool correction depending on a function defined by FCtDEF for continuous dressing					
PW	Point weight	Real, w/o signs			S	
R...	Arithmetic parameters SW 5 and higher: also as settable address identifier and with numerical extension	±0.0000001, ..., 9999 9999	Number of R parameters can be set by MD	R10=3 ;R parameter assignment X=R10 ;axis value R[R10]=6 ;indirect prog.		
REPOSA	Repositioning linear all axes: Linear repositioning with all axes				s	2
REPOSH	Repositioning semicircle: Repositioning in semicircle				s	2
REPOSHA	Repositioning semicircle all axes: Repositioning with all axes; geometry axes in semicircle				s	2
REPOSL	Repositioning linear: Linear repositioning				s	2
REPOSQ	Repositioning quarter-circle: Return to contour in a quarter-circle				s	2
REPOSQA	Repositioning quarter-circle all axes: Return to contour linear all axes; geometry axes in quarter-circle				s	2
RET	End of subprogram		Use in place of M17 – without function output to PLC	RET		
RMB	Repositioning at beginning of block (Repos mode begin of block)				m	26
RME	Repositioning at end of block (Repos mode end of block)				m	26
RMI [1]	Repositioning at interruption point (Repos mode interrupt)				m	26
RMN	Reapproach to nearest path point (Repos mode of nearest orbital block)				m	26

840 D
NCU 571

840 D
NCU 572
NCU 573

810D

840Di

Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
RND	Round contour corner	Real, w/o signs		RND=...	s	
RNDM	Modal rounding	Real, w/o signs		RNDM=... RNDM=0: switch off m.r.	m	
ROT	Programmable rotation	Rotation around 1st geometry axis: -180° .. 180° 2nd geo axis: -89.999°, ..., 90° 3rd geo axis: -180° .. 180°		ROT X... Y... Z... ROT RPL= ;separate block	s	3
ROTS	Programmable frame rotations with solid angles (rotation)			ROTS X... Y... ROTS Z... X... ROTS Y... Z... ;separate ROTS RPL= block	s	3
RP	Polar radius	Real			m,s [3]	
RPL	Rotation in the plane	Real, w/o signs			S	
RTLION	G0 with linear interpolation				m	55
RTLIOF	G0 without linear interpolation (single-axis interpolation)				m	55
S	Spindle speed or (with G4, G96) other meaning	REAL Display: ±999 999 999.9999 Program: ±3.4028ex38	Spindle speed in rpm G4: dwell time in spindle revolutions G96: cutting speed in m/min	S...: speed for master spindle S1...: speed for spindle 1	m, s	
SCALE	Programmable scaling (scale)			SCALE X... Y... Z... ; separate block	s	3
SD	Spline degree	Integer, w/o signs			S	
SETMS	Reset to the master spindle defined in machine data					
SETMS(n)	Set spindle n as master spindle					
SF	Starting point offset for thread cutting (spline offset)	0.0000, ..., 359.999°			m	
SOFT	Soft acceleration with jerk limitation				m	21
SON [6]	Nibbling ON (stroke ON)				m	35
SONS [6]	Nibbling ON in IPO cycle (stroke ON slow)				m	35
SPATH [1]	Path reference for FGROUP axes is arc length				m	45
SPCOF	Switch master spindle or spindle(s) from speed control to position control			SPCON SPCON (n)		
SPCON	Switch master spindle or spindle(s) from position control to speed control			SPCON SPCON (n)		
SPIF1 [1,6]	Fast NCK inputs/outputs for punching/nibbling byte 1 (stroke/punch interface 1)				m	38
SPIF2 [6]	Fast NCK inputs/outputs for punching/nibbling byte 2 (stroke/punch interface 2)				m	38

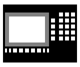
840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

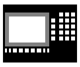
Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
SPLINE-PATH	Define spline grouping		Max. of 8 axes			
SPOF [1,6]	Stroke OFF, punching, nibbling OFF				m	35
SPN [6]	Number of path sections per block (stroke/punch number)	Integer			s	
SPP [6]	Length of path section (stroke/punch path)	Integer			m	
SPOS	Spindle position			SPOS=10 or SPOS[n]=10	m	
SPOSA	Spindle position across block boundaries			SPOSA=5 or SPOSA[n]=5	m	
SR	Retraction path (sparking out retract path)	Real, w/o signs			S	
SRA	Retraction path axial with external input (sparking out retract)			SRA[Y]=0.2	m	
ST	Sparking out time	Real, w/o signs			S	
STA	Sparking out time axial				m	
STAT	Position of joints	Integer			s	
STARTFIFO [1]	Execute; simultaneously fill preprocessing memory				m	4
STOPFIFO	Stop machining; fill preprocessing memory until STARTFIFO is detected, FIFO full or end of program				m	4
SUPA	Suppression of current zero offset, including programmed offsets, system frames, handwheel offsets (DRF), external zero offset and overlaid motion				s	9
T	Call tool (only change if specified in machine data; otherwise M6 command necessary)	1 ... 32 000	Call via T no.: or via tool identifier:	e.g. T3 or T=3 e.g. T="DRILL"		
TCARR	Request toolholder (number "m")	Integer	m=0: deselect active toolholder	TCARR=1		
TCOABS [1]	Determine tool length components from the current tool orientation		Necessary after reset, e.g. through manual setting		m	42
TCOFR	Determine tool length components from the orientation of the active frame				m	42
TCOFRX	Determine tool orientation of an active frame on selection of tool, tool points in X direction		Tool perpendicular to inclined surface		m	42
TCOFRY	Determine tool orientation of an active frame on selection of tool, tool points in Y direction		Tool perpendicular to inclined surface		m	42
TCOFRZ	Determine tool orientation of an active frame on selection of tool, tool points in Z direction		Tool perpendicular to inclined surface		m	42
TILT [5]	Tilt angle	Real			m	
TMOF	Deselect tool monitoring		T no. is only necessary if the tool with this number is not active.	TMOF (T no.)		
TMON	Select tool monitoring		T No. = 0: Deactivate monitoring for all tools	TMON (T no.)		

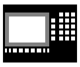
840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

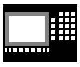
Name	Definition	Value assignment	Description, comment	Syntax	Modal/ non-modal	Group [2]
TOFRAME	Set current programmable frame to tool coordinate system		Frame rotation in tool direction		m	53
TOFRAMEX	X axis parallel to tool direction, secondary axis Y, Z				m	53
TOFRAMEY	Y axis parallel to tool direction, secondary axis Z, X				m	53
TOFRAMEZ	Z axis parallel to tool direction, secondary axis X, Y				m	53
TOROTOF	Frame rotations in tool direction OFF				m	53
TOROT	Z axis parallel to tool orientation		Frame rotations ON Rotation component of programmable frame		m	53
TOROTX	X axis parallel to tool orientation				m	53
TOROTY	Y axis parallel to tool orientation				m	53
TOROTZ	Z axis parallel to tool orientation				m	53
TOWSTD	Initial setting value for offsets in tool length		Inclusion of tool wear		m	56
TOWBCS	Wear values in basic coordinate system (BCS)				m	56
TOWKCS	Wear values in the coordinate system of the tool head for kinetic transformation (differs from MCS by tool rotation)				m	56
TOWMCS	Wear values in machine coordinate system (MCS)				m	56
TOWTCS	Wear values in the tool coordinate system (tool carrier ref. point T at the tool holder)				m	56
TOWWCS	Wear values in workpiece coordinate system (WCS)				m	56
TRAFOOF	Deactivate transformation			TRAFOOF( )		
TRANS	Programmable translation			TRANS X... Y... Z... ; separate block	s	3
TU	Axis angle	Integer		TU=2	s	
TURN	Number of turns for helix	0, ..., 999			s	
UPATH	Path reference for FGROUP axes is curve parameter				m	45
WAITM	Wait for marker in specified channel; end previous block with exact stop			WAITM(1,1,2)		
WAITMC	Wait for marker in specified channel; exact stop only if the other channels have not yet reached the marker			WAITMC(1,1,2)		
WAITP	Wait for end of traversing			WAITP(X) ; separate block		
WAITS	Waiting to reach spindle position			WAITS (main spindle) WAITS (n,n,n)		
WALIMOF	Working area limitation OFF			; separate block	m	28
WALIMON[1]	Working area limitation ON			; separate block	m	28
X	Axis	Real			m,s [3]	
Y	Axis	Real			m,s [3]	
Z	Axis	Real			m,s [3]	

12.1 List of statements


840 D
NCU 571


840 D
NCU 572
NCU 573


810D


840Di

Legend:

[1] Default setting at beginning of program (factory settings of the control, if nothing else programmed).

[2] The group numbering corresponds to the table "Overview of statements" in Section 11.3

[3] Absolute end points: modal; incremental end points: non-modal; otherwise modal/non-modal (m, s) depending on syntax of G function.

[4] As arc centers, IPO parameters act incrementally. They can be programmed in absolute mode with AC. With other meanings (e.g. pitch), the address modification is ignored.

[5] The vocabulary word is not valid for SINUMERIK FM-NC/810D

[6] The vocabulary word is not valid for SINUMERIK FM-NC/810D/NCU571

[7] The vocabulary word is not valid for SINUMERIK 810D

[8] The OEM can add two extra interpolation types. The names can be changed by the OEM.

[9] The vocabulary word is only valid for SINUMERIK FM-NC

[10] Extended address notation cannot be used for these functions.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12.2 List of addresses

12.2.1 Address letters

Letter	Definition	Numeric extension
A	Settable address identifier	x
B	Settable address identifier	x
C	Settable address identifier	x
D	Selection/deselection of tool length compensation, tool cutting edge	
E	Settable address identifier	
F	Feed Dwell time is seconds	x
G	G function	
H	H function	x
I	Settable address identifier	x
J	Settable address identifier	x
K	Settable address identifier	x
L	Subprograms, subprogram call	
M	M function	x
N	Subblock number	
O	Not assigned	
P	Program pass number	
Q	Settable address identifier	x
R	Variable identifier (arithmetic parameter) / variable address identifier without numerical extension	x
S	Spindle value Dwell time in spindle revolutions	x x
T	Tool number	x
U	Settable address identifier	x
V	Settable address identifier	x
W	Settable address identifier	x
X	Settable address identifier	x
Y	Settable address identifier	x
Z	Settable address identifier	x
%	Start character and separator for file transfer	
:	Main block number	
/	Skip identifier	

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12.2.2 Fixed addresses

Address identifier	Address type	Modal/ non-modal (m/s)	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Data type
L	Subprogram number	s									Integer without sign
P	Subprogram pass number	s									Integer without sign
N	Block number	s									Integer without sign
G	G function	See list of G functions									Integer without sign
F	Feed, dwell time	m, s		x						x	Real without sign
OVR	Override	m									Real without sign
S	Spindle, dwell time	m, s								x	Real without sign
SPOS	Spindle position	m				x	x	x			Real
SPOSA	Spindle position across block boundaries	m				x	x	x			Real
T	Tool number	m								x	Integer without sign
D	Offset number	m								x	Integer without sign
M, H,	Auxiliary functions	s								x	M: Integer without sign H: Real

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12.2.3 Fixed addresses with axis extension

Address identifier	Address type	Modal/ non-modal (m/s)	G70/ G71	G700/ G7100	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Data type
AX: Axis	Variable axis identifier	*)	x	x	x	x	x	x			Real
IP: Interpolation Parameter	Variable interpolation parameter	s	x	x	x	x	x				Real
POS: Positioning axis	Positioning axis	m	x	x	x	x	x	x	x		Real
POSA: Positioning axis above end of block	Positioning axis across block boundaries	m	x	x	x	x	x	x	x		Real
POSP: Positioning axis in parts	Positioning axis in parts (oscillation)	m	x	x	x	x	x	x			Real: end position/ Real: partial length Integer: option
PO: Polynomial [1)]	Polynomial coefficient	s	x	x							Real without sign 1 - 8 times
FA: Feed axial	Axial feed	m		x						x	Real without sign
FL: Feed limit	Axial feed limit	m		x							Real without sign
OVRA: Override	Axial override	m									Real without sign
ACC [2)]: Acceleration axial	Axial acceleration	m									Real without sign
FMA: Feed multiple axial	Synchronized feed axial	m		x							Real without sign
STA: Sparking-out time axial	Sparking out time axial	m									Real without sign
SRA: Sparking out retract	Retraction path on external input axial	m	x	x							Real without sign
OS: Oscillating on/off	Oscillation on/off	m									Integer without sign
OST1: Oscillating time 1	Stopping time at left reversal point (oscillation)	m									Real
OST2: Oscillating time 2	Stopping time at right reversal point (oscillation)	m									Real
OSP1: Oscillating Position 1	Left reversal point (oscillation)	m	x	x	x	x	x	x			Real
OSP2: Oscillating Position 2	Right reversal point (oscillation)	m	x	x	x	x	x	x			Real

12.2 List of addresses

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Address identifier	Address type	Modal/ non-modal (m/s)	G70/ G71	G700/ G7100	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Data type
OSE: Oscillating end position	Oscillation end position	m	x	x	x	x	x	x			Real
OSNSC: Oscillating: number spark out cycles	Oscillating: number of spark out cycles	m									Integer without sign
OSCTRL: Oscillating control	Oscillation control options	m									Integer without sign: set options, integer without sign: reset options
OSCILL: Oscillating	Axis assignment for oscillation, activate oscillation	m									Axis: 1 - 3 infeed axes
FDA: Feed DRF axial	Axis feedrate for handwheel override	s		x							Real without sign
FGREF	Reference radius	m	x	x							Real without sign
POLF	LIFTFAST position	m	x	x							Real without sign
FXS: Fixed stop	Activate travel to fixed stop	m									Integer without sign
FXST: Fixed stop torque	Torque limit for travel to fixed stop	m									Real
FXSW: Fixed stop window	Monitoring window for travel to fixed stop	m									Real

In these addresses, an axis or an expression of axis type is specified in square brackets. The data type in the above column shows the type of value assigned.
*) Absolute end points: modal, incremental end points: non-modal, otherwise modal/non-modal depending on syntax of G function.

1) The vocabulary word is **not** valid for SINUMERIK FM-NC.

2) The vocabulary word is **not** valid for SINUMERIK FM-NC/810D.

840 D
NCU 571

840 D
NCU 572
NCU 573

810D

840Di

12.2.4 Settable addresses

Address identifier	**Address type**	**Modal/ non-modal (m/s)**	**G70/ G71**	**G700/ G710**	**G90/ G91**	**IC**	**AC**	**DC, ACN, ACP**	**CIC, CAC, CDC, CACN, CACP**	**Qu**	**Max. no.**	**Data type**
Axis values and end points												
X, Y, Z, A, B, C	Axis	*)	x	x	x	x	x	x			8	Real
AP: Angle polar	Polar angle	m/s*			x	x	x				1	Real
RP: Polar radius	Polar radius	m/s*	x	x	x	x	x				1	Real without sign
Tool orientation												
A2, B2, C2 [1)]	Euler angles	s									3	Real
A3, B3, C3 [1)]	Direction vector component	s									3	Real
A4, B4, C4 for block beginning [1)]	Normal vector component	m									3	Real
A5, B5, C5 for end of block [1)]	Normal vector component	s									3	Real
LEAD: Lead angle [1)]	Lead angle	m									1	Real
TILT: Tilt angle [1)]	Tilt angle	m									1	Real
ORIS:[1)] Orientation Smoothing Factor	Orientation change (referring to the path)	m									1	Real
Interpolation parameters												
I, J, K** I1, J1, K1	Interpolation parameter Intermediate point coordinate	s s	x x	x x	 x	x** x	x** x				3 3	Real Real
RPL: Rotation plane	Rotation in the plane	s									1	Real
CR: Circle – Radius	Circle radius	s	x	x							1	Real without sign
AR: Angle circular	Arc angle										1	Real without sign
TURN	Number of turns for helix	s									1	Integer without sign
PL: Parameter - Interval - Length	Parameter interval length	s									1	Real without sign
PW: Point – Weight	Point weight	s									1	Real without sign
SD: Spline – Degree	Spline degree	s									1	Integer without sign
TU: Turn	Turn	m										Integer without sign
STAT: State	State	m										Integer without sign
SF: Spindle offset	Starting point offset for thread cutting	m									1	Real

12.2 List of addresses

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Address identifier	Address type	Modal/ non-modal (m/s)	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Max. no.	Data type
DISR: Distance for repositioning	Distance for repositioning	s	x	x							1	Real without sign
DISPR: Distance path for repositioning	Distance for repositioning	s	x	x							1	Real without sign
ALF: Angle lift fast	Angle for lift fast	m									1	Integer without sign
DILF: Distance lift fast	Distance for lift fast	m	x	x							1	Real
FP	Fixed point: Number of fixed point to be approached	s									1	Integer without sign
RNDM: Round modal	Modal rounding	m	x	x							1	Real without sign
RND: Round	Non-modal rounding	s	x	x							1	Real without sign
CHF: Chamfer	Chamfer non-modal	s	x	x							1	Real without sign
CHR: Chamfer	Chamfer in the original direction of movement	s	x	x							1	Real without sign
ANG: Angle	Contour angle	s									1	Real
ISD: Insertion depth	Insertion depth	m	x	x							1	Real
DISC: Distance	Transition circle overshoot in tool radius compensation	m	x	x							1	Real without sign
OFFN	Offset contour - normal	m	x	x							1	Real
DITS	Thread run-in path	m	x	x							1	Real
DITE	Thread run-out path	m	x	x							1	Real
Nibbling/punching												
SPN: Stroke/Punch Number [2)]	Number of path sections per block	s									1	INT
SPP: Stroke /Punch Path [2)]	Length of a path section	m									1	Real
Grinding												
ST: Sparking out time	Sparking-out time	s									1	Real without sign
SR: Sparking out retract path	Retraction path	s	x	x							1	Real without sign
Approximate positioning criteria												
ADIS	Approximate distance	m	x	x							1	Real without sign

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Address identifier	Address type	Modal/ non-modal (m/s)	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Max. no.	Data type
ADISPOS	Approximate distance for rapid traverse	m	x	x							1	Real without sign
Measurement												
MEAS: Measure	Measure with touch-trigger probe	s									1	Integer without sign
MEAW: Measure without deleting distance-to-go	Measure with touch-trigger probe without deleting distance-to-go	s									1	Integer without sign
Axis, spindle behavior												
LIMS: Limit spindle speed	Spindle speed limitation	m									1	Real without sign
Feeds												
FAD	Speed of the slow infeed motion	s		x							1	Real without sign
FD: Feed DRF	Path feed for handwheel override	s		x							1	Real without sign
FORI1	Feed for rotating orientation vector on large circle	m									1	Real without sign
FORI2	Feed for rotation superimposed on rotated orientation vector	m									1	Real without sign
FRC	Feed for radius and chamfer	s		x								Real without sign
FRCM	Feed for radius and chamfer, modal	m		x								Real without sign
OEM addresses												
OMA1: OEM – Address 1 [2)]	OEM – address 1	m				x	x	x			1	Real
OMA2: OEM – Address 2 [2)]	OEM – address 2	m				x	x	x			1	Real
OMA3: OEM – Address 3 [2)]	OEM – address 3	m				x	x	x			1	Real
OMA4: OEM – Address 4 [2)]	OEM – address 4	m				x	x	x			1	Real

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Address identifier	Address type	Modal/ non-modal (m/s)	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	Qu	Max. no.	Data type
OMA5: OEM – Address 5 [2)]	OEM – address 5	m				x	x	x			1	Real

*) Absolute end points: modal, incremental end points: non-modal, otherwise modal/non-modal depending on syntax of G function.

