SIEMENS

SINUMERIK

SINUMERIK 802D sl Turning

Programming and Operating Manual

Valid for

Control	Software version SINUMERIK
802D sl T/M	1.4

06/2009
6FC5398-1CP10-5BA0

Legal information

Warning notice system

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

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

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

⚠CAUTION
with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION
without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE
indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Proper use of Siemens products

Note the following:

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

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Ordernumber: 6FC5398-1CP10-5BA0
℗ 07/2009

Foreword

Structure of the documentation

The SINUMERIK documentation is organized in 3 parts:

- General documentation
- User documentation
- Manufacturer/service documentation

Information on the following topics is available at
http://www.siemens.com/motioncontrol/docu:

- Ordering documentation
  Here you can find an up-to-date overview of publications.
- Downloading documentation
  Links to more information for downloading files from Service & Support.
- Researching documentation online
  Information on DOConCD and direct access to the publications in DOConWEB.
- Compiling individual documentation on the basis of Siemens contents with the My Documentation Manager (MDM), refer to http://www.siemens.com/mdm

  My Documentation Manager provides you with a range of features for generating your own machine documentation.
- Training and FAQs
  Information on the range of training courses and FAQs (frequently asked questions) are available via the page navigation.

Target group

This publication is intended for programmers, planning engineers, machine operators and system operators.

Benefits

With the Programming and Operating Manual, the target group can develop, write, test and debug programs and software user interfaces.

In addition, it enables the target group to operate the hardware and software of a machine.

Standard scope

This documentation only describes the functionality of the standard version. Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

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.

For the sake of simplicity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation, or maintenance.

Technical support

If you have any technical questions, please contact our hotline:

	Europe / Africa
Phone	+49 180 5050 222
Fax	+49 180 5050 223
€ 0.14/min. from German landlines, mobile phone prices may differ.	
Internet	http://www.siemens.com/automation/support-request

	America
Phone	+1 423 262 2522
Fax	+1 423 262 2200
E-mail	mailto:techsupport.sea@siemens.com

	Asia/Pacific
Phone	+86 1064 757575
Fax	+86 1064 747474
E-mail	mailto:support.asia.automation@siemens.com

Note

Country telephone numbers for technical support are provided under the following Internet address:

http://www.automation.siemens.com/partner

Questions regarding documentation

If you have any queries (suggestions, corrections) in relation to this documentation, please fax or e-mail us:

Fax	+49 9131 98 2176
E-mail	mailto:docu.motioncontrol@siemens.com

A fax form is available in the appendix of this document.

SINUMERIK Internet address

http://www.siemens.com/sinumerik

EC Declaration of Conformity

The EC Declaration of Conformity for the EMC Directive can be found/obtained

- on the internet:
  http://support.automation.siemens.com
  under the product/order No. 15263595
- at the relevant regional office of the I DT MC Business Unit of Siemens AG.

Table of contents

Description 1

1.1 Control and display elements

Operator control elements

The defined functions are called up via the horizontal and vertical softkeys. For a description, please refer to this manual:

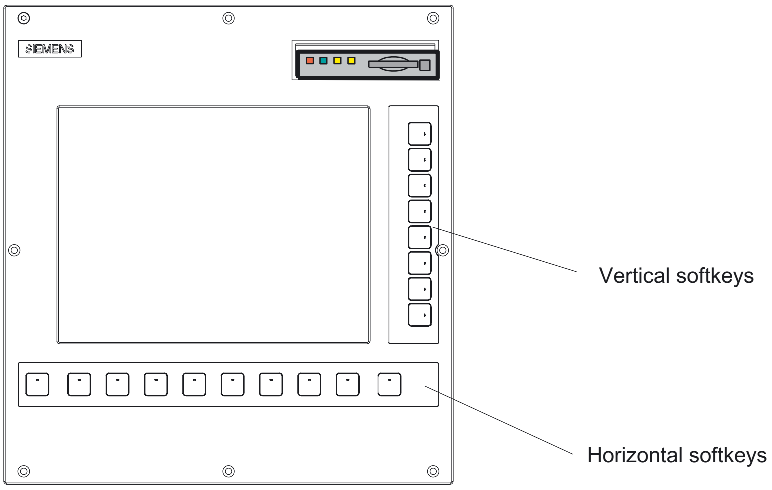
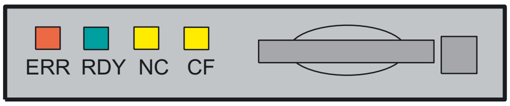


Figure 1-1 CNC operator panel

1.2 Error and status displays

LED displays on the CNC operator panel (PCU)

The following LEDs are installed on the CNC operator panel.



The individual LEDs and their functions are described in the table below.

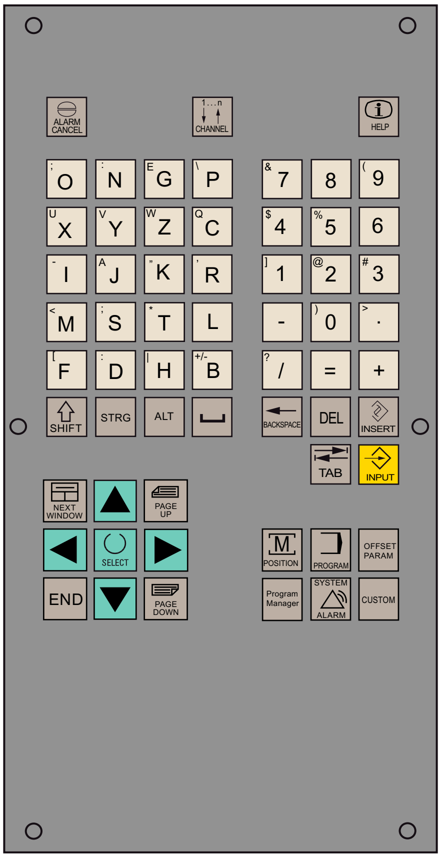
Table 1- 1 Status and error displays

LED	Significance
ERR (red)	Serious error, remedy through power OFF/ON
RDY (green)	Ready for operation
NC (yellow)	Signoflife monitoring
CF (yellow)	Reading from/writing to CF card

References

You can find information on error description in the SINUMERIK 802D sl Diagnostics Manual

1.3 Key definition of the full CNC keyboard (vertical format)



>	ETC key	Λ	Recall key
ALARM CANCEL	Acknowledge alarm key		
1...n CHANNEL	No function		
HELP	Info key	END	
SHIFT	Shift key	SELECT	Selection key / toggle key
CTRL	Control key	⌴	Space
ALT	ALT key	BACKSPACE	Delete key (backspace)