**) As circle center points, IPO parameters act incrementally. They can be programmed in absolute mode with AC. The address modification is ignored when the parameters have other meanings (e.g. thread pitch).

1) The vocabulary word is **not** valid for SINUMERIK FM-NC/810 D

2) The vocabulary word is not valid for SINUMERIK FM-NC/810D/NCU571.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12.3 List of G functions/preparatory functions

Legend for describing the G groups

No.: internal number, for e.g. PLC interface
X: No. for GCODE_RESET_VALUES not permitted

m: Modal
see: Non-modal

Def.: Default setting Siemens AG (SAG)
F : Milling, D : Turning or other settings
MH.: Default setting, see data supplied by machine manufacturer

Group 1: Modally active motion commands						
Name	**No.**	**Definition**	**X**	**m/s**	**SAG**	**MH**
G0	1.	Rapid traverse		m		
G1	2.	Linear interpolation (linear interpolation)		m	Default	
G2	3.	Circular interpolation clockwise		m		
G3	4.	Circular interpolation counterclockwise		m		
CIP	5.	Circle through points: Circular interpolation through intermediate point		m		
ASPLINE	6.	Akima spline		m		
BSPLINE	7.	B spline		m		
CSPLINE	8.	Cubic spline		m		
POLY #	9.	Polynomial: Polynomial interpolation		m		
G33	10.	Thread cutting with constant lead		m		
G331	11.	Tapping		m		
G332	12.	Retraction (tapping)		m		
OEMIPO1 ##	13.	OEM interpolation 1 *)		m		
OEMIPO2 ##	14.	OEM interpolation 2 *)		m		
CT	15.	Circle with tangential transition		m		
G34	16.	Increase in thread pitch (progressive change)		m		
G35	17.	Decrease in thread pitch (degressive change)		m		
INVCW	18.	Involute interpolation in CW direction		m		
INVCCW	19.	Involute interpolation in CCW direction		m		

*) The OEM can incorporate two additional interpolation modes. The names can be changed by the OEM.
If no function from the group is programmed with modal G functions, the default setting (which can be changed in the machine data) applies: $MC_GCODE_RESET_VALUES
The vocabulary word is **not** valid for SINUMERIK FM-NC.
The vocabulary word is **not** valid for SINUMERIK FM-NC/810D/NCU571.

Group 2: Non-modally active movements, dwell time						
G4	1.	Predefined dwell time	X	s		
G63	2.	Tapping without synchronization	X	s		
G74	3.	Reference point approach with synchronization	X	s		
G75	4.	Fixed point approach	X	s		
REPOSL	5.	Repositioning linear: Linear repositioning	X	s		
REPOSQ	6.	Repositioning quadrant: Repositioning in a quadrant	X	s		
REPOSH	7.	Repositioning semicircle: Repositioning in semicircle	X	s		
REPOSA	8.	Repositioning linear all axis: Linear repositioning with all axes	X	s		
REPOSQA	9.	Repositioning quadrant all axes: Linear repositioning with all axes, geometry axes in quadrant	X	s		

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Group 2: Non-modally active movements, dwell time

REPOSHA	10.	Repositioning semicircle all axes: Repositioning with all axes; geometry axes in semicircle	X	s		
G147	11.	Soft approach with straight line	X	s		
G247	12.	Soft approach with quadrant	X	s		
G347	13.	Soft approach with semicircle	X	s		
G148	14.	Soft retraction with straight line	X	s		
G248	15.	Soft retraction with quadrant	X	s		
G348	16.	Soft retraction with semicircle	X	s		
G05	17.	Oblique plunge-cut grinding	X	s		
G07	18.	Compensatory motion during oblique plunge-cut grinding	X	s		

Group 3: Programmable frame, working area limitation and pole programming

Name	No.	Definition	X	m/s	SAG	MH
TRANS	1.	TRANSLATION: programmable translation	X	s		
ROT	2.	ROTATION: programmable rotation	X	s		
SCALE	3.	SCALE: programmable scale	X	s		
MIRROR	4.	MIRROR: Programmable mirror	X	s		
ATRANS	5.	Additive TRANSLATION: programmable additive translation	X	s		
AROT	6.	Additive ROTATION: programmable rotation	X	s		
ASCALE	7.	Additive SCALE: programmable scale	X	s		
AMIRROR	8.	Additive MIRROR: Programmable mirror	X	s		
	9.	Not assigned				
G25	10.	Minimum working area limitation/spindle speed limit	X	s		
G26	11.	Maximum working area limitation/spindle	X	s		
G110	12.	Polar programming relative to last programmed set position	X	s		
G111	13.	Polar programming relative to origin of current workpiece coordinate system	X	s		
G112	14.	Polar programming relative to last valid pole	X	s		
G58	15.	Programmable offset, absolute axial substitution	X	s		
G59	16.	Programmable offset, additive axial substitution	X	s		
ROTS	17.	Rotation with solid angles	X	s		
AROTS	18.	Additive rotation with solid angles	X	s		

Group 4: FIFO

STARTFIFO	1.	Start FIFO Execute and simultaneously fill preprocessing memory		m	Default	
STOPFIFO	2.	STOP FIFO Stop machining; fill preprocessing memory until STARTFIFO is detected, FIFO full or end of program		m		
FIFOCTRL	3.	FIFO CTRL, Proprocessing memory control		m		

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Group 6: Plane selection						
Name	**No.**	**Definition**	**X**	**m/s**	**SAG**	**MH**
G17	1.	Plane selection 1st - 2nd geometry axis		m	Default	
G18	2.	Plane selection 3rd - 1st geometry axis		m		
G19	3.	Plane selection 2nd - 3rd geometry axis		m		

Group 7: Tool radius compensation						
G40	1.	No tool radius compensation		m	Default	
G41	2.	Tool radius compensation to left of contour	X	m		
G42	3.	Tool radius compensation to right of contour	X	m		

Group 8: Settable zero offset						
G500	1.	Deactivate all settable G54-G57 frames if G500 does not contain a value		m	Default	
G54	2.	1st settable zero offset		m		
G55	3.	2nd settable zero offset		m		
G56	4.	3rd settable zero offset		m		
G57	5.	4th settable zero offset		m		
G505	6.	5th settable zero offset		m		
G5xx	n+1	nth settable zero offset ng		m		
G599	100.	99th settable zero offset		m		

The G functions of this group activate a settable user – frame $P_UIFR[ ].
G54 corresponds to frame SP_UIFR[1], G505 corresponds to frame SP_UIFR[5].
The number of settable user frames and therefore the number of G functions in this group can be configured in the machine data SMC_MM_NUM_USER_FRAMES.

Group 9: Frame suppression						
G53	1.	Suppression of current frames: Programmable frame including system frame for TOROT and TOFRAME and active settable frame G54 ... G599	X	s		
SUPA	2.	Suppression as for G153 and including system frames for actual-value setting, scratching, ext. zero offset, PAROT, including handwheel offsets (DRF), [external zero offset], overlaid motion	X	s		
G153	3.	Suppression as for G53 and including all channel-specific and / or NCU-global basic frame	X	s		

Group 10: Exact stop – continuous-path mode						
G60	1.	Deceleration, exact stop		m	Default	
G64	2.	Continuous-path mode		m		
G641	3.	Continuous-path mode (G64) with programmable approximate distance		m		
G642	4.	Corner rounding with axial precision		m		
G643	5.	Block-internal axial corner rounding		m		
G644	6.	Corner rounding with specified axis dynamics		m		

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Group 11: Non-modal exact stop						
Name	**No.**	**Definition**	**X**	**m/s**	**SAG**	**MH**
G9	1.	Deceleration, exact stop	X	s		

Group 12: Block change criterion for exact stop (G60/G09)						
G601	1.	Block change on exact stop fine		m	Default	
G602	2.	Block change on exact stop coarse		m		
G603	3.	Block change on IPO – end of block		m		

Group 13: Workpiece dimensions in imperial/metric						
G70	1.	Input system inches (lengths)		m		
G71	2.	Input system metric (lengths)		m	Default	
G700	3.	Input system in inches; inch/min (lengths + velocity + system variable)		m		
G710	4.	Input system, metric; mm; mm/min (lengths + velocity + system variable)		m		

Group 14: Workpiece dimensions, absolute/incremental						
G90	1.	Absolute dimensions		m	Default	
G91	2.	Incremental dimensions		m		

Group 15: Feed type						
G93	1.	Inverse-time feedrate rpm		m		
G94	2.	Linear feedrate in mm/min, inches/min		m	Default	
G95	3.	Revolutional feedrate in mm/rev, inches/rev		m		
G96	4.	Constant cutting speed (type of feed as for G95) ON		m		
G97	5.	Constant cutting speed (type of feed as for G95) OFF		m		
G931	6.	Feedrate specification by travel time, deactivate const. path velocity		m		
G961	7.	Constant cutting speed (type of feed as for G94) ON		m		
G971	8.	Constant cutting speed (type of feed as for G94) OFF		m		
G942	9.	Freeze linear feedrate and constant cutting rate or spindle speed		m		
G952	10.	Freeze revolutional feedrate and const. cutting rate or spindle speed		m		
G962	11.	Linear or revolutional feedrate and constant cutting rate		m		
G972	12.	Freeze linear or revolutional feedrate and constant spindle speed		m		

Group 16: Feedrate override on inside and outside curvature						
CFC	1.	Constant feed at contour		m	Default	
CFTCP	2.	Constant feed in tool center point		m		
CFIN	3.	Constant feed at internal radius, acceleration at external radius		m		

840 D
NCU 571

840 D
NCU 572
NCU 573

810D

840Di

Group 17: Approach and retraction response, tool offset						
Name	**No.**	**Definition**	**X**	**m/s**	**SAG**	**MH**
NORM	1.	Normal position at start and end points		m	Default	
KONT	2.	Travel around contour at start and end points		m		

Group 18: Corner behavior, tool offset						
G450	1.	Transition circle (tool travels round workpiece corners on a circular path)		m	Default	
G451	2.	Intersection of equidistant paths (tool backs off from the workpiece corner)		m		

Group 19: Curve transition at beginning of spline						
BNAT	1.	Begin natural: natural transition to first spline block		m	Default	
BTAN	2.	Begin tangential: tangential transition to first spline block		m		
BAUTO	3.	Begin not a knot: (no node) Start is determined by the position of the 1st point		m		

Group 20: Curve transition at end of spline						
ENAT	1.	End natural: natural transition to next traversing block		m	Default	
ETAN	2.	End tangential: tangential transition to next traversing block at spline begin		m		
EAUTO	3.	End not a knot: (no node) End is determined by the position of the last point		m		

Group 21: Acceleration profile						
BRISK	1.	Brisk path acceleration		m	Default	
SOFT	2.	Soft path acceleration		m		
DRIVE	3.	Velocity-dependent path acceleration		m		

12.3 List of G functions/preparatory functions

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 22: Tool offset types						
CUT2D	1.	Cutter – compensation – type 2-dimensional 2 1/2D tool offset determined by G17 – G19		m	Default	
CUT2DF	2.	Cutter – compensation – type 2-dimensional frame – relative: 2 1/2D tool offset determined by frame The tool offset is effective in relation to the current frame (inclined plane)		m		
CUT3DC #	3.	Cutter – compensation – type 3-dimensional circumference: 3D tool compensation circumference milling		m		
CUT3DF #	4.	Cutter – compensation – type 3-dimensional face: 3D tool offset with inconstant tool orientation		m		
CUT3DFS #	5.	Cutter – compensation – type 3-dimensional face: 3D tool offset face milling with constant tool orientation independent of active frame		m		
CUT3DFF #	6.	Cutter – compensation – type 3-dimensional face frame: 3D tool offset face milling with constant tool orientation dependent on active frame		m		
CUT3DCC #	7.	Cutter – compensation – type 3-dimensional circumference: with limit surfaces		m		
CUT3DCCD #	8.	Cutter – compensation – type 3-dimensional circumference: with limit surfaces with differential tool		m		

Group 23: Collision monitoring on inside contours						
Name	**No.**	**Definition**	**X**	**m/s**	**SAG**	**MH**
CDOF	1.	Collision detection OFF: Collision monitoring OFF		m	Default	
CDON	2.	Collision detection ON: Collision monitoring ON		m		
CDOF2	3.	Collision detection OFF: (currently for CUT3DC only)		m		

Group 24: Feedforward control						
FFWOF	1.	Feedforward control OFF		m	Default	
FFWON	2.	Feedforward control ON		m		

Group 25: Tool orientation reference						
ORIWKS #	1.	Tool – orientation in workpiece coordinate system Tool orientation in workpiece coordinate system (WCS)		m	Default	
ORIMKS #	2.	Tool – orientation in machine coordinate system Tool orientation in machine coordinate system (MCS)		m		

The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 26: Repositioning point for REPOS						
RMB	1.	REPOS mode beginning of block: Reapproach to start of block position		m		
RMI	2.	REPOS – Mode interrupt: Reapproach to interruption point		m	Default	
RME	3.	REPOS – Mode end of block: Reapproach to end of block position		m		
RMN	4.	Repos – Mode end of nearest orbital block: Reapproach to nearest path point		m		

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Group 27: Tool offset for change in orientation at outside corners						
ORIC #	1.	Orientation change continuously: Orientation changes at outside corners are superimposed on the circle block to be inserted		m	Default	
ORID #	2.	Orientation change discontinuously: Orientation changes are performed before the circle block		m		

The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 28: Working area limitation ON/OFF						
WALIMON	1.	Working area limitation ON		m	Default	
WALIMOF	2.	Working area limitation OFF		m		

Group 29: Radius – diameter						
DIAMOF	1.	Diametral Diameter programming OFF; Radius programming for G90/G91		m	Default	
DIAMON	2.	Diametral Diameter programming ON for G90/G91		m		
DIAM90	3.	Diametral programming G90: Diameter programming for G90; Radius programming for G91		m		
DIAMCYCOF	4.	Diametral Radius programming for G90/G91: ON. The G-code of this group that was last active remains active for display		m		

Group 30: Compressor ON/OFF						
Name	**No.**	**Definition**	**X**	**m/s**	**SAG**	**MH**
COMPOF #	1.	Compressor OFF		m	Default	
COMPON #	2.	Compressor ON		m		
COMPCURV #	3.	Compressor ON: polynomials with constant curvature		m		
COMPCAD #	4.	Compressor ON: Optimized surface quality CAD program (SW 6 and higher)		m		

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Group 31: OEM – G group						
G810 #	1.	OEM – G function			Default	
G811 #	2.	OEM – G function				
G812 #	3.	OEM – G function				
G813 #	4.	OEM – G function				
G814 #	5.	OEM – G function				
G815 #	6.	OEM – G function				
G816 #	7.	OEM – G function				
G817 #	8.	OEM – G function				
G818 #	9.	OEM – G function				
G819 #	10.	OEM – G function				

Two G groups are reserved for the OEM. This enables the OEM to program functions that can be customized.
The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 32: OEM G group						
G820 #	1.	OEM – G function			Default	
G821 #	2.	OEM – G function				
G822 #	3.	OEM – G function				
G823 #	4.	OEM – G function				
G824 #	5.	OEM – G function				
G825 #	6.	OEM – G function				
G826 #	7.	OEM – G function				
G827 #	8.	OEM – G function				
G828 #	9.	OEM – G function				
G829 #	10.	OEM – G function				

Two G groups are reserved for the OEM. This enables the OEM to program functions that can be customized.