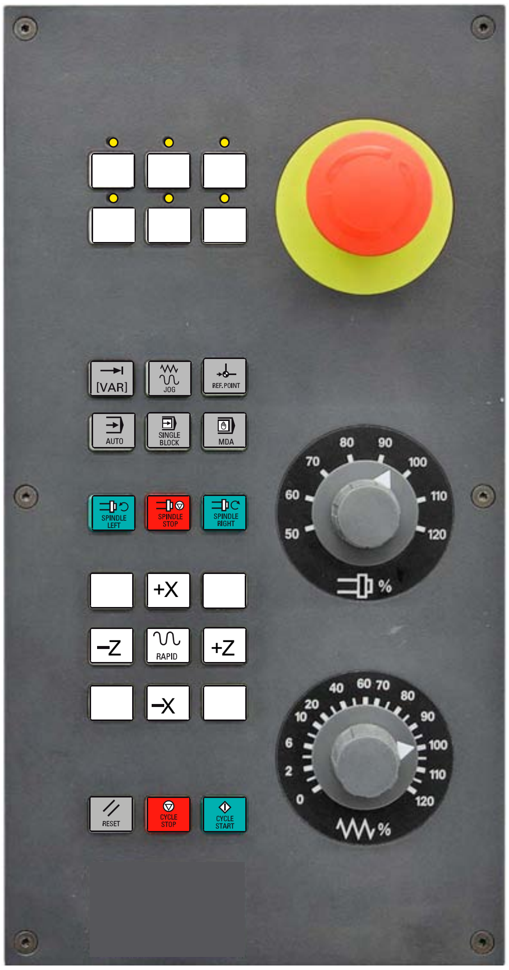
DEL	Delete key
INSERT	Insert key
TAB	Tabulator
INPUT	ENTER / Input key
POSITION	POSITION operating area key (Position operating area)
PROGRAM	PROGRAM operating area key (Program operating area)
OFFSET PARAM	OFFSET PARAM operating area key (Parameter operating area)
PROGRAM MANAGER	PROGRAM MANAGER operating area key (Program Manager operating area)
SYSTEM ALARM	SYSTEM/ALARM operating area key (System/Alarm operating area)
CUSTOM	CUSTOM operating area (User operating area)
NEXT WINDOW	not assigned
PAGE UP PAGE DOWN	Scroll keys
▲ ◀ ▶ ▼	Cursor keys
A J W Z	Alphanumeric keys Double assignment at the Shift level
(9) 0	Numeric keys Double assignment at the Shift level

Hot keys

In the part program editor and in the input fields of the HMI, the following functions can be carried out with certain key combinations on the full CNC keyboard:

Keystroke combination	Function
<CTRL> and <C>	Copy selected text
<CTRL> and <B>	Select text
<CTRL> and <X>	Cut selected text
<CTRL> and <V>	Paste copied text
<ALT> and <L>	Changeover to small letters
<ALT> and <H> or <HELP> key	Call help system
<ALT> and <S>	Switch-in and switch-out the Editor for Asian characters

1.4 Key definition of the machine control panel



RESET	RESET
CYCLE STOP	CYCLE STOP (NC STOP)
CYCLE START	CYCLE START (NC START)
●	EMERGENCY STOP
○ %	Spindle Speed Override Spindle override

○ □	User-defined key with LED
□	User-defined key without LED
→| [VAR]	INCREMENT Increment
JOG	JOG
REF. POINT	REFERENCE POINT Reference point
AUTO	AUTOMATIC
SINGLE BLOCK	SINGLE BLOCK Single block
MDA	MANUAL DATA Manual input
SPINDLE LEFT	SPINDLE START CCW Counterclockwise
SPINDLE STOP	SPINDLE STOP
SPINDLE RIGHT	SPINDLE START CW Clockwise
RAPID	RAPID TRAVERSE OVERLAY Rapid traverse override
+X -X	X axis
+Z -Z	Z axis
○ %	Feedrate override Feedrate control

Note

This documentation assumes an 802D standard machine control panel (MCP). Should you use a different MCP, the operation may be other than described herein.

1.5 Coordinate systems

As a rule, a coordinate system is formed from three mutually perpendicular coordinate axes. The positive directions of the coordinate axes are defined using the so-called "3-finger rule" of the right hand. The coordinate system is related to the workpiece and programming takes place independently of whether the tool or the workpiece is being traversed. When programming, it is always assumed that the tool traverses relative to the coordinate system of the workpiece, which is intended to be stationary.

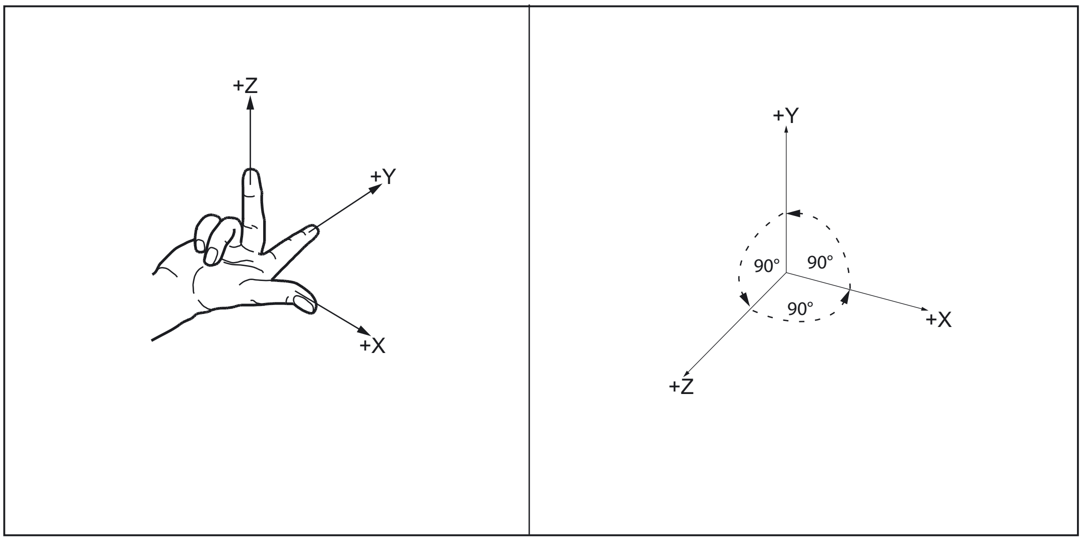


Figure 1-2 Determining the axis directions to one another; coordinate system for programming

Machine coordinate system (MCS)

The orientation of the coordinate system relative to the machine depends on the respective machine type. It can be rotated in different positions.

The directions of the axes follow the "3-finger rule" of the right hand. Seen from in front of the machine, the middle finger of the right hand points in the opposite direction to the infeed of the main spindle.

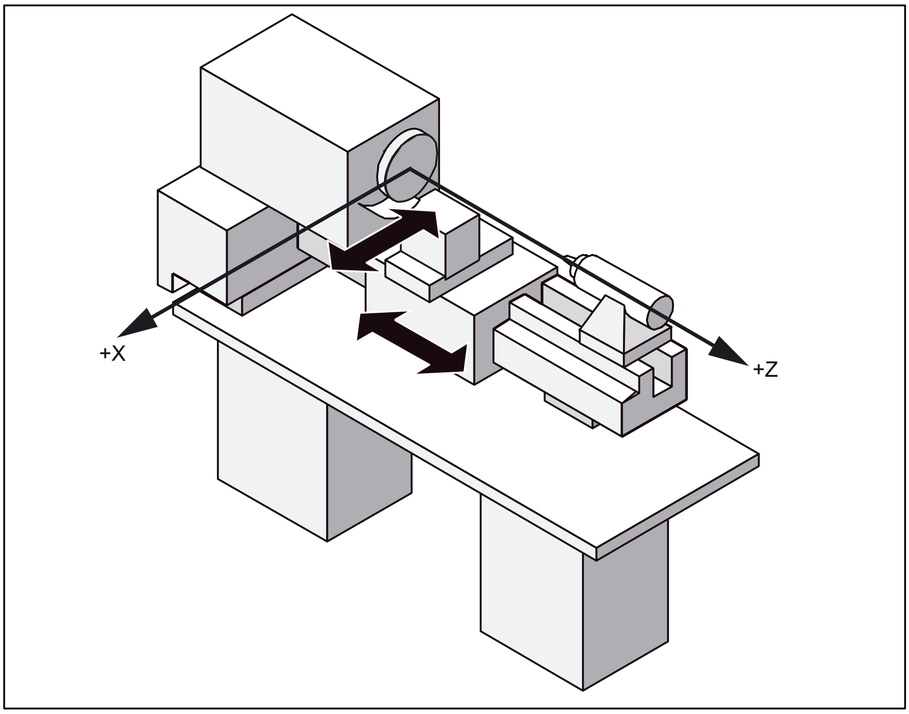


Figure 1-3 Machine coordinate axes using the example of a turning machine

The origin of this coordinate system is the **machine zero.**
This point is only a reference point which is defined by the machine manufacturer. It does not have to be approachable.