Group 33: Settable fine tool offset						
FTOCOF #	1.	Fine tool offset compensation OFF: Online fine tool offset OFF		m	Default	
FTOCON #	2.	Fine tool offset compensation on: Online fine tool offset ON	X	m		

Group 34: Tool orientation smoothing						
Name	**No.**	**Definition**	**X**	**m/s**	**SAG**	**MH**
OSOF #	1.	Tool orientation smoothing OFF		m	Default	
OSC #	2.	Continuous tool orientation smoothing		m		
OSS #	3.	Tool orientation smoothing at end of block		m		
OSSE #	4.	Tool orientation smoothing at start and end of block		m		

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Group 35: Punching and nibbling						
SPOF #	1.	Stroke/punch OFF: Stroke OFF, nibbling, punching OFF		m	Default	
SON #	2.	Stroke ON: Nibbling ON		m		
PON #	3.	Punch ON: Punching ON		m		
SONS #	4.	Stroke ON slow: Nibbling ON in IPO cycle	X	m		
PONS #	5.	Punch ON slow: Punching ON in IPO cycle	X	m		

Group 36: Delayed punching						
PDELAYON #	1.	Punch with delay ON		m	Default	
PDELAYOF #	2.	Punch with delay OFF		m		

The vocabulary word is **not** valid for SINUMERIK 810D/NCU571.

Group 37: Feed profile:						
FNORM #	1.	Feed normal: Feed normal (to DIN 66025)		m	Default	
FLIN #	2.	Feed linear: Feed linear variable		m		
FCUB #	3.	Feed cubic: Feedrate variable according to cubic spline		m		

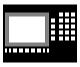
Group 38: Assignment of high-speed inputs/outputs for punching/nibbling						
SPIF1 #	1.	Stroke/punch interface 1: fast NCK inputs/outputs for punching/nibbling byte 1		m	Default	
SPIF2 #	2.	Stroke/punch interface 2: fast NCK inputs/outputs for punching/nibbling byte 2		m		

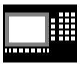
Group 39: Programmable contour accuracy						
CPRECOF	1.	Contour precision OFF: Programmable contour precision OFF		m	Default	
CPRECON	2.	Contour precision ON: Programmable contour precision ON		m		

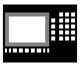
The vocabulary word is **not** valid for SINUMERIK NCU571.

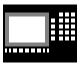
Group 40: Tool radius compensation, constant						
CUTCONOF	1.	Constant radius compensation OFF		m	Default	
CUTCONON	2.	Constant radius compensation ON		m		

Group 41: Thread cutting interruption						
Name	**No.**	**Definition**	**X**	**m/s**	**SAG**	**MH**
LFOF	1.	Interrupt thread cutting OFF		m	Default	
LFON	2.	Interrupt thread cutting ON		m		


840 D
NCU 571


840 D
NCU 572
NCU 573


810D


840Di

Group 42: Toolholder						
TCOABS	1.	Toolholder orientation absolute		m	Default	
TCOFR	2.	Toolholder orientation frame alignment of tool on Z axis		m		
TCOFRZ	3.	Orientable toolholder frame-related (tool on Z axis)		m		
TCOFRY	4.	Orientable toolholder frame-related (tool on Y axis)		m		
TCOFRX	5.	Orientable toolholder frame-related (tool on X axis)		m		

Group 43: Approach direction SAR						
G140	1.	SAR approach direction defined by G41/G42		m	Default	
G141	2.	SAR approach direction to left of contour		m		
G142	3.	SAR approach direction to right of contour		m		
G143	4.	SAR approach direction tangent-dependent		m		

Group 44: Path segmentation SAR						
G340	1.	Spatial approach block (depth and in plane (helix))		m	Default	
G341	2.	Initial infeed on perpendicular axis (z), then approach in plane		m		

Group 45: Path reference for FGROUP axes						
SPATH	1.	Path reference for FGROUP axes is arc length		m	Default	
UPATH	2.	Path reference for FGROUP axes is curve parameter		m		

Group 46: Plane definition for rapid lift						
LFTXT	1.	Tangential tool direction on retraction		m	Default	
LFWP	2.	Non-tangential tool direction on retraction		m		
LFPOS	3.	Axial retraction to a position		m		

Group 47: Mode switchover for external NC code						
G290	1.	Switchover to SINUMERIK mode (activate SINUMERIK language mode)		m	Default	
G291	2.	Switchover to ISO mode (activate ISO language mode)		m		

Group 48: TRC approach/retraction behavior						
G460	1.	Collision monitoring for approach and retraction block on		m	Default	
G461	2.	If no intersection in TRC block, extend border block with arc		m		
G462	3.	If no intersection in TRC block, extend border block with straight line		m		

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Group 49: Point-to-point movement TRC						
Name	**No.**	**Definition**	**X**	**m/s**	**SAG**	**MH**
CP	1.	**C**ontinuous **p**ath; path motion		m	Default	
PTP	2.	**p**oint **t**o **p**oint; Point-to-point motion (synchronized axis motion)		m		

Group 50: Orientation programming						
ORIEULER	1.	Orientation angle via Euler angle		m	Default	
ORIRPY	2.	Orientation angle via RPY angle		m		
ORIVIRT1	3.	Orientation angle via virtual orientation axes (definition 1)		m		
ORIVIRT2	4.	Orientation angle via virtual orientation axes (definition 2)		m		
ORIAXPOS	5.	Orientation angle via virtual orientation axes with rotary axis positions		m		

Group 51: Orientation interpolation						
ORIVECT	1.	Large-radius circular interpolation (identical to ORIPLANE)		m	Default	
ORIAXES	2.	Linear interpolation of machine axes or orientation axes		m		
ORIPATH	3.	Tool orientation trajectory referred to path		m		
ORIPLANE	4.	Interpolation in plane (identical to ORIVECT)		m		
ORICONCW	5.	Interpolation on a circular peripheral surface in CW direction		m		
ORICONCCW	6.	Interpolation on a conical peripheral surface in CCW direction		m		
ORICONIO	7.	Interpolation on a conical peripheral surface with intermediate orientation setting		m		
ORICONTO	8.	Interpolation on a conical peripheral surface in tangential transition		m		
ORICURVE	9.	Interpolation with additional space curve for orientation		m		

Group 52: Workpiece-related WCS						
PAROTOF	1.	Deactivate workpiece-related frame rotation		m	Default	
PAROT	2.	Align workpiece coordinate system (WCS) on workpiece		m		

Group 53: Frame rotations in tool direction						
TOROTOF	1.	Frame rotation in tool direction OFF		m	Default	
TOROT	2.	Frame rotation ON Z axis parallel to tool orientation		m		
TOROTZ	3.	Frame rotation ON Z axis parallel to tool orientation		m		
TOROTY	4.	Frame rotation ON Y axis parallel to tool orientation		m		
TOROTX	5.	Frame rotation ON X axis parallel to tool orientation		m		
TOFRAME	6.	Frame rotation in tool direction Z axis parallel to tool orientation		m		
TOFRAMEZ	7.	Frame rotation in tool direction Z axis parallel to tool orientation		m		
TOFRAMEY	8.	Frame rotation in tool direction Y axis parallel to tool orientation		m		
TOFRAMEX	9.	Frame rotation in tool direction X axis parallel to tool orientation		m		

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

Group 54: Interpolation of rotational vector						
Name	**No.**	**Definition**	**X**	**m/s**	**SAG**	**MH**
ORIROTA	1.	Orientation rotation absolute: Angle of rotation in relation to **absolute** direction of rotation		m	Default	
ORIROTR	2.	Orientation rotation relative: Angle of rotation **relative** to plane between start and end orientations		m		
ORIROTT	3.	Orientation rotation tangential: Angle of rotation **relative** to change in orientation vector		m		

Group 55: Rapid traverse with/without linear interpolation						
RTLION	1.	Rapid traverse (G0) with linear interpolation On: G0 with linear interpolation		m	Default	
RTLIOF	2.	Rapid traverse (G0) with linear interpolation Off: G0 without linear interpolation (single-axis interpolation)		m		

Group 56: Inclusion of tool wear						
TOWSTD	1.	Tool wear default initial setting value for offsets in tool length		m	Default	
TOWMCS	2.	Tool WearCoard MCS:Wear values in machine coordinate system (MCS)		m		
TOWWCS	3.	Tool WearCoard WCS:Wear values in workpiece coordinate system (WCS)		m		
TOWBCS	4.	Tool WearCoard BCS:Wear values in basic coordinate system (MCS)		m		
TOWTCS	5.	Tool WearCoard TCS:Wear values in the tool coordinate system (tool carrier ref. point T at the tool holder)		m		
TOWKCS	6.	Wear values in the coordinate system of the tool head for kinetic transformation (differs from MCS by tool rotation)		m		

Group 57: Automatic corner override						
FENDNORM	1.	Corner deceleration deactivated		m	Default	
G62	2.	Corner deceleration at inside coners when tool radiuse offset is active		m		
G621	3.	Corner deceleration at all corners		m		

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12.4 List of predefined subprograms

Some control functions are activated with subprogram call syntax.

1. Coordinate system					
Vocabulary word/ function identifier	**Parameter 1**	**Parameter 2**	**Parameter 3–15**	**Parameter 4–16**	**Description**
PRESETON	AXIS*: Name of machine axis	REAL: Preset offset G700/G7100 context	Parameter 3–15 same as 1 ...	Parameter 4–16 same as 2 ...	Sets the actual value for programmed axes. One axis identifier is programmed at a time, with its respective value in the next parameter. Preset offsets can be programmed for up to 8 axes with PRESETON.
DRFOF					Deletes the DRF offset for all axes assigned to the channel.

*) As a general rule, geometry or special axis identifiers can also be used instead of the machine axis identifier, as long as the reference is unambiguous.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12.4.1 Predefined subprogram calls

2. Axis groups			
	Parameter 1–8	**Description**	
FGROUP	Channel axis identifier	Variable F value reference: defines the axes to which the path feed refers. Maximum number of axes: 8 The default setting for the F value reference is activated with FGROUP ( ) without parameters.	
CLGON #	REAL: Max. speed of regulating wheel	Centerless grinding ON	
CLGOF #		Centerless grinding OFF	
	Parameter 1–8	**Parameter 2–9**	**Description**
SPLINEPATH	INT: spline group (must be 1)	AXIS: Geometry or special axis identifier	Definition of the spline group Maximum number of axes: 8
BRISKA	AXIS		Switch on brisk axis acceleration for the programmed axes
SOFTA	AXIS		Switch on jerk limited axis acceleration for programmed axes
DRIVEA ###	AXIS		Switch on knee-shaped acceleration characteristic for programmed axes
JERKA	AXIS		The acceleration behavior set in machine data $MA_AX_JERK_ENABLE is active for the programmed axes.

The vocabulary word is **not** valid for SINUMERIK FM-NC/810D/NCU571.
The vocabulary word is **not** valid for SINUMERIK 810D.
The vocabulary word is **only** valid for SINUMERIK FM-NC.

3. Coupled motion						
Vocabulary word/ subprogram identifier	**Parameter 1**	**Parameter 2**	**Parameter 3**	**Para-meter 4**	**Parameter 5**	**Description**
TANG #	AXIS: Axis identifier following axis	REAL: Master axis 1	AXIS: Master axis 2	REAL: Coupling factor	CHAR: Optional: "B": follow-up in basic coordinate system "W": follow-up in workpiece coordinate system	Preparatory instruction for the definition of a tangential follow-up: The tangent for the follow-up is determined by the two master axes specified. The coupling factor specifies the relationship between a change in the angle of tangent and the following axis. It is usually 1.
TANGON #	AXIS: Axis identifier following axis	REAL: Offset angle				Tangential follow-up mode ON
TANGOF #	AXIS: Axis identifier following axis					Tangential follow-up mode OFF
TLIFT #	AXIS: Following axis		REAL: Distance			Tangential lift: Tangential follow-up mode ON
TRAILON	AXIS: Following axis	AXIS: Leading axis	REAL: Coupling factor			Trailing ON: Asynchronous coupled motion ON
TRAILOF	AXIS: Following axis	AXIS: Leading axis				Trailing OFF: Asynchronous coupled motion OFF

The vocabulary word is **not** valid for SINUMERIK FM-NC/NCU571.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

6. Revolutional feedrate			
Vocabulary word/ function identifier	**Parameter 1**	**Parameter 2**	**Description**
FPRAON	AXIS: Axis for which revolutional feedrate is activated	AXIS: Axis/spindle from which the revolutional feedrate is derived. If no axis has been programmed, the revolutional feedrate is derived from the master spindle.	Feedrate per revolution axial ON: Axial revolutional feedrate ON
FPRAOF	AXIS: Axis for which revolutional feedrate is deactivated		Feedrate per revolution axial OFF: Axial revolutional feedrate OFF The revolutional feedrate can be deactivated for several axes at once. You can program as many axes as are permitted in a block.
FPR	AXIS: Axis/spindle from which the revolutional feedrate is derived. If no axis has been programmed, the revolutional feedrate is derived from the master spindle.		Feedrate per revolution: Selection of a rotary axis or spindle from which the revolutional feedrate of the path is derived if G95 is programmed. If no axis/spindle has been programmed, the revolutional feedrate is derived from the master spindle. The setting made with FPR is modal.

It is also possible to program a spindle instead of an axis: FPR(S1) or FPR(SPI(1))

7. Transformations			
Vocabulary word/ function identifier	**Parameter 1**	**Parameter 2**	**Description**
TRACYL	REAL: working diameter	INT: number of transfor-mation	Cylinder: Peripheral surface transformation Several transformations can be set per channel. The transformation number specifies which transformation is to be activated. If the 2nd parameter is omitted, the transformation group defined in the MD is activated.
TRANSMIT	INT: number of transformation		Transmit: Polar transformation Several transformations can be set per channel. The transformation number specifies which transformation is to be activated. If the parameter is omitted, the transformation group defined in the MD is activated.
TRAANG #	REAL: angle	INT: number of transfor-mation	Transformation inclined axis: Several transformations can be set per channel. The transformation number specifies which transformation is to be activated. If the 2nd parameter is omitted, the transformation group defined in the MD is activated. If the angle is not programmed: TRAANG (.2) or TRAANG the last angle is active modally.
TRAORI #	INT: number of transformation		Transformation oriented: 4, 5-axis transformation Several transformations can be set per channel. The transformation number specifies which transformation is to be activated.
TRACON	INT: number of transformation	REAL: Further para-meters, MD-dependent	Transformation Concentrated: Cascaded transformation; the meaning of the parameters depends on the type of cascading.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

7. Transformations			
Vocabulary word/ function identifier	**Parameter 1**	**Parameter 2**	**Description**
TRAFOOF			Deactivate transformation

For each transformation type, there is one command for one transformation per channel. If there are several transformations of the same transformation type per channel, the transformation can be selected with the corresponding command and parameters. It is possible to deselect the transformation by a transformation change or an explicit deselection.

#) The vocabulary word is **not** valid for SINUMERIK FM-NC/NCU571.

8. Spindle			
Vocabulary word/ subprogram identifier	**Parameter 1**	**Parameter 2 and others**	**Description**
SPCON	INT: spindle number	INT: spindle number	Spindle position control ON: Switch to position-controlled spindle operation
SPCOF	INT: spindle number	INT: spindle number	Spindle position control OFF: Switch to speed-controlled spindle operation
SETMS	INT: spindle number		Set master spindle: declare the spindle master spindle for the current channel. SETMS( ) without parameters activates the default setting in the machine data.

9. Grinding		
Vocabulary word/ subprogram identifier	**Parameter 1**	**Description**
GWPSON	INT: spindle number	Grinding wheel peripheral speed ON: Constant grinding wheel peripheral speed ON If the spindle number is not programmed, then grinding wheel peripheral speed is selected for the spindle of the active tool.
GWPSOF	INT: spindle number	Grinding wheel peripheral speed OFF. Constant grinding wheel peripheral speed OFF. If the spindle number is not programmed, grinding wheel peripheral speed is deselected for the spindle of the active tool.
TMON	INT: spindle number	Tool monitoring ON: If no T number is programmed, monitoring is activated for the active tool.
TMOF	INT: T number	Tool monitoring OFF: If no T number is programmed, monitoring is deactivated for the active tool.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

10. Stock removal					
Vocabulary word/ subprogram identifier	**Parameter 1**	**Parameter 2**	**Parameter 3**	**Parameter 4**	**Description**
CONTPRON	REAL [, 11]: contour table	CHAR: Stock removal method "L": Longitudinal turning: external machining "P": Face turning: External machining "N": Face turning: inside machining "G": Longitudinal turning: inside machining	INT: number of relief cuts	INT: Status of calculation 0: as before 1: Calculation forwards and backwards	Contour preparation on: Activate reference point editing The contour programs and NC blocks which are called in the following steps are divided into individual movements and stored in the contour table. The number of relief cuts is returned.
CONTDCON	REAL [, 6]: contour table	INT: 0: in programmed direction			Contour decoding The blocks for a contour are stored in a named table with one table line per block and coded to save memory.
EXECUTE	INT: error status				EXECUTE: Activate program execution. This switches back to normal program execution from reference point editing mode or after setting up a protection zone.

11. Execute table		
Vocabulary word/ subprogram identifier	**Parameter 1**	**Description**
EXECTAB	REAL [11]: Element from motion table	Execute table: Execute an element from a motion table.

12. Protection zones						
Vocabulary word/ function identifier	**Parameter 1**	**Parameter 2**	**Parameter 3**	**Parameter 4**	**Parameter 5**	**Description**
CPROTDEF	INT: Number of protection zone	BOOL: TRUE: Tool-oriented protection zone	INT: 0: Parameters 4 and 5 are not evaluated 1: Parameter 4 is evaluated 2: Parameter 5 is evaluated 3: Parameters 4 and 5 are evaluated	REAL: Limit in plus direction	REAL: Limit in minus direction	Channel-specific protection area definition: Definition of a channel-specific protection zone

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12. Protection zones						
Vocabulary word/ function identifier	**Parameter 1**	**Parameter 2**	**Parameter 3**	**Parameter 4**	**Parameter 5**	**Description**
NPROTDEF	INT: Number of protection zone	BOOL: TRUE: Tool-oriented protection zone	INT: 0: Parameters 4 and 5 are not evaluated 1: Parameter 4 is evaluated 2: Parameter 5 is evaluated 3: Parameters 4 and 5 are evaluated	REAL: Limit in plus direction	REAL: Limit in minus direction	NCK-specific protection area definition: Definition of a machine-specific protection zone
CPROT	INT: Number of protection zone	INT: option 0: Protection zone off 1: Preactivate protection zone 2: Protection zone ON	REAL: Offset for protection zone on 1st channel axis (= axis on which the geometry axes are mapped)	REAL: Offset of protection zone in 2nd channel axis	REAL: Offset of protection zone in 3rd channel axis	Channel-specific protection zone ON/OFF
NPROT	INT: Number of protection zone	INT: option 0: Protection zone off 1: Preactivate protection zone 2: Protection zone on	REAL: Offset for protection zone on 1st channel axis (= axis on which the geometry axes are mapped)	REAL: Offset of protection zone in 2nd channel axis	REAL: Offset of protection zone in 3rd channel axis	Machine-specific protection zone ON/OFF
EXECUTE	VAR INT: error status	EXECUTE: Activate program execution. This switches back to normal program execution from reference point editing mode or after setting up a protection zone.				

13. Preprocessing/single block		
STOPRE		Stop processing: Stop preprocessing until all prepared blocks are executed in main run.

14. Interrupts		
Vocabulary word/ function identifier	**Parameter 1**	**Description**
ENABLE #	INT: Number of interrupt input	Activate interrupt: Activates the interrupt routine assigned to the hardware input with the specified number. An interrupt is enabled after the SETINT instruction.
DISABLE #	INT: Number of interrupt input	Deactivate interrupt: Deactivates the interrupt routine assigned to the hardware input with the specified number. Fast retraction is not executed. The assignment between the hardware input and the interrupt routine made with SETINT remains valid and can be reactivated with ENABLE.
CLRINT #	INT: Number of interrupt input	Select interrupt: Cancel the assignment of interrupt routines and attributes to an interrupt input. The interrupt routine is deactivated and no reaction occurs when the interrupt is generated.

The vocabulary word is **not** valid for SINUMERIK FM-NC/810D.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

15. Motion synchronization		
CANCEL	INT: number of synchronized action	Aborts the modal motion-synchronous action with the specified ID

16. Function definition					
	Parameter 1	**Parameter 2**	**Parameter 3**	**Parameter 4–7**	**Description**
FCTDEF	INT: Function number	REAL: Lower limit value	REAL: Upper limit value	REAL: Coefficients a0 – a3	Define polynomial. This is evaluated in SYFCT or PUTFTOCF.

#) The vocabulary word is **not** valid for SINUMERIK FM-NC.

17. Communication			
Vocabulary word/subprogram identifier	**Parameter 1**	**Parameter 2**	**Description**
MMC #	STRING: Command	CHAR: Acknowledgment mode** "N": without acknowledgment "S": synchronous acknowledgment "A": asynchronous acknowledgment	MMC command: Command ON MMC command interpreter for the configuration of windows via NC program see /AM/ IM1 Start-Up Functions for the MMC

#) The vocabulary word is **not** valid for SINUMERIK FM-NC/810D.
**) Acknowledgment mode:
Commands are acknowledged on request from the executing component (channel, NC, ...).
Without acknowledgment: Program execution is continued when the command has been transmitted. The sender is not informed if the command cannot be executed successfully.

18. Program coordination							
Vocabulary word/sub-program identifier	**Parameter 1**	**Parameter 2**	**Parameter 3**	**Parameter 4**	**Para-meter 5**	**Para-meter 6–8**	**Description**
INIT #	INT: Channel number	STRING: Path	CHAR: Acknowledgement mode**				Selection of a module for execution in a channel.
START #	INT: Channel number	INT: Channel number					Starts selected programs simultaneously on multiple channels from running program. The command has no effect on the existing channel. 1 : 1st channel; 2 : 2nd channel.
WAITE #	INT: Channel number	INT: Channel number					Wait for end of program: Wait for end of program on another channel
WAITM #	INT: Marker number 0–9	INT: Channel number	INT: Channel number	INT: Channel number			Wait: Wait for a marker to be reached in other channels. The program waits until the WAITM with the relevant marker has been reached in the other channel. The number of the own channel can also be specified.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

18. Program coordination							
Vocabulary word/sub-program identifier	**Parameter 1**	**Parameter 2**	**Parameter 3**	**Parameter 4**	**Para-meter 5**	**Para-meter 6–8**	**Description**
WAITP	AXIS: Axis identifier	AXIS: Axis identifier	AXIS: Axis identifier	AXIS: Axis identifier	AXIS: Axis identifier	AXIS: Axis identifier	Wait for positioning axis: Wait for positioning axis to reach their programmed end point end point.
WAITS	INT: Spindle number	INT: Spindle number	INT: Spindle number	INT: Spindle number	INT: Spindle number		Wait for positioning spindle: Wait for programmed spindles previously programmed with SPOSA to reach their programmed end point.
RET							End of subprogram with no function output to the PLC.
GET #	AXIS	AXIS	AXIS	AXIS	AXIS	AXIS	Assign machine axis
GETD#	AXIS	AXIS	AXIS	AXIS	AXIS	AXIS	Assign machine axis directly
RELEASE #	AXIS	AXIS	AXIS	AXIS	AXIS	AXIS	Release machine axis
PUTFTOC #	REAL: Offset value	INT: Parameter number	INT: Channel number	INT: Spindle number			Put fine tool correction: fine tool compensation
PUTFTOCF #	INT: No. of function The number used here must be specified in FCTDEF.	VAR REAL: Reference value *)	INT: Parameter number	INT: Channel number	INT: Spindle number		Put fine tool correction function dependent: Change online tool compensation according to a function defined with FCTDEF (max. 3rd degree polynomial).

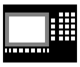
The SPI function can also be used to program a spindle instead of an axis: GET(SPI(1))

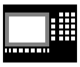
#) The vocabulary word is not valid for SINUMERIK FM-NC/NCU571.

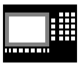
**) Acknowledgment mode:
Commands are acknowledged on request from the executing component (channel, NC, ...).
Without acknowledgment: Program execution is continued when the command has been transmitted. The executing component is not informed if the command cannot be executed successfully. Acknowledgment mode "N" or "n".
Synchronous acknowledgment: Program execution is interrupted until the receiving component has acknowledged the command. The next command positive acknowledgment.
An error is output with negative acknowledgment.
Acknowledgment mode "S", "s" or omit.
The acknowledgment behavior is defined for some commands and programmable for others.
The acknowledgment behavior is always synchronous for program coordination commands.
If the acknowledgment mode is omitted, synchronous acknowledgment is taken as the default.

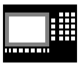
19. Data access		
	Parameter 1	**Description**
CHANDATA	INT: Channel number	Set channel number for channel data access (only permitted in initialization block); the subsequent accesses refer to the channel set with CHANDATA.

20. Messages			
	Parameter 1	**Parameter 2**	**Description**
MSG	STRING: Message		Message modal: the message is active until the next message is queued


840 D
NCU 571


840 D
NCU 572
NCU 573


810D


840Di

22. Alarms			
	Parameter 1	**Parameter 2**	**Description**
SETAL	INT: Alarm number (cycle alarms)		Set alarm set alarm

23. Compensation			
Vocabulary word/sub-program identifier	**Parameter 1–4**		**Description**
QECLRNON #	AXIS: Axis number		Quadrant error compensation learning ON
QECLRNOF #			Quadrant error compensation learning OFF

#) The vocabulary word is **not** valid for SINUMERIK FM-NC.

24. Tool management					
	Parameter 1	**Parameter 2**	**Parameter 3**		**Description**
DELT #	STRING [32]: Tool name	INT: Duplo number			Delete tool. Duplo number can be omitted.
GETSELT #	VAR INT: T number (return value)	INT: Spindle number			Get selected T number. If no spindle number is specified, the command for the master spindle applies.
SETPIECE #	INT: Workpiece count	INT: Spindle number			Set number of workpieces for all tools assigned to the spindle. If the spindle number is omitted, the command applies to the master spindle.
SETDNO	INT: Tool no. T	INT: Tool edge no.	INT: D no.		Set D no. of tool (T) and its tool edge to new
DZERO					Set D numbers of all tools of the TO unit assigned to the channel to invalid
DELDL	INT: Tool no. T	INT: D no.			Delete all additive offsets of the tool edge (or of a tool if D is not specified)
SETMTH	INT: Toolholder no.				Set toolholder no.
POSM	INT: Location no. for positioning	INT: No. of the magazine to be moved	INT: Location number of the internal magazine	INT: Magazine number of the internal magazine	Position magazine
SETTIA	VAR INT: Status=result of operation (return value)	INT: Magazine no.	INT: Wear group no.		Deactivate tool from wear group
SETTA	VAR INT: Status=result of operation (return value)	INT: Magazine no.	INT: Wear group no.		Activate tool from wear group

840 D
NCU 571

840 D
NCU 572
NCU 573

810D

840Di

24. Tool management					
	Parameter 1	**Parameter 2**	**Parameter 3**		**Description**
RESETMON	VAR INT: Status=result of operation (return value)	INT: Internal T no.	INT: D no. of tool		Set actual value of tool to setpoint

#) The vocabulary word is **not** valid for SINUMERIK FM-NC.

25. Synchronous spindle							
	Para-meter 1	**Para-meter 2**	**Parameter 3**	**Parameter 4**	**Parameter 5**	**Parameter 6**	**Description**
COUPDEF #	AXIS: Following axis	AXIS: Leading axis	REAL: Enumera-tor trans-formation ratio (FA)	REAL: Denomina-tor trans-formation ratio (LA)	STRING[8]: Block change behavior: "NOC": no block change control, block change is enabled immediately, "FINE": block change on "synchronism", "COARSE": block change on synchronism coarse and "IPOSTOP": block change in setpoint-dependent termination of overlaid movement. If the block change behavior is not specified, no change takes place in the defined behavior	STRING[2]: "DV": Setpoint linkage "AV": Actual-value linkage	Couple definition: Definition of synchronous spindle group
COUPDEL #	AXIS: Following axis	AXIS: Leading axis					Couple delete: Delete synchronous spindle group
COUPRES #	AXIS: Following axis	AXIS: Leading axis					Couple reset: Reset synchronous spindle group. The program-med values are invalidated. The machine data values are valid.

For synchronous spindles, the axis parameters are programmed with SPI(1) or S1.

26. Structure instructions in the STEP editor (editor-based program support)					
	Parameter 1	**Parameter 2**	**Parameter 3**		**Description**
SEFORM	STRING[128]: section name	INT: plane	STRING[128]: icon		Current section name for STEP editor

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

#) The vocabulary word is **not** valid for SINUMERIK 810 D.

Vocabulary word/sub-program identifier	**Parameter 1**	**Parameter 2**	**Parameter 3**	**Parameter 4**	**Description**
COUPON #	AXIS: Following axis	AXIS: Leading axis	REAL: Activation position of following axis		Couple on: Activate ELG group/synchronous spindle pair. If no activation positions are specified, the couple is activated as quickly as possible (ramp). If an activation position is specified for the following axis or spindle, this refers absolutely or incrementally to the leading axis/spindle.Only if the 3rd parameter is specified do parameters 4 and 5 have to be programmed.
COUPOF #	AXIS: Following axis	AXIS: Leading axis	REAL: Deactivation position of following axis (absolute)	REAL: Deactivation position of following axis (absolute)	Couple off: Deactivate ELG group/synchronous spindle pair. The couple parameters are retained. If positions are specified, the couple is only canceled when all the specified positions have been overtraveled. The following spindle continues to revolve at the last speed programmed before deactivation of the couple.
WAITC #	AXIS: Axis/spindle	STRING[8]: Block change criterion	AXIS: Axis/spindle	STRING[8]: Block change criterion	Wait for couple condition: Wait until couple block change conditions have been met for the axes/spindles. Up to 2 axes/spindles can be programmed. Block change condition: "NOC": no block change control, block change is enabled immediately, "FINE": block change on "synchronism fine". "COARSE": block change on "synchronism coarse" and "IPOSTOP": block change in setpoint-dependent termination of overlaid movement. If the block change behavior is not specified, the set behavior is applicable and there is no change.
AXCTSWE	AXIS: Axis/spindle				Advance container axis

#)The vocabulary word is **not** valid for SINUMERIK 810D.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12.4.2 Predefined subprogram calls in motion-synchronous actions

The following predefined subprograms appear exclusively in motion-synchronous actions.

27. Synchronous procedures				
Vocabulary word/ function identifier	**Parameter 1**	**Parameter 2**	**Parameter 3 to 5**	**Description**
STOPREOF				Stop preparation OFF: A synchronized action with a STOPREOF command causes a preprocessing stop after the next output block (= block for the main run). The preprocessing stop is canceled with the end of the output block or when the STOPREOF condition is fulfilled. All synchronized action instructions with the STOPREOF command are therefore interpreted as having been executed.
RDISABLE				Read-in disable
DELDTG	AXIS: Axis for axial delete distance-to-go (optional). If the axis is omitted, delete distance-to-go is triggered for the path distance			Delete distance-to-go A synchronized action with the DELDTG command causes a preprocessing stop after the next output block (= block for the main run). The preprocessing stop is canceled with the end of the output block or when the first DELDTG condition is fulfilled. The axial distance to the destination point on an axial delete distance-to-go is stored in $AA_DELT[<axis>]; the distance-to-go is stored in $AC_DELT.
SYNFCT	INT: Number of polynomial function defined with FCTDEF.	VAR REAL: Result variable *)	VAR REAL: Input variable **)	If the condition in the motion synchronous action is fulfilled, the polynomial determined by the first expression is evaluated at the input variable. The upper and lower range of the value is limited and the input variable is assigned.
FTOC	INT: Number of polynomial function defined with FCTDEF.	VAR REAL: Input variable **)	INT: Length 1, 2, 3 INT: Channel number INT: spindle number	Modify tool fine compensation according to a function defined with FCTDEF (polynomial no higher than 3rd degree). The number used here must be specified in FCTDEF.

*) Only special system variables can be used for the result variable. These are described in the Programming Guide Advanced in the section on "Write main run variable".

**) Only special system variables can be used for the result variable. These variables are described in the Programming Guide Advanced in the list of system variables.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12.4.3 Predefined functions

Predefined functions are invoked by means of a function call. Function calls return a value. They can be included as an operand in an expression.

1. Coordinate system

Vocabulary word/ function identifier	Result	Parameter 1	Parameter 2			Description
CTRANS	FRAME	AXIS	REAL: Offset	Parameter 3–15 same as 1 ...	Parameter 4–16 same as 2 ...	Translation: Zero offset for multiple axis. One axis identifier is programmed at a time, with its respective value in the next parameter. CTRANS can be used to program offset for up to 8 axes.
CROT	FRAME	AXIS	REAL: Rotation	Parameter 3/5 same as 1 ...	Parameter 4/6 same as 2 ...	Rotation: Rotation of the current coordinate system. Maximum number of parameters: 6 (one axis identifier and one value per geometry axis).
CSCALE	FRAME	AXIS	REAL: Scale factor	Parameter 3–15 same as 1 ...	Parameter 4–16 same as 2 ...	Scale: Scale factor for multiple axes. The maximum number of parameters is 2 * maximum number of axes (axis identifier and value respectively). One axis identifier is programmed at a time, with its respective value in the next parameter. CSCALE can be used to program scale factors for up to 8 axes.
CMIRROR	FRAME	AXIS		Parameter 2–8 same as 1 ...		Mirror: Mirror on a coordinate axis
MEAFRAME	FRAME	2-dim. REAL array	2-dim. REAL array	Parameter 3: REAL variable		Frame calculation from 3 measuring points in space

Frame functions CTRANS, CSCALE, CROT and CMIRROR are used to generate frame expressions.

2. Geometry functions

Vocabulary word/ function identifier	Result	Parameter 1	Parameter 2	Parameter 3	Description
CALCDAT	BOOL: Error status	VAR REAL [,2]: Table with input points (abscissa and ordinate for points 1, 2, 3 etc.)	INT: Number of input points for calculation (3 or 4)	VAR REAL [3]: Result: Abscissa, ordinate and radius of calculated circle center point	CALCDAT: calculate circle data Calculates radius and center point of a circle from 3 or 4 points (according to parameter 1) which must lie on a circle. The points must be different.

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

2. Geometry functions					
Vocabulary word/ function identifier	**Result**	**Parameter 1**	**Parameter 2**	**Parameter 3**	**Description**
INTERSEC	BOOL: Error status	VAR REAL [11]: First contour element	VAR REAL [11]: Second contour element	VAR REAL [2]: Result vector: intersection coordinate, abscissa and ordinate	Intersection: Calculation of intersection The intersection between two contour elements is calculated. The intersection coordinates are return values. The error status indicates whether an intersection was found.