The traversing range of the **machine axes** can by in the negative range.

Workpiece coordinate system (WCS)

To describe the geometry of a workpiece in the workpiece program, a right-handed, right-angled coordinate system is also used.
The **workpiece zero** can be freely selected by the programmer in the Z axis. In the X axis, it lies in the turning center.

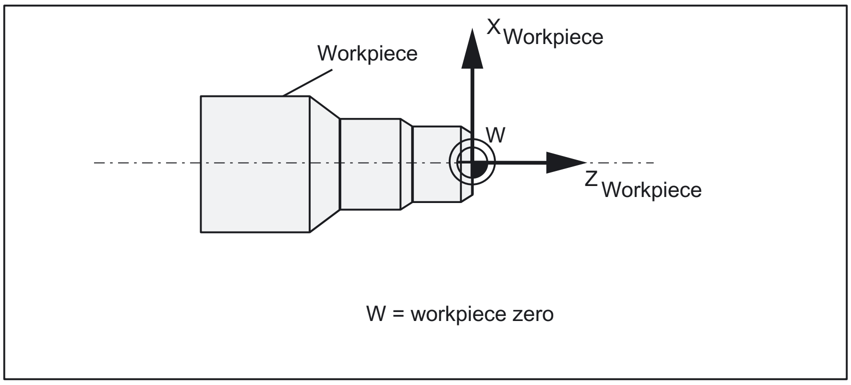


Figure 1-4 Workpiece Coordinate System

Relative coordinate system (REL)

In addition to the machine and workpiece coordinate systems, the control system provides a relative coordinate system. This coordinate system is used for setting reference points that can be freely selected and have no influence on the active workpiece coordinate system. All axis movements are displayed relative to these reference points.

Note

You can activate and display the actual value of the relevant coordinate system by pressing the vertical softkey "MCS/WCS REL" in the Position operating area.

Clamping the workpiece

For machining, the workpiece is clamped on the machine. The workpiece must be aligned such that the axes of the workpiece coordinate system run in parallel with those of the machine. Any resulting offset of the machine zero with reference to the workpiece zero is determined along the Z axis and entered in a data area intended for the **settable work offset**. In the NC program, this offset is activated during program execution, e.g. using a programmed **G54**.

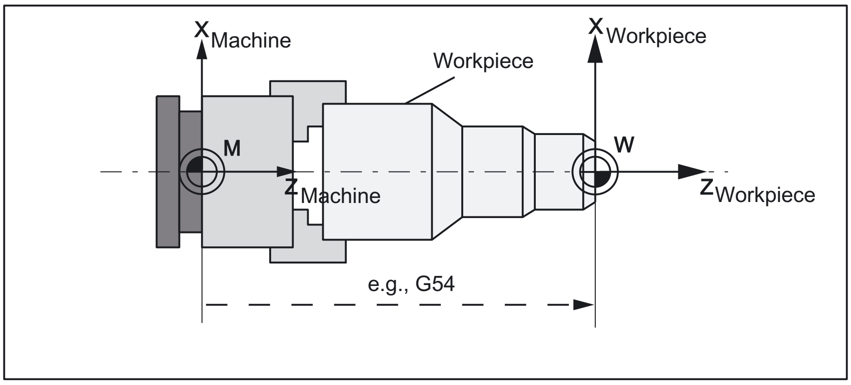


Figure 1-5 Workpiece on the machine

Current workpiece coordinate system

The programmed work offset TRANS can be used to generate an offset with reference to the workpiece coordinate system resulting in the current workpiece coordinate system resulting in the current workpiece coordinate system (see section "Programmable work offset: TRANS").

Software interface 2

2.1 Screen layout

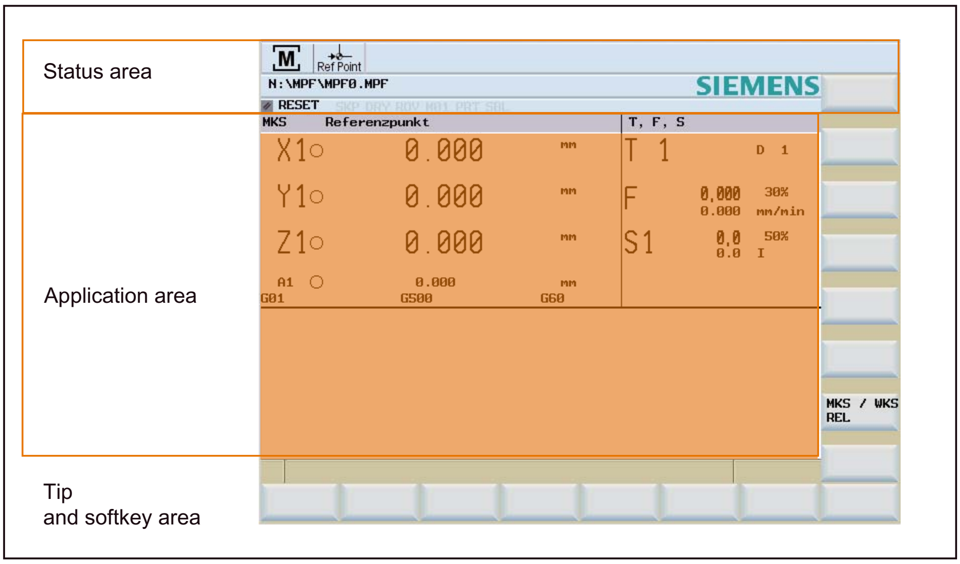


Figure 2-1 Screen layout

The screen is divided into the following main areas:

- Status area
- Application area
- Note and softkey area

Status area

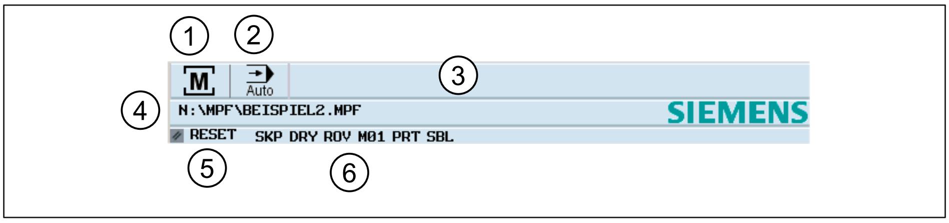


Figure 2-2 Status area

Table 2- 1 Explanation of the screen controls in the status area

Numbering	Display	Icon	Significance
①	Active operating area	M	Position (operating area key <POSITION>)
		🔧	System (operating area key <SYSTEM>)
			Program (operating area key <PROGRAM>)
		NC	Program Manager (operating area key <PROGRAM MANAGER>)
			Parameter (operating area key <OFFSET PARAM>)
		△	Alarm (operating area key <ALARM>)
②	Active mode	Ref Point	Approaching a reference point
		Jog	JOG
		[100]	JOG INC; 1 INC, 10 INC, 100 INC, 1000 INC, VAR INC (incremental evaluation in the JOG mode)
		MDA	MDA

Numbering	Display	Icon	Significance
		Auto	AUTOMATIC
③	Alarm and message line		In addition, the following is displayed: 1. Alarm number with alarm text, or 2. Message text
④	Selected part program (main program)		
⑤	Program state	RESET	Program canceled / default state
		RUN	Program is running
		STOP	Program stopped
⑥	Program control in automatic mode		

Note and softkey area

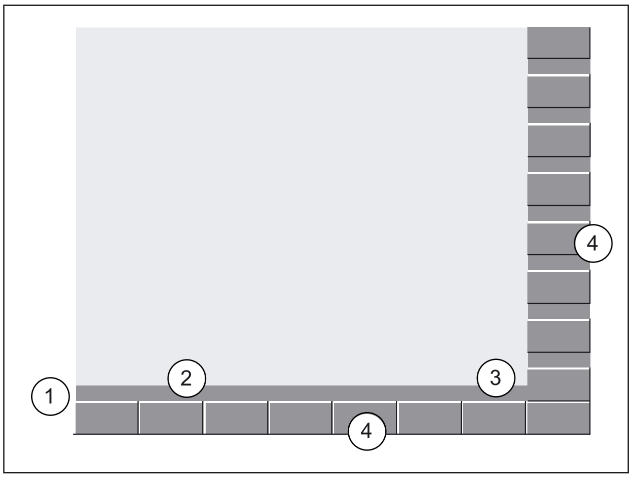


Figure 2-3 Note and softkey area

Table 2- 2 Explanation of the screen controls in the note and softkey area

Screen item	Display	Significance
①	⇧	RECALL symbol Pressing the <RECALL> key lets you return to the higher menu level.
②		Information line Displays notes and information for the operator and fault states

Screen item	Display	Significance
③		HMI status information
	➡	ETC is possible (pressing this key displays the horizontal softkey bar providing further functions.)
	“Ł“	Mixed notation active (uppercase/lowercase letters)
	🔌	RS232 connection active
	💻	Connection to commissioning and diagnostic tools (e.g. Programming Tool 802) active
	RCS	RCS network connection active
④		Softkey bar vertical and horizontal

Display of the softkeys in the document

To make the softkeys easier to locate, the horizontal and vertical softkeys are displayed in different basic colors.

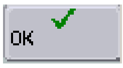
□	Horizontal softkey
░	Vertical softkey

2.2 Standard softkeys

Use this softkey to close the screen.

Use this softkey to cancel the input; the window is closed.

Selecting this softkey will complete your input and start the calculation.



Selecting this softkey will complete your input and accept the values you have entered.

2.3 Operating areas

The functions of the control system can be carried out in the following operating areas:

M MACHINE	POSITION	Machine operation
OFFSET PARAM	OFFSET PARAM	Entering the compensation values and setting data
PROGRAM	PROGRAM	Creation of part programs
PROGRAM MANAGER	PROGRAM MANAGER	Part program directory
SHIFT + SYSTEM ALARM	SYSTEM	Diagnostics, commissioning
SYSTEM ALARM	ALARM	Alarm and message lists
CUSTOM	CUSTOM	Users can call their own application

To change to another operating area, press the relevant key on the CNC full keyboard (hard key).

Protection levels

The SINUMERIK 802D sl provides a concept of protection levels for enabling data areas. The control system is delivered with default passwords for the protection levels 1 to 3.

Protection level 1	Experts password
Protection level 2	Manufacturer password
Protection level 3	User password

These control the various access rigths.

In the menus listed below the input and modification of data depends on the protection level set:

- Tool offsets
- Work offsets
- Setting data
- RS232 settings
- Program creation / program correction

2.4 The help system

Comprehensive online help is stored in the control system. Some help topics are:

- Product brief of all important operating functions
- Overview and product brief of the NC commands
- Explanation of the drive parameters
- Explanation of the drive alarms

Operating sequence

HELP

You can call the help system from any operating area either by pressing the Info key or by using the key combination <ALT+H>.

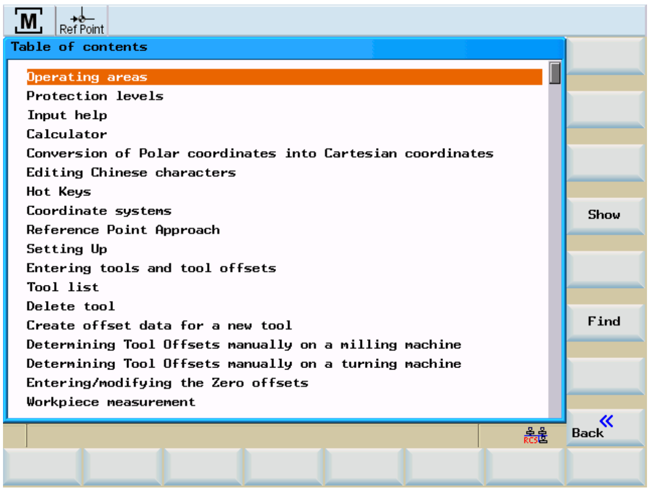


Table of contents

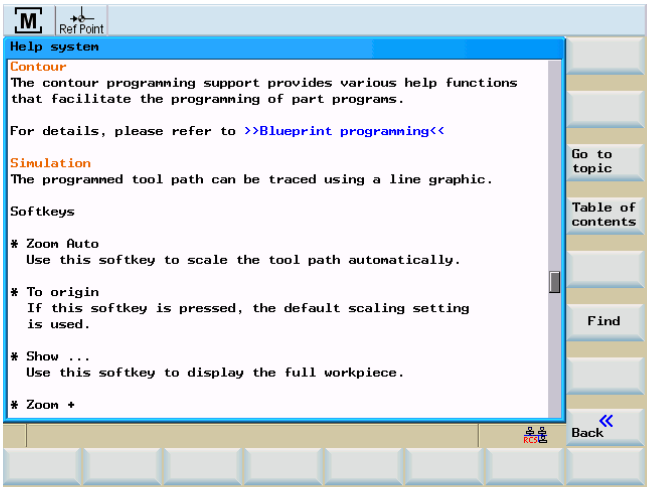
Operating areas
Protection levels
Input help
Calculator
Conversion of Polar coordinates into Cartesian coordinates
Editing Chinese characters
Hot Keys
Coordinate systems
Reference Point Approach
Setting Up
Entering tools and tool offsets
Tool list
Delete tool
Create offset data for a new tool
Determining Tool Offsets manually on a milling machine
Determining Tool Offsets manually on a turning machine
Entering/modifying the Zero offsets
Workpiece measurement

Figure 2-4 Help system: Table of contents

Softkeys

Show

This function opens the selected topic.



Help system

Contour
The contour programming support provides various help functions that facilitate the programming of part programs.

For details, please refer to >>Blueprint programming<<

Simulation
The programmed tool path can be traced using a line graphic.

Softkeys

```
* Zoom Auto
  Use this softkey to scale the tool path automatically.

* To origin
  If this softkey is pressed, the default scaling setting
  is used.

* Show ...
  Use this softkey to display the full workpiece.

* Zoom +
```

Figure 2-5 Help system: Description of the topic

Go to Topic

Use this function to select cross references. A cross reference is marked by the characters ">>....<<". This softkey is only displayed if a cross reference is displayed in the application area.

Back to Topic

If you select a cross-reference, the "Back to topic" softkey will also be displayed. Select this function to go back to the previous screen.

Find

Use this function to search for a term in the table of contents. Type the term you are looking for and start the search process.

Help in the "Program editor" area

The help system offers an explanation for each NC operation. To display the infotext directly, position the cursor after the appropriate operation and press the Info key. The NC instruction must be written using uppercase letters.

Turning On, Reference Point Approach

3

3.1 Switching on, reference point approach

Note

When turning on the SINUMERIK 802D sl and the machine, please also observe the machine documentation, since turning on and reference point approach are machine-dependent functions.