3. Axis functions				
	Result	**Parameter 1**	**Parameter 2**	**Description**
AXNAME	AXIS: Axis identifier	STRING [ ]: Input string		AXNAME: Get axis identifier Converts the input string to an axis identifier. An alarm is generated if the input string does not contain a valid axis identifier.
SPI	AXIS: Axis identifier	INT: Axis number		SPI: Convert spindle to axis Converts a spindle number to an axis identifier. An alarm is generated if the passed parameter does not contain a valid spindle number.
ISAXIS	BOOL TRUE: Axis exists: Otherwise: FALSE	INT: Number of the geometry axis (1 to 3)		Check whether the geometry axis 1 to 3 specified as parameter exists in accordance with $MC_AXCONF_GEOAX_ASSIGN_TAB.
AXSTRING	STRING	AXIS		Convert axis identifier into string.

4. Tool management				
	Result	**Parameter 1**	**Parameter 2**	**Description**
NEWT #	INT: T number	STRING [32]: Tool identifier	INT: Duplo number	Create new tool (prepare tool data). The duplo number can be omitted.
GETT #	INT: T number	STRING [32]: Tool identifier	INT: Duplo number	Get T number for tool identifier
GETACTT #	INT: Status	INT: T number	STRING [32]: tool name	Get active tool from a group of tools with the same name

#) The vocabulary word is **not** valid for SINUMERIK FM-NC.

5. Arithmetics				
	Result	**Parameter 1**	**Parameter 2**	**Description**
SIN	REAL	REAL		Sine
ASIN	REAL	REAL		Arcsine
COS	REAL	REAL		Cosine
ACOS	REAL	REAL		Arccosine
TAN	REAL	REAL		Tangent
ATAN2	REAL	REAL	REAL	Arctangent 2
SQRT	REAL	REAL		Square root
POT	REAL	REAL		Square

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

5. Arithmetics				
	Result	**Parameter 1**	**Parameter 2**	**Description**
TRUNC	REAL	REAL		Truncate decimal places
ROUND	REAL	REAL		Round decimal places
ABS	REAL	REAL		Generate absolute value
LN	REAL	REAL		Natural logarithm
EXP	REAL	REAL		Exponential function e^x

6. String functions				
	Result	**Parameter 1**	**Parameter 2 to 3**	**Description**
ISNUMBER	BOOL	STRING		Check whether the input string can be converted to a number. Result is TRUE if conversion is possible.
ISVAR	BOOL	STRING		Check whether the transfer parameter contains a variable known in the NC. (Machine data, setting data, system variable, general variables like GUD´s Result is TRUE, if all the following check produce a positive results according to the (STRING) transfer parameter: – the identifier exists – it is a 1- or 2-dimensional array – an array index is allowed For axial variables, the axis names are accepted as an index but not checked.
NUMBER	REAL	STRING		Convert the input string into a number.
TOUPPER	STRING	STRING		Convert all alphabetic characters in the input string to upper case.
TOLOWER	STRING	STRING		Convert all alphabetic characters in the input string to lower case.
STRLEN	INT	STRING		Result is the length of the input string up to the end of the string (0).
INDEX	INT	STRING	CHAR	Find the character (2nd parameter) in the input string (1st parameter). The reply gives the place at which the character was first found. The search is from left to right. The 1st character in the string has the index 0.
RINDEX	INT	STRING	CHAR	Find the character (2nd parameter) in the input string (1st parameter). The reply gives the place at which the character was first found. The search is from right to left. The 1st character in the string has the index 0.
MINDEX	INT	STRING	STRING	Find one of the characters specified in the 2nd parameter in the input string (1st parameter). The place where one of the characters was first found is output. The search is from left to right. The first character in the string has the index 0.
SUBSTR	STRING	STRING	INT	Returns the substring of the input string (1st parameter) defined by the start character (2nd parameter) and number of characters (3rd parameter). Example: SUBSTR("Hello world",1,5) returns "ello"

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810D | 840Di

12.4.4 Data types

Data types		
Type	**Comment**	**Value range**
INT	Integers with sign	± (2^{31} –1)
REAL	Real numbers (fractions with decimal point, LONG REAL to IEEE)	± (10^{-300} ... 10^{+300})
BOOL	Boolean value TRUE, FALSE or 1, 0	1, 0
CHAR	1 character in ASCII code	0 ... 255
STRING	Character string, number of characters in [...] (max. of 200 characters)	Sequence of values with 0 ... 255
AXIS	Axis identifiers only (axis addresses)	All axis identifiers available on the channel
FRAME	Geometrical parameters for translation, rotation, scale, mirror	—

■

Appendix

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

A Abbreviations

ASCII	American Standard Code for Information Interchange
ASUB	Asynchronous Subroutine
BA	Mode of operation
BAG	Mode Group
BCD	Binary Coded Decimals
BCS	Basic Coordinate System
BIN	Binary Files
BP	Basic Program
C Bus	Communication Bus
C1 .. C4	Channel 1 to channel 4
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CNC	Computerized Numerical Control
COM	Communication
COR	Coordinate Rotation
CPU	Central Processing Unit
CR	Carriage Return
CSF	Control System Flowchart (PLC programming method)
CTS	Clear To Send (serial data interfaces)
CUTOM	Cutter Radius Compensation (Tool radius compensation)

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

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: The rotary axis is moved along the shortest path to the absolute position within one revolution.

DCE Data Communications Equipment

DDE Dynamic Data Exchange

DIO Data Input/Output: Data transfer display

DIR Directory

DLL Dynamic Link Library: Module which can be accessed by a running program. Often contains program sections that are required by different programs.

DOS Disk Operating System: Operating system

DPM Dual-Port Memory

DPR Dual-Port RAM

DRAM Dynamic Random Access Memory

DRF Differential Resolver Function

DRY Dry Run

DSB Decoding Single Block

DTE Data Terminal Equipment

DW Data Word

EIA Code Special punchtape code, number of punched holes per character always odd

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

ENC	Encoder
EPROM	Erasable Programmable Read Only Memory
FB	Function Block
FC	Function Call: Function block in the PLC
FDD	Feed Drive
FDD	Floppy Disk Drive
FEPROM	Flash EPROM
FIFO	First-In-First-Out: Memory which operates without address specification from which data are read in the same order as they are stored.
FM	Function Module
FM-NC	Function Module – Numerical Control
FPU	Floating Point Unit
FRA	Frame Block
FRAME	Data Record (frame)
FST	Feed Stop
GUD	Global User Data
HD	Hard Disk
HEX	Abbreviation for hexadecimal
HHU	Handheld Unit
HMI	Human Machine Interface: SINUMERIK operating functions for operator control, programming and simulation. MMC and HMI are identical in meaning.
HW	Hardware

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

I	Input
I/O	Input/Output
I/RF	Infeed/Regenerative Feedback Unit (power supply) of SIMODRIVE 611(D)
IK (GD)	Implicit Communication (Global Data)
IKA	Interpolative Compensation Interpolative compensation
IM	Interface Module
IMR	Interface Module Receive
IMS	Interface Module Send
INC	Increment Increment
INI	Initializing Data
IPO	Interpolator
IS	Interface Signal
ISO Code	Special punchtape code, number of punched holes per character always even
JOG	Jog mode
$K_Ü$	Transmission Ratio
K_v	Servo Gain Factor
LAD	Ladder Diagram (PLC programming method)
LEC	Leadscrew Error Compensation
LF	Line Feed
LUD	Local User Data
MB	Megabyte

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

MC Measuring Circuit

MCP Machine Control Panel

MCS Machine Coordinate System (Machine)

MD Machine Data

MDA Manual Data Automatic (MDI)

MMC Human Machine Communication: User interface on numerical control systems for operator control, programming and simulation. MMC and HMI are identical in meaning.

MPF Main Program File: NC parts program (main program)

MPI Multi Port Interface

MSD Main Spindle Drive

NC Numerical Control

NCK Numerical Control Kernel (with block preparation, traversing range, etc.)

NCU Numerical Control Unit: Hardware unit of the NCK

NURBS Non Uniform Rational B-Spline

O Output

OB Organization Block in the PLC

OEM Original Equipment Manufacturer: The manufacturer of equipment that is marketed by another vendor, typically under a different name.

OI Operator Interface

OP Operator Panel

OP Operator Panel

OPI Operator Panel Interface

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

P Bus	Peripheral Bus
PC	Personal Computer
PCIN	Name of SW for exchanging data with the control system
PCMCIA	Personal Computer Memory Card International Association
PDB	Product Database
PG	Programming Device
PLC	Programmable Logic Control
PP	Production Planning
RAM	Random Access Memory (read-write memory)
REF	Reference Point Approach Function
REPOS	Reposition Function
ROV	Rapid Override
RPA	R Parameter Active: Memory area in the NCK for R-NCK for R parameter numbers
RPY	Roll Pitch Yaw: Type of coordinate system rotation
RS-232	Serial Interface (definition of interchange lines between DTE and DCE)
RTS	Request To Send (serial data interfaces)
SBL	Single Block
SD	Setting Date
SDB	System Data Block
SEA	Setting Data Active: Identification (file type) for setting data
SFB	System Function Block

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

SFC	System Function Call
SK	Softkey
SKP	Skip Block
SM	Stepper Motor
SOP	Shopfloor-Oriented Programming
SPF	Sub-Program File (subroutine file)
SR	Subroutine
SRK	Cutter Radius Compensation
SRAM	Static RAM (battery-backed)
SSI	Serial Synchronous Interface
STL	Statement List
SW	Software
SYF	System Files
T	Tool
TC	Tool Change
TEA	Testing Data Active: Identifier for machine data
TLC	Tool Length Compensation
TNRC	Tool Nose Radius Compensation
TO	Tool Offset
TOA	Tool Offset Active: Identification (file type) for tool offsets

840D NCU 571 840D NCU 572 NCU 573 810D 840Di

TRANSMIT	Transform Milling into Turning: Coordinate conversion on turning machines for milling operations
TRC	Tool Radius Compensation
UFR	User Frame: Zero offset
WCS	Workpiece Coordinate System (Work)
WOA	Work Offset Active
WPD	Workpiece Directory
WO	Work Offset
ZO	Zero Offset (WO)
ZOA	Zero Offset Active (WOA): Identification (file type) for zero offset data

840D
NCU 571

840D
NCU 572
NCU 573

810D

840Di

B Terms

Important terms are listed below in alphabetical order, accompanied by explanations. Cross-references to other entries in this glossary are indicated by the symbol "->".

A

A spline

The A spline runs tangentially through the programmed interpolation points (3rd degree polynomial).

Absolute dimension

A destination for an axis movement is defined by a dimension that refers to the origin of the currently active coordinate system. See also -> incremental dimension.

AC control (Adaptive Control)

A process variable (e.g. path-specific or axial feedrate) can be controlled as a function of another, measured process variable (e.g. spindle current). Typical application: To maintain a constant chip removal volume during grinding.

Acceleration with jerk limitation

In order to obtain the optimum acceleration gradient for the machine while providing effective protection for the mechanical components, the machining program offers a choice between instantaneous acceleration and continuous (smooth) acceleration.

Access rights

The CNC program blocks and data are protected by a 7-level system of access restrictions:

- Three password levels for system manufacturers, machine manufacturers and users and
- Four keyswitch settings which can be evaluated via the PLC.

Activate/deactivate

Working area limitation is a means of restricting the axis movement over and above the restrictions imposed by the limit switches. A pair of values delimiting the protected zone area can be specified for each axis.

Address

Addresses are fixed or variable identifiers for axes (X, Y, ...), spindle speed (S), feedrate (F), circle radius (CR), etc.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Alarms

All -> messages and alarms are displayed in plain text on the operator panel. Alarm text also includes the date, time and corresponding symbol for the reset criterion.
Alarms and messages are displayed separately.

1. Alarms and messages in the parts program
   Alarms and messages can be displayed directly from the parts program in plaintext.
2. Alarms and messages from PLC
   Alarms and messages relating to the machine can be displayed from the PLC program in plaintext. No additional function block packages are required for this purpose.

Analog input/output module

Analog input/output modules are signal transducers for analog process signals.
Analog input modules convert analog measured values into digital values that can be processed in the CPU.
Analog output modules convert digital values into manipulated variables.

Approach fixed machine point

Approach motion towards one of the predefined -> fixed machine points.

Archiving

Exporting files and/or directories to an **external** storage device.

Asynchronous subroutine

- A parts program that can be started asynchronously (or independently) by means of an interrupt signal (e.g. "High-speed NC input" signal) while the parts program is active (SW package 3 and lower).

- A parts program that can be started asynchronously (or independently) of the current program status by means of an interrupt signal (e.g. "High-speed NC input" signal) (SW package 4 and higher).

Automatic

Control system operating mode (block-sequential to DIN): Mode in NC systems in which a -> parts program is selected and continuously executed.

Auxiliary functions

Auxiliary functions can be used to pass -> parameters to the -> PLC in -> parts programs, triggering reactions there which are defined by the machine manufacturer.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Axes
CNC axes are classified according to their functional scope as:
- Axes: Interpolative path axes
- Positioning axes: Non-interpolative infeed and positioning axes with axis-specific feedrates; axes can move across block limits. Positioning axes need not be involved in workpiece machining as such and include tool feeders, tool magazines, etc.

Axis address
See -> axis identifier

Axis identifier
In compliance with DIN 66217, axes are identified as X, Y and Z for a right-handed rectangular -> coordinate system.
-> Rotary axes rotating around X, Y, Z are assigned the identifiers A, B, C. Additional axes, which are parallel to those specified, can be identified with other letters.

Axis name
See -> axis identifier

Axis/spindle replacement
An axis/spindle is permanently assigned to a particular channel via machine data. This MD assignment can be "undone" by program commands and the axis/spindle then assigned to another channel.

B

B spline
The programmed positions for the B spline are not interpolation points, but merely "check points". The curve generated does not pass directly through these check points, but only in their vicinity (1st, 2nd or 3rd degree polynomial).

Back up
A copy of the memory contents (hard disk) stored on an external device for data backup and/or archiving.

Backlash compensation
Compensation of a mechanical machine backlash, e.g. backlash due to reversal of leadscrews. The backlash compensation can be entered separately for each axis.

Backup battery
The backup battery provides non-volatile storage for the -> user program in the -> CPU and ensures that defined data areas and flags, timers and counters are retentive.

Base axis
Axis whose setpoint or actual value is employed in calculating a compensatory value.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Basic coordinate system Cartesian coordinate system, is mapped onto machine coordinate system by means of transformation.
In the -> parts program, the programmer uses the axis names of the basic coordinate system. The basic coordinate system exists in parallel to the -> machine coordinate system when no -> transformation is active. The difference between the systems relates only to the axis identifiers.

Baud rate Rate at which data transmission takes place (bit/s).

Blank The unmachined workpiece.

Block All files required for programming and program execution are known as blocks.

Block A section of a -> parts program terminated with a line feed.
A distinction is made between -> main blocks and -> subblocks.

Block search The block search function allows selection of any point in the parts program at which machining must start or be continued. The function is provided for the purpose of testing parts programs or continuing machining after an interruption.

Booting Loading the system program after Power ON.

Bus connector A bus connector is an S7-300 accessory that is supplied with the -> I/O modules. The bus connector extends the -> S7-300 bus from the -> CPU or an I/O module to the next adjacent I/O module.

C

C axis Axis about which the tool spindle describes a controlled rotational and positioning movement.

C spline The C spline is the best known and the most widely used spline. The spline passes through each of the interpolation points at a tangent and along the axis of curvature. 3rd-degree polynomials are used.

Channel structure The channel structure makes it possible to process the -> programs of individual channels simultaneously and asynchronously.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Circular interpolation
The -> tool is required to travel in a circle between defined points on the contour at a specified feed while machining the workpiece.

Clearance control (3D), sensor-driven
A position offset for a specific axis can be controlled as a function of a measured process variable (e.g. analog input, spindle current...). This function can automatically maintain a constant clearance to meet the technological requirements of the machining operation.

CNC
-> NC

CNC high-level language
The high-level language offers: -> user variables, -> predefined user variables, -> system variables, -> indirect programming, -> arithmetic and angular functions, -> relational and logic operations, -> program jumps and branches, -> program coordination (SINUMERIK 840D), -> macros.

CNC programming language
The CNC programming language is based on DIN 66025 with high-level language expansions. The -> CNC programming language and -> high-level language expansions support the definition of macros (sequenced statements).

COM
Numerical control component for the implementation and coordination of communication.

Command axis
Command axes are started from synchronized actions in response to an event (command). They can be positioned, started and stopped fully asynchronous to the parts program.

Compensation axis
Axis having a setpoint or actual value modified by the compensation value.

Compensation table
Table of interpolation points. It supplies the compensation values of the compensation axis for selected positions of the base axis.

Compensation value
Difference between the axis position measured by the position sensor and the desired, programmed axis position.

Connecting cables
Connecting cables are pre-assembled or user-assembled 2-wire cables with a connector at each end. They are used to connect the -> CPU via the -> multipoint interface (MPI) to a -> programming device or to other CPUs.

840D NCU 571 840D NCU 572 NCU 573 810D 840Di

Continuous-path mode
The purpose of continuous-path mode is to avoid rapid deceleration -> of the path axes at parts program block boundaries and to make the transition to the next block at as constant a velocity as possible.

Contour
Outline of a -> workpiece.

Contour monitoring
The following error is monitored within a definable tolerance band as a measure of contour accuracy. Overloading of the drive, for example, may result in an unacceptably large following error. In such cases, an alarm is output and the axes stopped.

Coordinate system
See -> machine coordinate system, -> workpiece coordinate system

CPU
Central Processor Unit -> programmable controller

Cycle
Protected subroutine for executing a recurring machining operation on the -> workpiece.

Cycles support
The available cycles are listed in menu "Cycle support" in the "Program" operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plaintext.

D

Data block

1. Data unit of the -> PLC which can be accessed by -> HIGHSTEP programs.
2. Data unit of the -> NC: Data blocks contain data definitions for global user data. These data can be initialized directly when they are defined.