Operating sequence

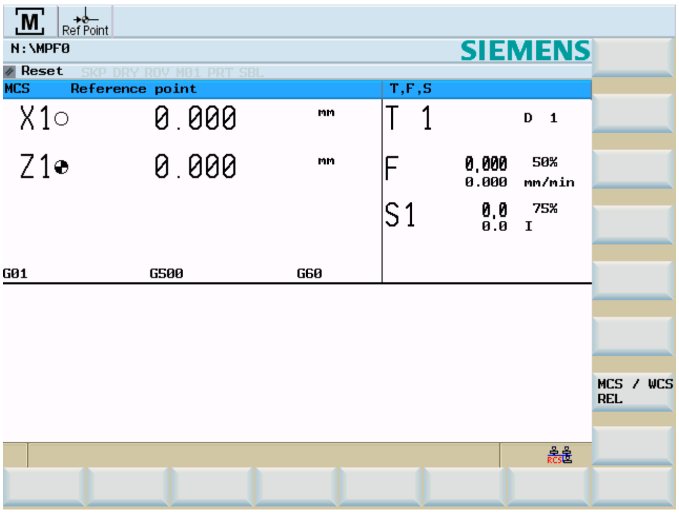
First, switch on the power supply for the CNC and the machine.

M

After the control system has booted, you are in the "Position" operating area, in the "Reference point approach" mode.

Ref Point

The 'Reference point' window is active.



N:\MPF0

Reset SKP DRY ROV M01 PRT SBL

MCS Reference point

X1 ○ 0.000 mm

Z1 ◕ 0.000 mm

G01 G500 G60

T,F,S

T 1 D 1

F 0.000 50% 0.000 mm/min

S1 0.0 75% 0.0 I

Figure 3-1 Reference-point approach start screen

The "Reference point" window displays whether the axes are referenced.

○ Axis must be referenced

◕ Axis is referenced/synchronized

+X

Press the arrow keys.

-Z

If you select the wrong approach direction, no motion is carried out.

One after the other, move each axis to the reference point.
You can exit the function by selecting another operating mode (MDA, AUTOMATIC or JOG).

JOG

To access the functions described below, you need to select <JOG> mode.

4 Set up

Preliminary remarks

Before you can work with the CNC, set up the machine, the tools, etc. as follows:

- Enter the tools and the tool offsets.
- Enter/modify the work offset
- Enter the setting data.

4.1 Entering tools and tool offsets

4.1.1 Entering tools and tool offsets

Functionality

The tool offsets consist of several data describing the geometry, the wear and the tool type. Each tool contains a defined number of cutting edge parameters dependent on the particular tool type. Tools are identified by a number (T number).

See also Chapter "Tool and tool compensation (Page 282)"

Operating sequences

OFFSET PARAM

Press the <OFFSET PARAM> key.

Tool list

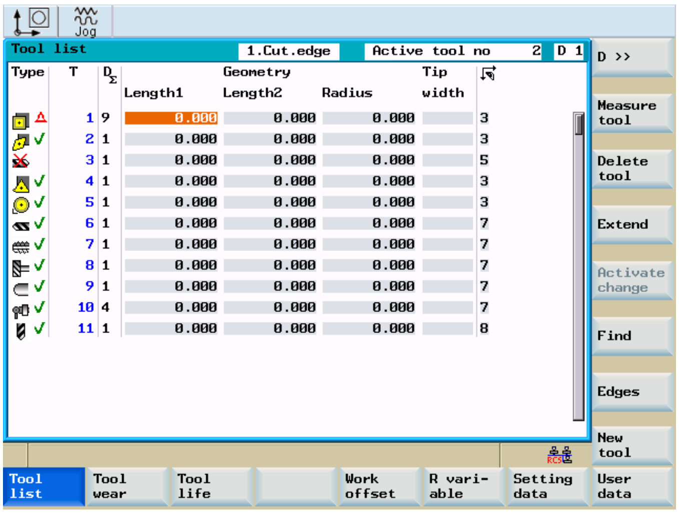
The function opens the "Tool list" window with the tool offset data. The window contains a list of the tools that have been created. Use the cursor keys and the Page Up / Page Down keys to navigate in this list.

Position the cursor bar on the input field to be modified and enter the value(s).

INPUT

Confirm with <Input> or by moving the cursor.

Standard tool list



Type	T	D_{Σ}	Geometry Length1	Length2	Radius	Tip width	
△	1	9	0.000	0.000	0.000		3
✓	2	1	0.000	0.000	0.000		3
	3	1	0.000	0.000	0.000		5
✓	4	1	0.000	0.000	0.000		3
✓	5	1	0.000	0.000	0.000		3
✓	6	1	0.000	0.000	0.000		7
✓	7	1	0.000	0.000	0.000		7
✓	8	1	0.000	0.000	0.000		7
✓	9	1	0.000	0.000	0.000		7
✓	10	4	0.000	0.000	0.000		7
✓	11	1	0.000	0.000	0.000		8

Figure 4-1 Tool list

The tool nose radius compensation parameters of the T tools are shown in the tool list.

Contents of the tool list:

Table 4- 1 Tool list

Symbol/ Header	Content
Type	Cutting edge type of the tool and tool monitoring symbols (refer to the Chapter "Tool monitoring")
T	Tool number
D∑	Number of tool cutting edges
Geometry	Tool geometry
Cutting tip width	Tip width of the cutting edge
	Cutting edge position of the cutting edge

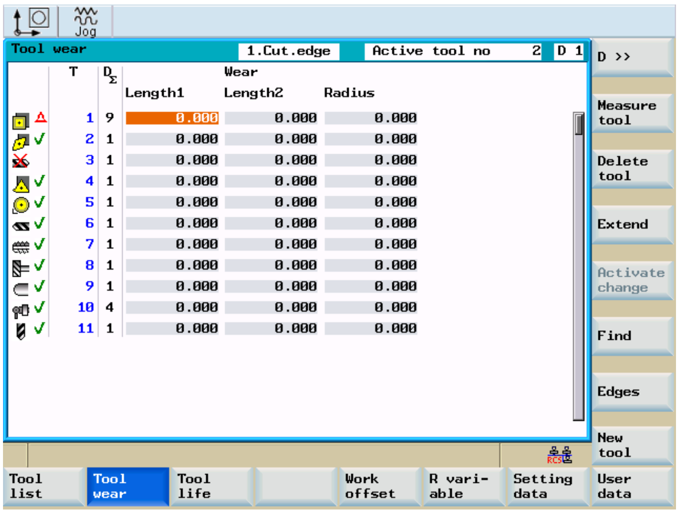
The following is displayed in the "tool list" line:

- The particular cutting edge number for all tools. Can be selected using softkey "D >>".
- The tool number and cutting edge number currently selected at the machine (e.g. 2, D 1)

Tool wear, standard

Tool wear

The function opens the "tool wear" window. The window contains a list of the tools that have been created and the wear data of the currently selected cutting edge. Use the cursor keys and the Page Up / Page Down keys to navigate in this list.