Data transfer program PCIN
PCIN is a routine for transmitting and receiving CNC user data, e.g. parts programs, tool offsets, etc. via the serial interface. The PCIN program can run under MS-DOS on standard industrial PCs.

Data word
A data unit, two bytes in size, within a -> PLC data block.

Deletion of distance-to-go
Command in parts program which stops machining and clears the remaining path distance to go.

Design

- The SINUMERIK FM-NC is installed in the CPU tier of the SIMATIC S7-300. The 200mm wide, fully encapsulated module has the same external design as the SIMATIC S7-300 modules.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

- The SINUMERIK 840D is installed as a compact module in the SIMODRIVE 611D converter system. It has the same dimensions as a 50mm wide SIMODRIVE 611D module. The SINUMERIK 840D comprises the NCU module and the NCU box.

- The SINUMERIK 810D has the same design as the SIMODRIVE 611D with a width of 150mm. The following components are integrated: SIMATIC S7-CPU, 5 digital servo drive controls and 3 SIMODRIVE 611D power modules.

Diagnosis

1. Control operating area
2. The control incorporates a self-diagnosis program and test routines for servicing: Status, alarm and service displays.

Digital input/output module

Digital modules are signal transducers for binary process signals.

Dimensions in metric and inch systems

Position and lead/pitch values can be programmed in inches in the machining program. The control is set to a basic system regardless of the programmable unit of measure (G70/G71).

DRF

Differential Resolver Function NC function which generates an incremental zero offset in AUTOMATIC mode in conjunction with an electronic handwheel.

Drift compensation

When the CNC axes are in the constant motion phase, automatic drift compensation is implemented in the analog speed control.

Drive

- SINUMERIK FM-NC has an analog ±10V interface to the SIMODRIVE 611A converter system.
- The SINUMERIK 840D control system is linked to the SIMODRIVE 611D converter system via a high-speed digital parallel bus.

Editor

The editor makes it possible to create, modify, extend, join and insert programs/texts/program blocks.

Electronic handwheel

Electronic handwheels can be used to traverse the selected axes simultaneously in manual mode. The handwheel clicks are analyzed by the increment analyzer.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Exact stop When an exact stop is programmed, a position specified in the block is approached accurately and, where appropriate, very slowly. In order to reduce the approach time, -> exact stop limits are defined for rapid traverse and feed.

Exact stop limit When all path axes reach their exact stop limits, the control responds as if it had reached its destination point precisely. The -> parts program continues execution at the next block.

External zero offset A zero offset specified by the -> PLC.

F

Fast retraction from contour When an interrupt is received, it is possible to initiate a motion via the CNC machining program which allows the tool to be retracted quickly from the workpiece contour currently being machined. The retraction angle and the distance retracted can also be parameterized. An interrupt routine can be executed after the rapid retraction. (SINUMERIK 840D, 840Di, 810D).

Feedforward control, dynamic Contour inaccuracies resulting from following errors can be almost completely eliminated by the dynamic, acceleration-dependent feedforward control function. Feedforward control ensures an excellent degree of machining accuracy even at high tool path velocities. Feedforward control can only be selected or deselected for all axes together via the parts program.

Feedrate override The current feedrate setting entered via the control panel or by the PLC is overlaid on the programmed feedrate (0–200%). The feedrate can also be corrected by a programmable percentage factor (1–200%) in the machining program.
An offset can also be applied via motion-synchronous actions independently of the running program.

Finished-part contour Contour of the finished workpiece. See also -> blank.

Fixed machine point A point defined uniquely by the machine tool, such as the reference point.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Fixed-point approach

Machine tools can execute defined approaches to fixed points such as tool-change points, loading points, pallet-change points, etc. The coordinates of these points are stored on the control. Where possible, the control moves these axes in -> rapid traverse.

Frame

A frame is a calculation rule that translates one Cartesian coordinate system into another Cartesian coordinate system. A frame contains the components -> zero offset, -> rotation, -> scaling and -> mirroring.

G

General reset

The following memories of the -> CPU are erased by a general reset operation:

- -> Working memory
- Read/write area of the -> load memory
- -> System memory
- -> Backup memory

Geometry

Description of a -> workpiece in the -> workpiece coordinate system.

Geometry axis

Geometry axes are used to describe a 2- or 3-dimensional area in the workpiece coordinate system.

Global main run/subroutine

Each global main run/subroutine can be stored only once under its name in the directory. However, the same name can be used in different directories.

Ground

"Ground" is the term applied to all the electrically inactive, interconnected parts of a piece of equipment which cannot carry any hazardous contact voltage even in the event of a fault.

H

Helical interpolation

The helical interpolation function is ideal for machining internal and external threads using form milling cutters and for milling lubrication grooves. The helix comprises two movements:

1. Circular movement in one plane
2. Linear movement perpendicular to this plane.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

High-speed digital inputs/outputs
As an example, high-speed CNC program routines (interrupt routines) can be started via the digital inputs. High-speed, program-driven switching functions can be initiated via the digital CNC outputs (SINUMERIK 840D).

HIGHSTEP
Combination of the programming features for the -> PLC in the S7-300/400 range.

I

I/O module
I/O modules create the link between the CPU and the process. I/O modules are:

- ->Digital input/output modules
- ->Analog input/output modules
- ->Simulator modules

Identifier
In accordance with DIN 66025, identifiers (names) for variables (arithmetic variables, system variables, user variables), for subroutines, for vocabulary words and for words can contain several address letters. These letters have the same meaning as the words in the block syntax. Identifiers must be unique. Identical identifiers must not be used for different objects.

Inch system of measurement
System of measurement that defines distances in "inches" and fractions thereof.

Inclined axis
Fixed angular interpolation with allowance for an inclined infeed axis or grinding wheel through specification of the angle. The axes are programmed and displayed in the Cartesian coordinate system.

Increment
Travel path length specification based on number of increments. The number of increments can be stored as a -> setting data or selected with keys labeled with 10, 100, 1000, 10 000.

Incremental dimension
A destination for axis traversal is defined by a distance to be covered and a direction referenced to a point already reached. See also -> absolute dimension.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Initialization block

Initialization blocks are special -> program blocks. They contain values which must be assigned before the program is executed.
Initialization blocks are used primarily for initializing predefined data or global user data.

Initialization file

An initialization file can be created for each -> workpiece. In it, the various variable value instructions which apply exclusively to one workpiece can be stored.

Intermediate blocks

Movements with selected tool offset (G41/G42) can be interrupted by a limited number of intermediate blocks (blocks without axis motions in the offset plane). When such blocks are used, the tool offset can still be calculated correctly. The permissible number of intermediate blocks read in advance by the control can be set via system parameters.

Interpolation cycle

The interpolation cycle is a multiple of the basic system cycle. It specifies the cycle time for updating the setpoint interface to the position controllers. The interpolation cycle determines the resolution of the velocity profiles.

Interpolative compensation

Interpolative compensation provides a means of compensating for leadscrew errors (LEC) and measuring-system errors (MSEC) resulting from the production process.

Interpolator

Logical unit of the -> NCK which determines intermediate values for the movements to be traversed on the individual axes on the basis of destination positions specified in the parts program.

Interrupt routine

Interrupt routines are special -> subroutines which can be started by events (external signals) in the machining process. The parts program block being processed is aborted and the axis position at the instant of interruption is stored automatically.
See -> ASUB

Inverse-time feedrate

On SINUMERIK 840D controls, it is possible to program the time required to traverse the path of a block instead of the feedrate speed for the axis movement (G93).

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

J

JOG

Control system operating mode (setup): The machine can be set up in Jog mode. Individual axes and spindles can be jogged by means of direction keys. Other functions in Jog mode are -> reference point approach, -> Repos and -> Preset -> (set actual value).

K

Keyswitch

1. **S7-300**: The keyswitch is the mode selector switch on the -> CPU. The keyswitch is operated by means of a removable key.
2. **840D**: The keyswitch on the -> machine control panel has 4 positions which are assigned functions by the operating system of the control. There are also three keys of different colors belonging to the keyswitch that can be removed in the specified positions.

$K_Ü$

Transmission Ratio

K_v

Servo gain factor, control variable of a control loop

L

Languages

The user interface texts, system messages and alarms are available in five system languages (floppy disk):
English, French, German, Italian and **Spanish**.
The user can select **two** of the listed languages at a time in the control.

Leadscrew error compensation

Compensation of mechanical inaccuracies in a leadscrew involved in the feed motion. Errors are compensated by the control based on stored deviation measurements.

Limit speed

Minimum/maximum (spindle) speed: The maximum speed of a spindle can be limited by values defined in the machine data, the -> PLC or -> setting data.

Linear axis

The linear axis is an axis which, in contrast to a rotary axis, describes a straight line.

Linear interpolation

The tool travels along a straight line to the destination point while machining the workpiece.

Look Ahead

The **Look Ahead** function is a means of optimizing the machining velocity by looking ahead over a parameterizable number of traversing blocks.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Look Ahead for contour violations
The control detects and reports the following types of collision:
1. Path is shorter than tool radius.
2. Width of inside corner is less than the tool diameter.

M

Machine
Control operating area

Machine axes
Axes which exist physically on the machine tool.

Machine control panel
An operator panel on a machine tool with operating elements such as keys, rotary switches, etc. and simple indicators such as LEDs. It is used for direct control of the machine tool via the PLC.

Machine coordinate system
System of coordinates based on the axes of the machine tool.

Machine zero
A fixed point on the machine tool which can be referenced by all (derived) measurement systems.

Machining channel
A channel structure makes it possible to reduce downtimes by allowing sequences of motions to be executed in parallel. For example, a loading gantry can execute its movements during a machining operation. In this case, a CNC channel ranks as an autonomous CNC control complete with decoding, block preparation and interpolation.

Macros
Multiple programming language instructions can be combined in a single statement. This abbreviated sequence of instructions is called in the CNC program under a user-defined name. The macro executes the instructions sequentially.

Main block
A block prefixed by ":" containing all the parameters required to start execution of a -> parts program.

Main program
-> Parts program identified by a number or name in which other main programs, subroutines or -> cycles may be called.

Main run
Parts program blocks which have been decoded and prepared by the preprocessor are executed during the "main run".

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

MDA

Control system operating mode: Manual Data Automatic. In the MDA mode, individual program blocks or block sequences with no reference to a main program or subroutine can be input and executed immediately afterwards through actuation of the NC Start key.

Measuring circuits

- SINUMERIK FM-NC: The requisite control circuits for axes and spindles are integrated in the control module as standard. A maximum total of 4 axes and spindles can be implemented, with no more than 2 spindles.
- SINUMERIK 840D: The signals from the sensors are analyzed in the SIMODRIVE 611D drive modules. The maximum configuration depends on the HW and/or SW of the control. Up to 12 axes and spindles are permissible per channel.
  A maximum of 31 axes or 20 spindles are permissible per NCU.

Messages

All messages programmed in the parts program and -> alarms detected by the system are displayed in plain text on the operator panel. Alarms and messages are displayed separately.

Metric system

Standardized system of units for lengths in millimeters (mm), meters (m), etc.

Mirroring

Mirroring exchanges the leading signs of the coordinate values of a contour in relation to an axis. Analogously, several axes can be mirrored simultaneously.

Mode

An operating concept on a SINUMERIK control. The modes -> Jog, -> MDA, -> Automatic are defined.

Mode group

All axes/spindles are assigned to one and only one channel at any given time. Each channel is assigned to a mode group
The same -> mode is always assigned to the channels of a mode group.

Motion synchronization

This function can be used to initiate actions that are synchronized with the machining operation. The starting point of the actions is defined by a condition (e.g. status of a PLC input, time elapsed since beginning of a block). The start of motion-synchronous actions is not tied to block boundaries. Examples of typical motion-synchronous actions are: Transfer M and H auxiliary functions to the PLC or deletion of distance-to-go for specific axes.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Multipoint interface

The multipoint interface (MPI) is a 9-pin sub-D port. A parameterizable number of devices can be connected to an MPI for the purpose of communicating with one another:

- Programming devices
- MMI (HMI) systems
- Other automation systems

The "Multipoint Interface MPI" parameter block of the CPU contains the -> parameters which define the properties of the multipoint interface.

N

NC

Numerical Control It incorporates all the components of the machine tool control system: -> NCK, -> PLC, -> MMC, -> COM.
Note: CNC (computerized numerical control) would be a more appropriate description for the SINUMERIK 840/840iD or 810D controls. computerized numerical control.

NCK

Numerical Control Kernel: Component of the NC control which executes -> parts programs and essentially coordinates the movements on the machine tool.

Network

A network is the interconnection of several S7-300s and other terminal devices such as a programming device, for example, interlinked by means of -> connecting cables. The networked devices interchange data via the network.

Node number

The node number is the "contact address" of a -> CPU or the -> programming device or another intelligent I/O module if these devices are exchanging data with one another via a -> network. The node number is assigned to the CPU or the programming device by the S7 tool -> "S7 Configuration".

NRK

Numeric Robotic Kernel (operating system of the -> NCK)

NURBS

Motion control and path interpolation are implemented internally in the control on the basis of NURBS (Non-Uniform Rational B Splines). A standard procedure is thus available (SINUMERIK 840D/840iD) as an internal control function for all modes of interpolation.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

O

Oblique-plane machining
Drilling and milling operations on workpiece surfaces which are oblique to the coordinate planes of the machine are supported by the "Oblique surface machining" function. The position of the oblique plane can be defined by inclining the coordinate system (see FRAME programming).

OEM
The scope for implementing individual solutions (OEM applications) for the SINUMERIK 840D has been provided for machine manufacturers who wish to create their own operator interface or integrate process-oriented functions in the control.

Offset memory
Data area in the control in which tool offset data are stored.

Online tool offset
This function can be used for grinding tools only.
The reduction in size of the grinding wheel resulting from dressing is transferred as a tool offset to the currently active tool and immediately applied.

Operator interface
The operator interface (OI) is the human-machine interface of a CNC. It takes the form of a screen and has eight horizontal and eight vertical softkeys.

Oriented spindle stop
Stops the workpiece spindle at a specified orientation angle, e.g. to perform an additional machining operation at a specific position.

Oriented tool retraction
RETTOOL: If machining is interrupted (e.g. when a tool breaks), a program command can be used to retract the tool in a user-specified orientation by a defined distance.

Override
Manual or programmable control feature which enables the user to override programmed feedrates or speeds in order to adapt them to a specific workpiece or material.

P

Parameters
1. **S7-300**: The S7-300 uses two types of parameter:
   - Parameter of a STEP7 statement
     A parameter of a STEP7 statement is the address of the operand to be processed or a constant.
   - Parameter of a -> parameter block
     A parameter of a parameter block determines the behavior of a module.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

2. **840D/840Di/810D**:
   - Control operating area
   - Computation parameter, can be set any number of times or queried by the programmer for any purpose in the parts program.

Parts program
A sequence of instructions to the NC control which combine to produce a specific -> workpiece by performing certain machining operations on a given -> blank.

Parts program management
The parts program management function can be organized according to -> workpieces. The quantity of programs and data to be managed is dependent on the control memory capacity and can also be configured via MD settings. Each file (programs and data) can be given a name consisting of a maximum of 16 alphanumeric characters.

Path axis
Path axes are all the machining axes in the -> channel which are controlled by the -> interpolator such that they start, accelerate, stop and reach their end positions simultaneously.

Path feed
The path feed acts on -> path axes. It represents the geometrical sum of the feeds on the participating -> path axes.

Path velocity
The maximum programmable path velocity depends on the input resolution. With a resolution of 0.1mm, for example, the maximum programmable path velocity is 1000 m/min.

PG
Programming Device

PLC
Programmable Logic Control Component of the -> NC: Programmable controller for processing the control logic on the machine tool.

PLC program memory

- SINUMERIK 840D/840Di:
  The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory. The PLC user memory can be expanded using additional memory module

- SINUMERIK 810D:
  The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory on the CPU. An optional memory expansion is available for the user memory provided in the basic configuration of the S/-CPU.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

PLC programming
The PLC is programmed with the **STEP7** software. The STEP 7 programming software is based on the standard **WINDOWS** operating system and incorporates the functionality of STEP5 programming with innovative expansions and developments.

Polar coordinates
A coordinate system which defines the position of a point on a plane in terms of its distance from the origin and the angle formed by the radius vector with a defined axis.

Polynomial interpolation
Polynomial interpolation provides a means of generating a very wide range of curves, including **straight-line, parabolic and exponential functions** (SINUMERIK 840D/840Di/810D).

Positioning axis
An axis which performs an auxiliary movement on a machine tool (e.g. tool magazine, pallet transport). Positioning axes are axes that do not interpolate with the -> path axes.

Power ON
The action of switching the control off and then on again.

Preprocessing memory, dynamic
The traversing blocks are preprocessed prior to execution and stored in a "preprocessing memory". Block sequences can be executed at a very fast rate from the memory. Blocks are uploaded continuously to the preprocessing memory during machining.

Preprocessing stop
Program command. The next block in a parts program is not executed until all other blocks which have already been preprocessed and stored in the preprocessing memory have been executed -> "main run".

Preset
The control zero point can be redefined in the machine coordinate system by means of the Preset function. Preset does not cause the axes to move; instead, a new position value is entered for the current axis positions.

Program
1. Control operating area
2. Sequence of instructions to the control system.

Programmable frames
Programmable -> frames can be used to define new coordinate system starting points dynamically while the parts program is running. A distinction is made between absolute definition using a new frame and additive definition with reference to an existing starting point.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Programmable logic controller
Programmable logic controllers (PLC) are electronic controllers whose functions are stored as a program in the control unit. The design and wiring of the unit are not, therefore, dependent on the control functions. Programmable logic controllers have the same structure as a computer, i.e. they consist of a CPU with memory, input/output modules and an internal bus system. The I/Os and programming language are selected according to the requirements of the control technology involved.

Programmable working area limitation
Limitation of the movement area of the tool to within defined, programmable limits.

Programming key
Characters and character sequences which have a defined meaning in the programming language for -> parts programs (see Programming Guide).

Protection zone
Three-dimensional area within a -> working area which the tool tip is not permitted to enter (programmable via MD).

Q

Quadrant error compensation
Contour errors on quadrant transitions caused by frictional fluctuations on guideways can be largely eliminated by means of quadrant error compensation. A circularity test is performed to parameterize the quadrant error compensation function.

R

R parameter
Calculation parameter. The programmer can assign or request the values of the R parameter in the -> parts program as required (R variable).

Rail
This rail is used to mount the modules of the S7-300 system.

Rapid traverse
The highest traversing speed of an axis used, for example, to bring the tool from an idle position to the -> workpiece contour or retract it from the workpiece contour.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Reference point

Point on the machine tool with which the measuring system of the -> machine axes is referenced.

Reference point approach

If the position measuring system used is not an absolute-value encoder, then a reference point approach operation is required to ensure that the actual values supplied by the measuring system are in accordance with the machine coordinate values.

REPOS

1. Reapproach contour, triggered by operator
   REPOS allows the tool to be returned to the interrupt position by means of the direction keys.
2. Programmed contour reapproach
   A selection of approach strategies are available in the form of program commands: Approach point of interruption, approach start of block, approach end of block, approach a point on the path between start of block and interruption.

Revolutional feedrate

The axis feedrate is adjusted as a function of the speed of the master spindle in the channel (programmed with G95).

Rigid tapping

This function is used to tap holes without the use of a compensating chuck. The spindle is controlled as an interpolative rotary axis and drill axis, with the result that threads are tapped precisely to the final drilling depth, for example, in blind tapped holes (precondition: Spindle axis mode).

Rotary axis

Rotary axes cause the tool or workpiece to rotate to a specified angle position.

Rotary axis, continuously turning

The range of motion of a rotary axis can be set to a modulo value (in machine data) or defined as continuous in both directions, depending on the application. Continuously turning rotary axes are used, for example, for eccentric machining, grinding and winding.

Rotation

Component of a -> frame which defines a rotation of the coordinate system through a specific angle.

Rounding axis

Rounding axes cause the workpiece or tool to rotate to an angle position described on a graduated grid. When the grid position has been reached, the axis is "in position".

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

S

S7 Configuration

S7 Configuration is a tool for parameterizing modules. S7 Configuration is used to set a variety of -> parameter blocks of the -> CPU and the I/O modules on the -> programming device. These parameters are uploaded to the CPU.

S7-300 bus

The S7-300 bus is a serial data bus which supplies modules with the appropriate voltage and via which they exchange data with one another. The connection between the modules is made by means of -> bus connectors.

Safety functions

The control includes continuously active monitoring functions which detect faults in the -> CNC, the programmable controller (-> PLC) and the machine so early that damage to the workpiece, tool or machine rarely occurs. In the event of a fault, the machining operation is interrupted and the drives stopped. The cause of the malfunction is logged and an alarm issued. At the same time, the PLC is notified that a CNC alarm is pending.

Scaling

Component of a -> frame which causes axis-specific scale alterations.

Serial RS-232 interface

For data input/output, HMI Advanced with PCU 50 and HMI Embedded with PCU 20 has two serial RS-232 interfaces COM 1 and COM 2.
Machining programs and manufacturer and user data can be imported and exported via these interfaces.

Services

Control operating area

Setting data

Data which provide the control with information about properties of the machine tool in a way defined by the system software.
Unlike -> machine data, setting data can be modified by the user.

Softkey

A key whose name appears on an area of the screen. The choice of softkeys displayed is adapted dynamically to the operating situation. The freely assignable function keys (softkeys) are assigned to functions defined in the software.

Software limit switches

Software limit switches define the limits of the travel range of an axis and prevent the slide contacting the hardware limit switches. Two pairs of values can be assigned per axis and activated separately via the -> PLC.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Spindles

The spindle functionality is a two-level construct:

1. Spindles: Speed-controlled or position-controlled spindle drives for SINUMERIK 840D/840Di or 810D.

2. Auxiliary spindles: Speed-controlled spindle drives without actual position sensor, e.g. for power tools. "Auxiliary spindle" function package, e.g. for power tools.

Spline interpolation

Using the spline interpolation function, the control is able to generate a smooth curve from just a small number of specified interpolation points along a setpoint contour.

Standard cycles

Standard cycles are used to program machining operations which repeat frequently:

- For drilling/milling
- For measuring tools and workpieces
- For turning (SINUMERIK FM-NC 840D)

The available cycles are listed in menu "Cycle support" in the "Program" operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plaintext.

Subblock

Block prefixed by "N" containing information for a machining step such as a position parameter.

Subroutine

A sequence of instructions of a -> parts program which can be called repeatedly with different initial parameters. A subroutine is called from within a main program. Each subroutine can be blocked against unauthorized output and display with MMC 102/103 or HMI Advanced/Embedded. -> Cycles are a type of subroutine.

Synchronization

Instructions in -> parts programs for coordination of the operations in different -> channels at specific machining points.

Synchronized actions

1. Auxiliary function output
   While a workpiece is being machined, technological functions (-> auxiliary functions) can be output from the CNC program to the PLC. These auxiliary functions control, for example, ancillary equipment on the machine tool such as the sleeve, gripper, chuck, etc.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

2. High-speed auxiliary function output
   The acknowledgement times for the -> auxiliary functions can be minimized and unnecessary halts in the machining process avoided for time-critical switching functions.

Synchronized actions can be combined to form programs (technology cycles). Axis programs can be started in the same IPO cycle, for example, by scanning digital inputs.

Synchronized axes

Synchronized axes require the same amount of time to traverse their path as -> geometry axes for their path.

Synchronous spindle

Accurate angular synchronism between one master spindle and one or more slave spindles. Enables flying transfer of a workpiece from spindle 1 to spindle 2 on turning machines.
In addition to speed synchronism, it is also possible to program the relative angular positions of the spindles, e.g. on-the-fly, position-oriented transfer of inclined workpieces.
Several pairs of synchronous spindles can be implemented.

System variable

A variable which exists although it has not been programmed by the -> parts program programmer. It is defined by the data type and the variable name, which is prefixed with **$**. See also -> User-defined variable.

T

Teach In

Teach In is a means of creating or correcting parts programs. The individual program blocks can be input via the keyboard and executed immediately.
Positions approached via the direction keys or handwheel can also be stored.
Additional information such as G functions, feedrates or M functions can be entered in the same block.

Text editor

-> Editor

Tool

Tool required for machining workpieces, e.g. drill, miller, etc.

Tool

A tool employed to shape the workpiece, for example, a turning tool, milling cutter, drill, laser beam, grinding wheel, etc.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Tool nose radius compensation
A contour is programmed on the assumption that a pointed tool will be used. Since this is not always the case in practice, the curvature radius of the tool being used is specified so that the control can make allowance for it. The curvature center point is guided equidistantly to the contour at an offset corresponding to the curvature radius.

Tool offset
A tool is selected by programming a **T function** (5 decades, integer) in the block. Up to nine tool edges (D addresses) can be assigned to each T number.
The number of tools to be managed in the control is set in parameterization.
Tool length compensation is selected by programming D numbers.

Tool radius compensation
In order to program a desired -> workpiece contour directly, the control must traverse a path equidistant to the programmed contour, taking into account the radius of the tool used (G41/G42).

Transformation
Programming in a Cartesian coordinate system, execution in a non-Cartesian coordinate system (e.g. with machine axes as rotary axes). Employed in conjunction with Transmit, Inclined Axis, 5-Axis Transformation.

Transmit
This function is used to mill the outside contours on turned parts, e.g. four-sided parts (linear axis with rotary axis). 3D interpolation with two linear axes and one rotary axis is also possible.

This function brings the following benefits:
- Simplified programming
- Improved machine efficiency through complete machining: Turning and milling on the same machine without reclamping.

Travel to fixed stop
This function allows axes (tailstocks, sleeves) to be traversed to a fixed stop position in order, for example, to clamp workpieces. The contact pressure can be defined in the parts program.

Traversing range
The maximum permissible travel range for linear axes is ± 9 decades. The absolute value depends on the selected input and position control resolution and the unit of measurement (inch or metric).

840D
NCU 571

840D
NCU 572
NCU 573

810D

840Di

U

User memory
All programs and data such as parts programs, subroutines, comments, tool offsets, zero offsets/frames and channel and program user data can be stored in the common CNC user memory.

User program
-> Parts program

User-defined variable
Users can define variables in the -> parts program or data block (global user data) for their own use. A definition contains a data type specification and the variable name. See also -> system variable.

V

Variable definition
A variable is defined through the specification of a data type and a variable name. The variable name can be used to address the value of the variable.

Velocity control
In order to achieve an acceptable travel velocity in movements which call for very small adjustments of position in a block, the control can -> look ahead.

Vocabulary words
Words with a specific notation which have a defined meaning in the programming language for -> parts programs.

W

Working memory
The working storage is a Random Access Memory in the -> CPU which the processor accesses as it executes the application program.

Working space
Three-dimensional zone into which the tool tip can be moved on account of the physical design of the machine tool.
See also -> protection zone.

Work offset
-> Zero offset

Workpiece
Part to be produced/machined by the machine tool.

Workpiece contour
Setpoint contour of the -> workpiece to be produced/machined.

Workpiece coordinate system
The origin of the workpiece coordinate system (WCS) is the -> workpiece zero. In machining operations programmed in the WCS, the dimensions and directions refer to this coordinate system.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

Workpiece zero

The workpiece zero (W) forms the starting point for the -> workpiece coordinate system and defines this WCS in relation to the machine zero (M).

X

Y

Z

Zero offset

Specification of a new reference point for a coordinate system through reference to an existing zero and a -> frame.
-> Work offset

1. Settable
   SINUMERIK 840D/840Di: A parameterizable number of settable zero offsets is available for each CNC axis. Each of the zero offsets can be selected by G functions and selection is exclusive.
2. External
   All offsets which define the position of the workpiece zero can be overlaid with an external zero offset
   - defined by handwheel (DRF offset) or
   - defined by the PLC.
3. Programmable
   Zero offsets can be programmed for all path and positioning axes by means of the TRANS instruction.

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

C References

General Documentation

/BU/ SINUMERIK 840D/840Di/810D/802S, C, D
Ordering Information
Catalog NC 60
Order number: E86060-K4460-A101-A9-7600

/ST7/ **SIMATIC**
SIMATIC S7 Programmable Logic Controllers
Catalog ST 70
Order number: E86060-K4670-A111-A3

/Z/ SINUMERIK, SIROTEC, SIMODRIVE
Connections & System Components
Catalog NC Z
Order number: E86060-K4490-A001-A8-7600

Electronic Documentation

/CD1/ The SINUMERIK System (11.02 Edition)
DOC ON CD
(includes all SINUMERIK 840D/840Di/810D/802 and SIMODRIVE publications)
Order number: 6FC5 298-6CA00-0BG3

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

User Documentation

/AUK/ SINUMERIK 840D/810D
AutoTurn Short Operating Guide (10.02 Edition)
Order number: 6FC5 298-4AA30-0BP3

/AUP/ SINUMERIK 840D/810D
AutoTurn Graphic Programming System (02.02 Edition)
Operator's Guide
Programming/Setup
Order number: 6FC5 298-4AA40-0BP3

/BA/ SINUMERIK 840D/810D
Operator's Guide MMC (10.00 Edition)
Order number: 6FC5 298-6AA00-0BP0

/BAD/ SINUMERIK 840D/840Di/810D
Operator's Guide: HMI Advanced (11.02 Edition)
Order number: 6FC5 298-6AF00-0BP2

/BEM/ SINUMERIK 840D/810D
Operator's Guide: HMI Embedded (11.02 Edition)
Order number: 6FC5 298-6AC00-0BP2

/BAH/ SINUMERIK 840D/840Di/810D
Operator's Guide HT 6 (HPU New) (06.02 Edition)
Order number: 6FC5 298-0AD60-0BP2

/BAK/ SINUMERIK 840D/840Di/810D
Short Operating Guide (02.01 Edition)
Order number: 6FC5 298-6AA10-0BP0

/BAM/ SINUMERIK 810D/840D
Operator's Guide ManualTurn (08.02 Edition)
Order number: 6FC5 298-6AD00-0BP0

/BAS/ SINUMERIK 840D/810D
Operator's Guide ShopMill (11.02 Edition)
Order number: 6FC5 298-6AD10-0BP1

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

/BAT/ SINUMERIK 810D/840D
Operator's Guide ShopTurn (03.03 Edition)
Order number: 6FC5 298-6AD50-0BP2

/BNM/ SINUMERIK 840D/840Di/810D
User's Guide Measuring Cycles (11.02 Edition)
Order number: 6FC5 298-6AA70-0BP2

/CAD/ SINUMERIK 840D/840Di/810D
Operator's Guide CAD Reader (03.02 Edition)
Order number: (is part of the online help)

/DA/ SINUMERIK 840D/840Di/810D
Diagnostics Guide (11.02 Edition)
Order number: 6FC5 298-6AA20-0BP3

/KAM/ SINUMERIK 840D/810D
Short Guide ManualTurn (04.01 Edition)
Order number: 6FC5 298-5AD40-0BP0

/KAS/ SINUMERIK 840D/810D
Short Guide ShopMill (04.01 Edition)
Order number: 6FC5 298-5AD30-0BP0

/KAT/ SINUMERIK 840D/810D
Short Guide ShopTurn (07.01 Edition)
Order number: 6FC5 298-6AF20-0BP0

/PG/ SINUMERIK 840D/840Di/810D
Programming Guide Fundamentals (11.02 Edition)
Order number: 6FC5 298-6AB00-0BP2

/PGA/ SINUMERIK 840D/840Di/810D
Programming Guide Advanced (11.02 Edition)
Order number: 6FC5 298-6AB10-0BP2

/PGK/ SINUMERIK 840D/840Di/810D
Short Guide Programming (02.01 Edition)
Order number: 6FC5 298-6AB30-0BP1

/PGM/ SINUMERIK 840D/840Di/810D
Programming Guide ISO Milling (11.02 Edition)
Order number: 6FC5 298-6AC20-0BP2

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

/PGT/ SINUMERIK 840D/840Di/810D
Programming Guide ISO Turning (11.02 Edition)
Order number: 6FC5 298-6AC10-0BP2

/PGZ/ SINUMERIK 840D/840Di/810D
Programming Guide Cycles (11.02 Edition)
Order number: 6FC5 298-6AB40-0BP2

/PI/ **PCIN 4.4**
Software for Data Transfer to/from MMC Module
Order number: 6FX2 060-4AA00-4XB0 (English, French, German)
Order from: WK Fürth

/SYI/ SINUMERIK 840Di
System Overview (02.01 Edition)
Order number: 6FC5 298-6AE40-0BP0

Manufacturer/Service Documentation

a) Lists

/LIS/ SINUMERIK 840D/840Di/810D
SIMODRIVE 611D
Lists (11.02 Edition)
Order number: 6FC5 297-6AB70-0BP3

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

b) Hardware

/BH/ SINUMERIK 840D/840Di/810D
Operator Components Manual (HW) (11.02 Edition)
Order number: 6FC5 297-6AA50-0BP2

/BHA/ SIMODRIVE **Sensor**
Absolute Encoder with PROFIBUS DP
User's Guide (HW) (02.99 Edition)
Order number: 6SN1 197-0AB10-0YP1

/EMV/ SINUMERIK, SIROTEC, SIMODRIVE
EMC Installation Guideline (06.99 Edition)
Planning Guide (HW)
Order number: 6FC5 297-0AD30-0BP1

/GHA/ **ADI4 - Analog Drive Interface for 4 Axes** (09.02 Edition)
Equipment Manual
Order number: 6FC5 297-0BA01-0BP0

/PHC/ SINUMERIK 810D
Configuring Manual (HW) (03.02 Edition)
Order number: 6FC5 297-6AD10-0BP0

/PHD/ SINUMERIK 840D
Configuring Manual NCU 561.2-573.4 (HW) (10.02 Edition)
Order number: 6FC5 297-6AC10-0BP2

/PMH/ SIMODRIVE **Sensor**
Hollow-Shaft Measuring System SIMAG H
Configuring/Installation Guide (HW) (07.02 Edition)
Order number: 6SN1197-0AB30-0BP1

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

c) Software

/FB1/ SINUMERIK 840D/840Di/810D
Description of Functions, Basic Machine (Part 1) (11.02 Edition)
(the various manuals are listed below)
Order number: 6FC5 297-6AC20-0BP2

A2 Various Interface Signals
A3 Axis Monitoring, Protection Zones
B1 Continuous Path Mode, Exact Stop and Look Ahead
B2 Acceleration
D1 Diagnostic Tools
D2 Interactive Programming
F1 Travel to Fixed Stop
G2 Velocities, Setpoint/Actual Value Systems, Closed-Loop Control
H2 Output of Auxiliary Functions to PLC
K1 Mode Group, Channels, Program Operation Mode
K2 Coordinate Systems, Axis Types, Axis Configurations,
Actual-Value System for Workpiece, External Zero Offset
K4 Communication
N2 EMERGENCY STOP
P1 Transverse Axes
P3 Basic PLC Program
R1 Reference Point Approach
S1 Spindles
V1 Feeds
W1 Tool Compensation