Tool wear   1.Cut.edge   Active tool no 2 D 1

	T	D∑	Wear Length1	Length2	Radius
⚠	1	9	0.000	0.000	0.000
✓	2	1	0.000	0.000	0.000
	3	1	0.000	0.000	0.000
✓	4	1	0.000	0.000	0.000
✓	5	1	0.000	0.000	0.000
✓	6	1	0.000	0.000	0.000
✓	7	1	0.000	0.000	0.000
✓	8	1	0.000	0.000	0.000
✓	9	1	0.000	0.000	0.000
✓	10	4	0.000	0.000	0.000
✓	11	1	0.000	0.000	0.000

Figure 4-2 Tool wear, standard

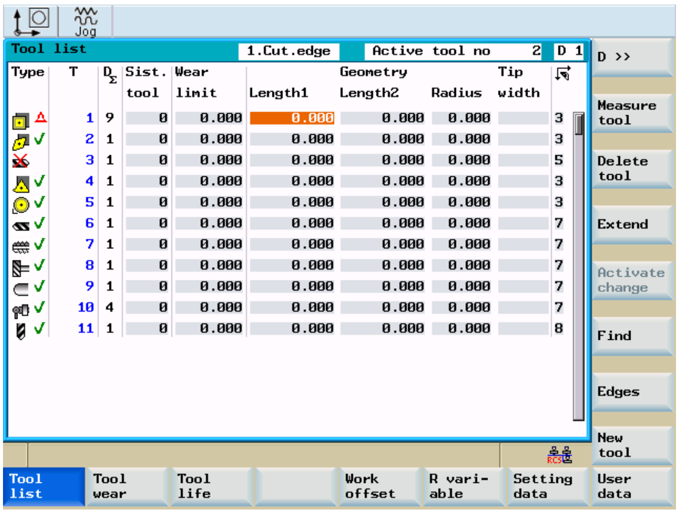
User-defined tool list

After having activated display MD394 DISPLAY_TOOL_LIST_SISTER_TOOL with "1", you may define the following additional cutting edge parameters for the tool:

- Sister tool
- Wear limit

Note

The input values from the user fields "Sister tool" and "Wear limit" from the "Tool list" tab are stored in the tool variables $TC_DP24 (wear limit) and $TC_DP25 (sister tool).



Tool list 1.Cut.edge Active tool no 2 D 1

Type	T	D_Σ	Sist.	Wear		Geometry		Tip	
			tool	limit	Length1	Length2	Radius	width	
△	1	9	0	0.000	0.000	0.000	0.000		3
✓	2	1	0	0.000	0.000	0.000	0.000		3
	3	1	0	0.000	0.000	0.000	0.000		5
✓	4	1	0	0.000	0.000	0.000	0.000		3
✓	5	1	0	0.000	0.000	0.000	0.000		3
✓	6	1	0	0.000	0.000	0.000	0.000		7
✓	7	1	0	0.000	0.000	0.000	0.000		7
✓	8	1	0	0.000	0.000	0.000	0.000		7
✓	9	1	0	0.000	0.000	0.000	0.000		7
✓	10	4	0	0.000	0.000	0.000	0.000		7
✓	11	1	0	0.000	0.000	0.000	0.000		8

Figure 4-3 User-defined tool list

Extended

For special tools, use the "extended" softkey function, which provides a complete cutting edge parameter list.

Softkeys

Tool measurement

Use this softkey to determine the tool offset data (only effective in the JOG mode!).

Measuring Manual

Use this softkey to determine the tool offset data manually.

Measuring Auto

Use this softkey to determine the tool offset data semi-automatically (only applies in conjunction with a probe).

Calibrate probe

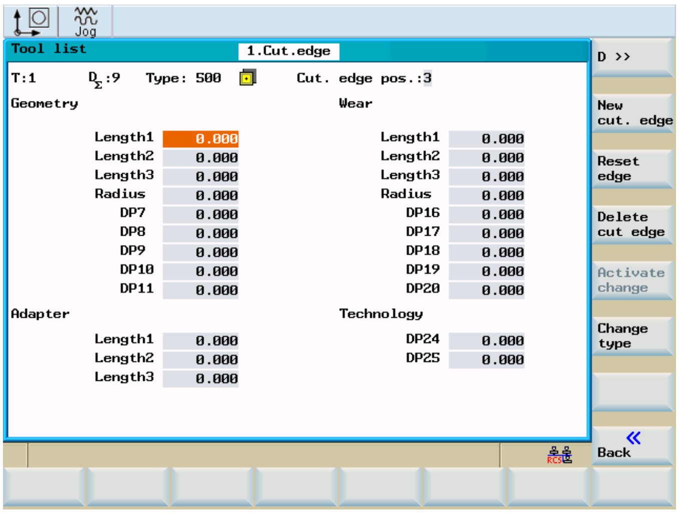
Calibrating the measuring probe.

Deleting a tool

The tool is deleted and removed from the tool list.

Extended

A complete list of the cutting edge parameters is displayed using the "Extended" function.



T:1 D_Σ:9 Type: 500 Cut. edge pos.:3

Geometry		Wear	
Length1	0.000	Length1	0.000
Length2	0.000	Length2	0.000
Length3	0.000	Length3	0.000
Radius	0.000	Radius	0.000
DP7	0.000	DP16	0.000
DP8	0.000	DP17	0.000
DP9	0.000	DP18	0.000
DP10	0.000	DP19	0.000
DP11	0.000	DP20	0.000
Adapter		**Technology**	
Length1	0.000	DP24	0.000
Length2	0.000	DP25	0.000
Length3	0.000		

Figure 4-4 Input screen for special tool

For the meanings of the cutting edge parameters, please refer to the Section "Programming".

Cutting edges

Opens a lower-level menu bar offering all functions required to create and display further edges.

D >>

Use this softkey to select the next higher edge number.

New cutting edge

Use this softkey to create a new edge.

Reset cutting edge

Use this softkey to reset all offset values of the edge to zero.

Delete cutting edge

Cutting edge is deleted.

Change active.

Modified values are activated.

Change type

This function is intended to change the tool type. Select the tool type using the appropriate softkey.

Search

Find tool number:

Type the number of the tool you are looking for and use the "OK" softkey to start the search. If the tool you are looking for exists, the cursor is positioned on the appropriate line.

New tool

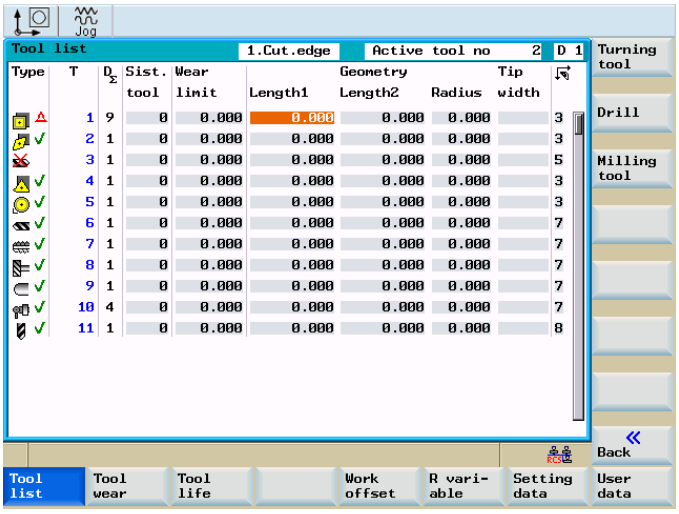
Use this softkey to create tool offset data for a new tool.

4.1.2 Create new tool

Operating sequence

New tool

This function offers another two softkey functions to select the tool type "Turning tool", "Drill" or "Milling tool". After selecting the tool type, enter the desired "Tool number" (3 digits max.) in the input field and select the "Cutting edge position" and the "Type".