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

/FB2/

SINUMERIK 840D/840Di/810D
Description of Functions, Extended Functions (Part 2) (11.02 Edition)
including FM-NC: Turning, Stepping Motor
(the various manuals are listed below)
Order number: 6FC5 297-6AC30-0BP2

A4	Digital and Analog NCK I/Os
B3	Several Operator Panels and NCUs
B4	Operation via PG/PC
F3	Remote Diagnostics
H1	Jog with/without Handwheel
K3	Compensations
K5	Mode Groups, Channels, Axis Replacement
L1	FM-NC Local Bus
M1	Kinematic Transformation
M5	Measurements
N3	Software Cams, Position Switching Signals
N4	Punching and Nibbling
P2	Positioning Axes
P5	Oscillation
R2	Rotary Axes
S3	Synchronous Spindles
S5	Synchronized Actions (SW 3 and lower, higher /FBSY/)
S6	Stepper Motor Control
S7	Memory Configuration
T1	Indexing Axes
W3	Tool Change
W4	Grinding

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

/FB3/ SINUMERIK 840D/840Di/810D
Description of Functions Special Functions (Part 3) (11.02 Edition)
(the various manuals are listed below)
Order number: 6FC5 297-6AC80-0BP2

F2 3-Axis to 5-Axis Transformation
G1 Gantry Axes
G3 Cycle Times
K6 Contour Tunnel Monitoring
M3 Coupled Axes and ESR (previously Coupled Motion and Master/Slave Couplings)
S8 Constant Workpiece Speed for Centerless Grinding
T3 Tangential Control
TE0 Installation and Activation of Compile Cycles
TE1 Clearance Control
TE2 Analog Axis
TE3 Speed/Torque Coupling Master-Slave
TE4 Transformation Package Handling
TE5 Setpoint Exchange
TE6 MCS Coupling
TE7 Retrace Support
TE8 Unclocked Path-Synchronous Switching Signal Output
V2 Preprocessing
W3 3D Tool Radius Compensation

/FBA/ SIMODRIVE 611D/SINUMERIK 840D/810D
Description of Functions, Drive Functions (11.02 Edition)
(the various sections are listed below)
Order number: 6SN1 197-0AA80-0BP9

DB1 Operational Messages/Alarm Reactions
DD1 Diagnostic Functions
DD2 Speed Control Loop
DE1 Extended Drive Functions
DF1 Enable Commands
DG1 Encoder Parameterization
DL1 Linear Motor MD
DM1 Calculation of Motor/Power Section Parameters and Controller Data
DS1 Current Control Loop
DÜ1 Monitors/Limitations

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

/FBAN/
SINUMERIK 840D/SIMODRIVE 611 DIGITAL
Description of Functions
ANA-MODULE (02.00 Edition)
Order number: 6SN1 197-0AB80-0BP0

/FBD/
SINUMERIK 840D
Description of Functions **Digitizing** (07.99 Edition)
Order number: 6FC5 297-4AC50-0BP0
DI1 Start-Up
DI2 Scanning with Tactile Sensors (scancad scan)
DI3 Scanning with Lasers (scancad laser)
DI4 Milling Program Generation (scancad mill)

/FBDN/
IT Solutions
NC Data Management Server (DNC NT-2000) (01.02 Edition)
Description of Functions
Order number: 6FC5 297-5AE50-0BP2

/FBDT/
SINUMERIK 840D/840Di/810D
IT Solutions
SinDNC Data Transfer via Network (09.02 Edition)
Description of Functions
Order number: 6FC5 297-5AE70-0BP0

/FBFA/
SINUMERIK 840D/840Di/810D
Description of Functions
ISO Dialects for SINUMERIK (11.02 Edition)
Order number: 6FC5 297-6AE10-0BP2

/FBFE/
SINUMERIK 840D/840Di/810D
Description of Functions **Remote Diagnostics** (11.02 Edition)
Order number: 6FC5 297-0AF00-0BP2

/FBH/
SINUMERIK 840D/840Di/810D
HMI Programming Package (11.02 Edition)
Orcer number: (is part of the SW delivery)
Part 1 User's Guide
Part 2 Description of Functions

/FBHLA/
SINUMERIK 840D/SIMODRIVE 611 digital
Description of Functions
HLA Module (04.00 Edition)
Order number: 6SN1 197-0AB60-0BP2

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

/FBMA/
SINUMERIK 840D/810D
Description of Functions **ManualTurn** (08.02 Edition)
Order number: 6FC5 297-5AD50-0BP2

/FBO/
SINUMERIK 840D/810D
Description of Functions
Configuring of OP 030 Operator Interface (09.01 Edition)
(the various sections are listed below)
Order number: 6FC5 297-6AC40-0BP0

BA Operator's Guide
EU Development Environment (Configuring Package)
PS Online only: Configuring Syntax (Configuring Package)
PSE Introduction to Configuring of Operator Interface
IK Screen Kit: Software Update and Configuration

/FBP/
SINUMERIK 840D
Description of Functions **C-PLC Programming** (03.96 Edition)
Order number: 6FC5 297-3AB60-0BP0

/FBR/
SINUMERIK 840D/810D
IT Solutions
Description of Functions **Computer Link** (SINCOM) (09.01 Edition)
Order number: 6FC5 297-6AD60-0BP0

NFL Host Computer Interface
NPL PLC/NCK Interface

/FBSI/
SINUMERIK 840D/SIMODRIVE
Description of Functions **SINUMERIK Safety Integrated** (09.02 Edition)
Order number: 6FC5 297-6AB80-0BP1

/FBSP/
SINUMERIK 840D/810D
Description of Functions **ShopMill** (11.02 Edition)
Order number: 6FC5 297-6AD80-0BP1

/FBST/
SIMATIC
Description of Functions **FM STEPDRIVE/SIMOSTEP** (01.01 Edition)
Order number: 6SN1 197-0AA70-0YP4

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

/FBSY/ SINUMERIK 840D/810D
Description of Functions **Synchronized Actions** (10.02 Edition)
for Wood, Glass, Ceramics and Presses
Order number: 6FC5 297-6AD40-0BP2

/FBT/ SINUMERIK 840D/810D
Description of Functions **ShopTurn** (03.03 Edition)
Order number: 6FC5 297-6AD70-0BP2

/FBTC/ SINUMERIK 840D/810D
IT Solutions
SINUMERIK Tool Data Communication SinTDC (01.02 Edition)
Description of Function
Bestellnummer: 6FC5 297-5AF30-0BP0

/FBTD/ SINUMERIK 840D/810D
IT Solutions
Tool Information System (SinTDI) with Online Help (03.01 Edition)
Description of Functions
Order number: 6FC5 297-6AE00-0BP0

/FBU/ **SIMODRIVE 611 universal / universal E**
Closed-Loop Control Component for Speed Control and Positioning
Description of Functions (02.02 Edition)
Order number: 6SN1 197-0AB20-0BP5

/FBW/ SINUMERIK 840D/840Di/810D
Description of Functions **Tool Management** (10.02 Edition)
Order number: 6FC5 297-6AC60-0BP1

/FBWI/ SINUMERIK 840D/840Di/810D
Description of Functions **WinTPM** (02.02 Edition)
Order number: This document is part of the software

/HBA/ SINUMERIK 840D/840Di/810D
Manual @Event (03.02 Edition)
Order number: 6AU1900-0CL20-0AA0

/HBI/ SINUMERIK 840Di
Manual (09.02 Edition)
Order number: 6FC5 297-6AE60-0BP1

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

/INC/ SINUMERIK 840D840Di//810D
Commissioning Tool **SINUMERIK SinuCOM NC** (02.02 Edition)
System Description
Order number: (an integral part of the online Help for the start-up tool)

/PAP/ SIMODRIVE **Sensor**
Absolute Encoder with PROFIBUS DP
User's Guide (02.99 Edition)
Order number: 6SN1 197-0AB10-0YP1

/PFK/ **SIMODRIVE**
Planning Guide **1FT5/1FT6/1FK6 Motors** (12.01 Edition)
AC Servo Motors for Feedrate and
Main Spindle Drives
Order number: 6SN1 197-0AC20-0BP0

/PJE/ SINUMERIK 840D/810D
HMI Embedded Configuring Package (08.01 Edition)
Description of Functions: Software Update, Configuration, Installation
Order number: 6FC5 297-6EA10-0BP0
(the document PS Configuring Syntax is supplied with the software and available as a pdf file)

/PJFE/ **SIMODRIVE**
Planning Guide **Built-In Synchronous Motors 1FE1**
Three-Phase AC Motors for Main Spindle Drives (09.01 Edition)
Order number: 6SN1 197-0AC00-0BP1

/PJLM/ **SIMODRIVE**
Planning Guide **Linear Motors 1FN1, 1FN3** (11.01 Edition)
ALL General Information about Linear Motors
1FN1 1FN1 Three-Phase Linear Motor
1FN3 1FN3 Three-Phase Linear Motor
CON Connections
Order number: 6SN1 197-0AB70-0BP2

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

/PJM/ **SIMODRIVE**
Planning Guide **Motors** (11.00 Edition)
Three-Phase AC Motors for Feed and Main Spindle Drives
Order number: 6SN1 197-0AA20-0BP5

/PJU/ **SIMODRIVE 611**
Planning Guide **Inverters** (08.02 Edition)
Order number: 6SN1 197-0AA00-0BP6

/PMS/ **SIMODRIVE** (04.02 Edition)
Planning Guide **ECO Motor Spindle** for Main Spindle Drives
Order number: 6SN1 197-0AD04-0BP0

/POS1/ **SIMODRIVE POSMO A**
User's Guide (08.02 Edition)
Distributed Positioning Motor on PROFIBUS DP
Order number: 6SN2 197-0AA00-0BP3

/POS2/ **SIMODRIVE POSMO A**
Installation Instructions (enclosed with POSMO A)

/POS3/ **SIMODRIVE POSMO SI/CD/CA** (08.02 Edition)
Operator's Guide
Distributed Servo Drive Systems
Order number: 6SN2 197-0AA20-0BP3

/PPH/ **SIMODRIVE**
Planning Guide **1PH2/1PH4/1PH7 Motors** (12.01 Edition)
AC Induction Motors for Main Spindle Drives
Order number: 6SN1 197-0AC60-0BP0

/PPM/ **SIMODRIVE**
Planning Guide **Hollow-Shaft Motors** (10.01 Edition)
Hollow-Shaft Motors for Main Spindle Drives
1PM4 and 1PM6
Order number: 6SN1 197-0AD03-0BP0

/S7H/ **SIMATIC S7-300**
- Reference Manual: CPU Data (HW Description) (2002 Edition)
- Reference Manual: Module Data
- Technological Functions Manual
- Installation Manual
Order number: 6ES7 398-8FA10-8AA0

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

/S7HT/ **SIMATIC S7-300**
Manual STEP 7, Fundamentals, V. 3.1 (03.97 Edition)
Order number: 6ES7 810-4CA02-8AA0

/S7HR/ **SIMATIC S7-300**
Manual (03.97 Edition)
STEP7, Reference Manuals, V3.1
Order number: 6ES7 810-4CA02-8AR0

/S7S/ **SIMATIC S7-300**
FM 353 Stepper Drive Positioning Module (04.97 Edition)
Order in conjunction with configuring package

/S7L/ **SIMATIC S7-300**
FM 354 Positioning Module for Servo Drive (04.97 Edition)
Order together with configuring package

/S7M/ **SIMATIC S7-300**
FM 357.2 Multimodule for Servo and Stepper Drives (01.01 Edition)
Order together with configuring package

/SP/ **SIMODRIVE 611-A/611-D,**
SimoPro 3.1
Program for Configuring Machine-Tool Drives
Order number: 6SC6 111-6PC00-0AA❑
Order from: WK Fürth

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

d) Installation and Start-Up

/IAA/ **SIMODRIVE 611A**
Installation and Start-Up Guide (10.00 Edition)
Order number: 6SN 1197-0AA60-0BP6

/IAC/ SINUMERIK 810D
Installation and Start-Up Guide (03.02 Edition)
(including description of SIMODRIVE 611D start-up software)
Order number: 6FC5 297-6AD20-0BP0

/IAD/ SINUMERIK 840D/SIMODRIVE 611D
Installation and Start-Up Guide (11.02 Edition)
(including description of SIMODRIVE 611D start-up software)
Order number: 6FC5 297-6AB10-0BP2

/IAM/ SINUMERIK 840D/840Di/810D
HMI/MMC Installation and Start-Up Guide (11.02 Edition)
Order number: 6FC5 297-6AE20-0BP2

AE1	Updates/Options
BE1	Expand the operator interface
HE1	Online Help
IM2	Start-Up HMI Embedded
IM4	Start-Up HMI Advanced
TX1	Setting Foreign Language Texts

■

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

D Index

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

V

W

Z

840D NCU 571 | 840D NCU 572 NCU 573 | 810D

E Commands, identifiers

840D
NCU 571

840D
NCU 572
NCU 573

810D

840Di

840D NCU 571 | 840D NCU 572 NCU 573 | 810D

840D NCU 571 | 840D NCU 572 NCU 573 | 810D | 840Di

840D NCU 571 | 840D NCU 572 NCU 573 | 810D

W

X

Y

Z

To SIEMENS AG A&D MC BMS P.O. Box 3180 D-91050 Erlangen, Germany Phone: ++49-(0)180-5050-222 [Hotline] Fax: ++49-(0)9131-98-2176 [Documentation] Email: motioncontrol.docu@erlf.siemens.de	**Suggestions** **Corrections** For publication: SINUMERIK 840D/840Di/810D Fundamentals User Documentation
From Name Company/Dept. Address: Zip Code: Town: Phone: / Fax: /	Programming Guide Order No.: 6FC5298-6AB00-0BP2 Edition: 11.02 Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.

Suggestions and/or corrections

Overview of SINUMERIK 840D/840Di/810D Documentation (11.2002)

General Documentation

SINUMERIK
840D/810D
Brochure

SINUMERIK
840D/840Di/
810D/
Catalog
Ordering Info.
NC 60 ***)**

SINUMERIK
SIROTEC
SIMODRIVE
Accessories
Catalog
Accessories
NC-Z

User Documentation

SINUMERIK
840D/810D/
FM-NC
AutoTurn
– Short Guide
– Programming/
Setup

SINUMERIK
840D/840Di/
810D
Operator's Guide
– HT 6

SINUMERIK
840D/840Di/
810D
Diagnostics
Guide *)

SINUMERIK
840D/840Di/
810D
Operator's Guide *)
– Short Guide
– HMI Embedded
– HMI Advanced

User Documentation

SINUMERIK
840D/840Di/
810D
Program. Guide
– Short Guide
– Fundamentals ***)**
– Advanced ***)**
– Cycles
– Measuring Cycles
– ISO Turning/Milling

SINUMERIK
840D/810D
Operator's Guide
– **ManualTurn**
– Short Guide ManualTurn
– **ShopMill**
– Short Guide ShopMill
– **ShopTurn**
– Short Guide ShopTurn

SINUMERIK
840Di
System Overview

Manufacturer/Service Documentation

SINUMERIK
Configuring
(HW) ***)**
– 810D
– 840D

SINUMERIK
840D/840Di/
810D
Operator
Components
(HW) ***)**

SINUMERIK
840D/810D
Description of
Functions
– ManualTurn
– ShopMill
– ShopTurn

SINUMERIK
840D/840Di/
810D
Description of
Functions
Synchronized
Actions

Manufacturer/Service Documentation

SINUMERIK
SIMODRIVE
611D
840D/810D
Description of
Functions
Drive Functions *)

SINUMERIK
840D/840Di/
810D
Description of
Functions
– Basic Machine ***)**
– Extended Functions
– Special Functions

SINUMERIK
840D/840Di/
810D
Description of
Functions
Tool Manage-
ment

SINUMERIK
840D/810D
Configuring Kit
HMI Embedded

SINUMERIK
840D/810D
Description of
Functions
Operator Interface
OP 030

SINUMERIK
840D/810D
IT Solutions
– Computer Link
– Tool Data Information System
– NC Data Management
– NC Data Transfer
– Tool Data Communication

Manufacturer/Service Documentation

SINUMERIK
SIMODRIVE
Description of
Functions
SINUMERIK
Safety Integrated

SINUMERIK
840D
Description of
Functions
Digitizing

SINUMERIK
SIMODRIVE
Installation &
Start-Up Guide *)
– 810D
– 840D/611D
– HMI

SINUMERIK
SIMODRIVE
840D/840Di
810D
611D
Lists *)

SINUMERIK
SIMODRIVE
840D
611D
Description of
Functions
Linear Motor

SINUMERIK
SIMODRIVE
840D
611D
Description of Functions
– Hydraulics Module
– Analog Module

SINUMERIK
SIMODRIVE
SIROTEC
EMC
Guidelines

Electronic Documentation

SINUMERIK
SIMODRIVE
840D/840Di/
810D
611, Motors
DOC ON CD *)
The SINUMERIK System

Manufacturer/Service Documentation

SINUMERIK
840D/840Di/
810D
Description of
Functions
ISO Dialects for
SINUMERIK

SINUMERIK
840Di
Manual
(HW + Installation
and Start-Up)

SINUMERIK
840D/810D
Description of
Functions
Remote Diagnosis

SINUMERIK
840D/840Di/
810D
Manual
@ Event

***)** These documents are a minimum requirement

Siemens AG
Automation & Drives
Motion Control Systems
P.O. Box 3180, D-91050 Erlangen
Germany

www.ad.siemens.de

Order No.: 6FC5298-6AB00-0BP2

Printed in Germany