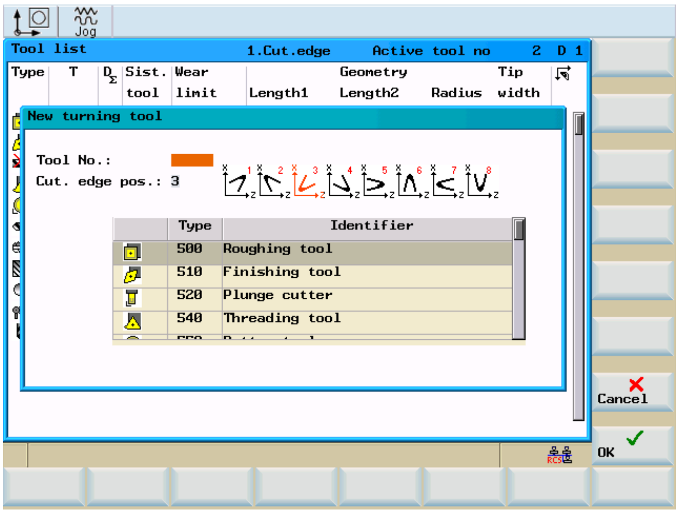


Jog

Tool list | 1.Cut.edge | Active tool no 2 | D 1

Type	T	D	Sist. tool	Wear limit	Geometry Length1	Length2	Radius	Tip width	
	1	9	0	0.000	0.000	0.000	0.000		3
✓	2	1	0	0.000	0.000	0.000	0.000		3
	3	1	0	0.000	0.000	0.000	0.000		5
✓	4	1	0	0.000	0.000	0.000	0.000		3
✓	5	1	0	0.000	0.000	0.000	0.000		3
✓	6	1	0	0.000	0.000	0.000	0.000		7
✓	7	1	0	0.000	0.000	0.000	0.000		7
✓	8	1	0	0.000	0.000	0.000	0.000		7
✓	9	1	0	0.000	0.000	0.000	0.000		7
✓	10	4	0	0.000	0.000	0.000	0.000		7
✓	11	1	0	0.000	0.000	0.000	0.000		8

Figure 4-5 "New tool" window



Jog

Tool list | 1.Cut.edge | Active tool no 2 | D 1

Type | T | D | Sist. tool | Wear limit | Geometry Length1 | Length2 | Radius | Tip width

New turning tool

Tool No.:
Cut. edge pos.: 3

Type	Identifier
500	Roughing tool
510	Finishing tool
520	Plunge cutter
540	Threading tool

Figure 4-6 Input of the tool number and cutting edge position of a turning tool

---

Note

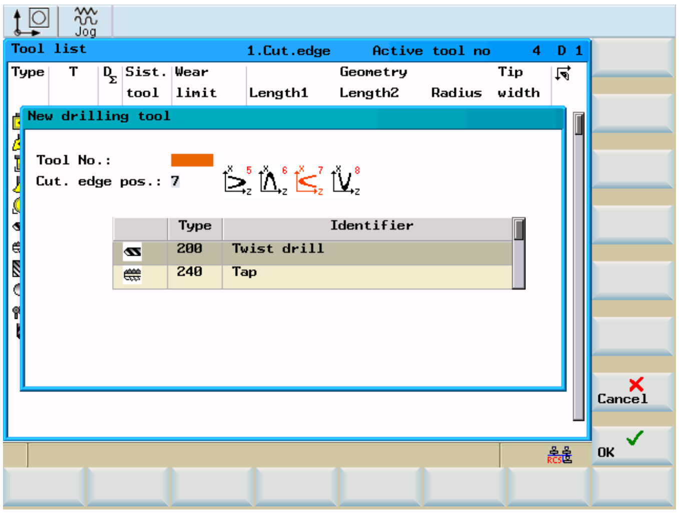
The coordinate system of tools for turning depends on the following display machine data:

MD290 CTM_POS_COORDINATE_SYSTEM

= 0 -> position of the tool after the turning center (refer to the diagram above)

= 2 -> position of the tool before the turning center

For drills and milling tools, the cutting edge position that corresponds to the machining direction is selected.



Tool list   1.Cut.edge   Active tool no 4 D 1

Type	T	D	Sist. tool	Wear limit	Length1	Geometry Length2	Radius	Tip width

New drilling tool

Tool No.:

Cut. edge pos.: 7

	Type	Identifier
	200	Twist drill
	240	Tap

Figure 4-7 Entering a tool number and cutting edge position for a drill

OK ✓

Select "OK" to confirm your input. A data record loaded with zero will be included in the tool list.

4.1.3 Determining the tool offsets (manually)

Note

Assignment of Length 1 or Length 2 to the axis is dependent on the tool type (turning tool, drill) (see the following figures).

For the turning tool, the reference point for the X axis is a diameter dimension!

Note

The axis coordinates used for the calculation refer to the machine coordinate system.

Functionality

Tool measurement

This function can be used to determine the unknown geometry of a tool T.

Using the actual position of the point F (machine coordinate) and the reference point, the control system can calculate the offset value assigned to length 1 or length 2 for the axis.

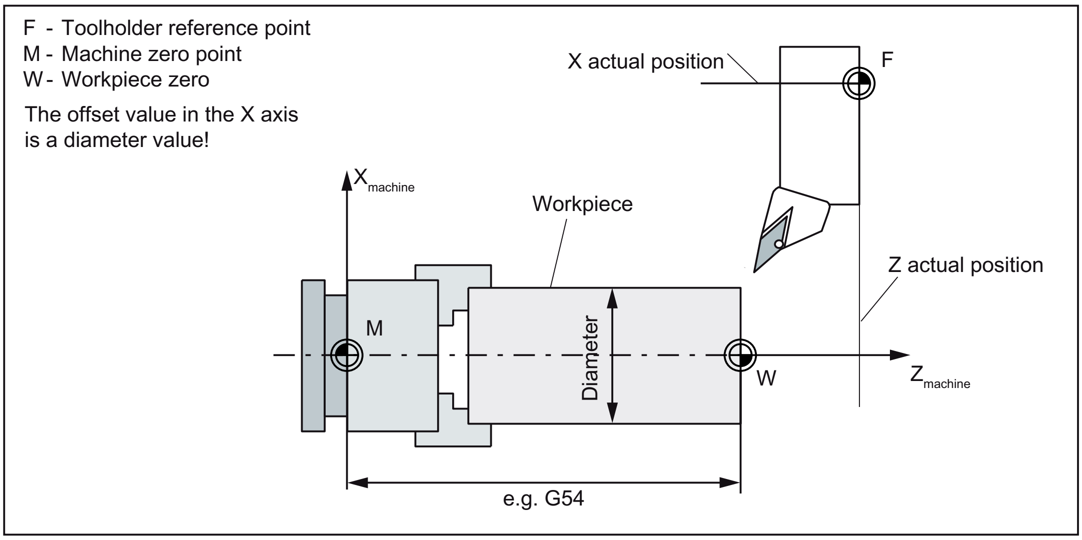


Figure 4-8 Determining the length offsets using the example of a turning tool

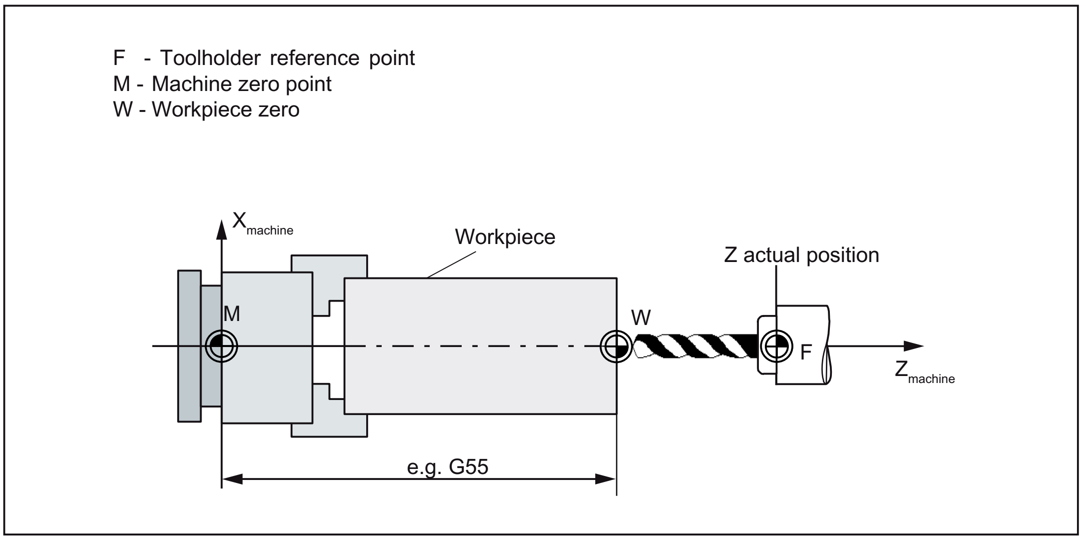


Figure 4-9 Determining the length offsets using the example of a drill: Length 1 / Z axis

---

Note

The figure "Determining the length offsets using the example of a drill: Length 1/Z axis" will only apply, if setting data SD42950 $SC_TOOL_LENGTH_TYPE and SD42940 $SC_TOOL_LENGTH_CONST are equal to "0". Otherwise Length 2 will apply for the drilling and the milling tool.

Prerequisite

A tool must be loaded to use the "Measure tool" function.

Display machine data

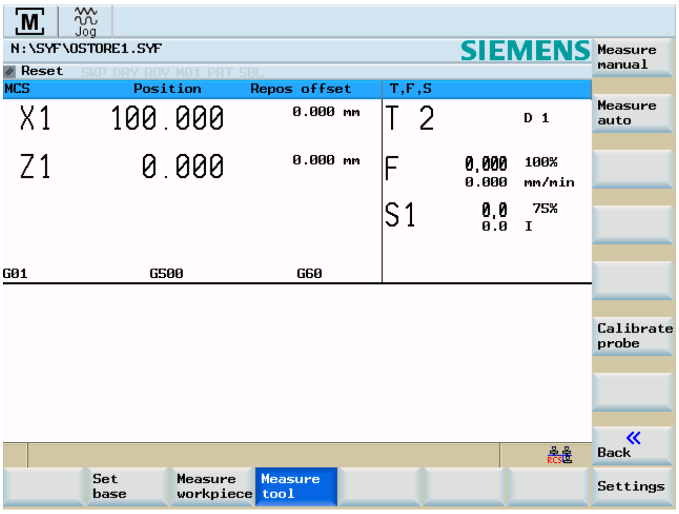
The following display machine data define the display in the "Tool measurement manual" window:

- MD290 CTM_POS_COORDINATE_SYSTEM
  - = 0 -> position of the tool after the turning center (refer to the diagrams above)
  - = 2 -> position of the tool before the turning center
- MD361 USER_MEAS_TOOL_CHANGE
  - = 0 -> not possible to edit the "T" and "D" fields

    The "T" tool currently selected at the machine and its tool offset "D" are manually measured.
  - = 1 -> it is possible to edit the "T" and "D" fields

Tool measurement

Tools, that have not been selected at the machine can also be manually measured.



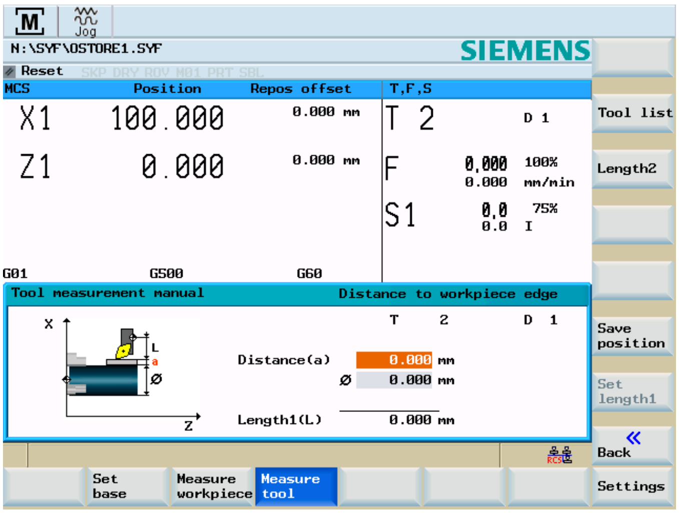
M | Jog
N:\SYF\OSTORE1.SYF SIEMENS
Reset SKP DRY ROV M01 PRT SBL

```
MCS        Position     Repos offset   T,F,S
X1         100.000      0.000 mm       T 2            D 1
Z1         0.000        0.000 mm       F   0.000  100%
                                           0.000  mm/min
                                       S1  0.0    75%
                                           0.0    I
G01        G500         G60
```

Figure 4-10 Selecting manual or semiautomatic measuring

Measuring Manual

The "Tool measurement manual" window is opened with the default setting "Measure length 1 in the X axis".



M | Jog
N:\SYF\OSTORE1.SYF SIEMENS
Reset SKP DRY ROV M01 PRT SBL

```
MCS        Position     Repos offset   T,F,S
X1         100.000      0.000 mm       T 2            D 1
Z1         0.000        0.000 mm       F   0.000  100%
                                           0.000  mm/min
                                       S1  0.0    75%
                                           0.0    I
G01        G500         G60
```

Tool measurement manual   Distance to workpiece edge

```
T  2        D  1
Distance(a)   0.000 mm
Ø             0.000 mm
Length1(L)    0.000 mm
```

Figure 4-11 "Tool measurement manual" window, Length1 (L)

Workpiece parameters and operating sequence to manually measure the tool "Length1"

Enter the following workpiece parameters for the particular length calculation of the tool:

- The thickness of a spacer can be taken into account in the calculation in the distance field (a).
- Enter the workpiece diameter in the "Ø" field.

Save position

- With the cutting edge of the tool, move in the X axis up to the edge of the clamped workpiece or to the spacer. Press "Save position".

  The actual position that has been approached is taken into account in the control.

Set length1

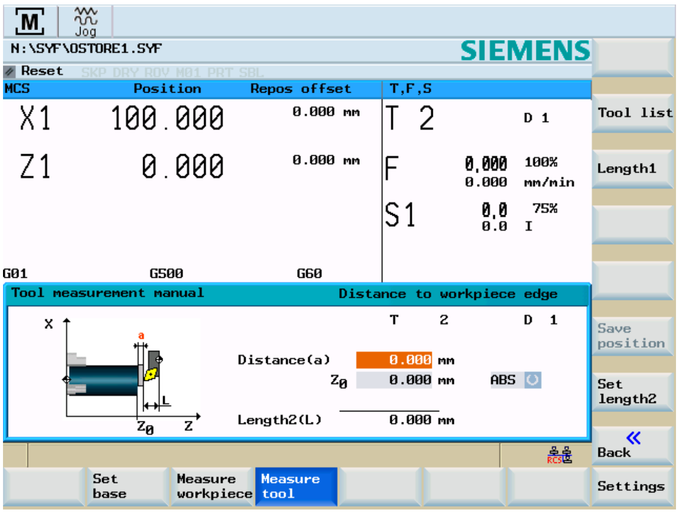
- Press "Set Length1".

  The length value is calculated and saved in the tool offset data.

Workpiece parameters and operating sequence to manually measure the tool "Length2"

Length2

To determine length 2, press "Length2".



N:\SYF\OSTORE1.SYF

Reset SKP DRY ROV M01 PRT SBL

MCS	Position	Repos offset
X1	100.000	0.000 mm
Z1	0.000	0.000 mm

G01 G500 G60

T,F,S
T 2 D 1
F 0.000 0.000 100% mm/min
S1 0.0 0.0 75% I

Tool measurement manual — Distance to workpiece edge

T 2 D 1

Distance(a) 0.000 mm
Z0 0.000 mm ABS
Length2(L) 0.000 mm

Figure 4-12 "Tool measurement manual" window, for Length2 (L)

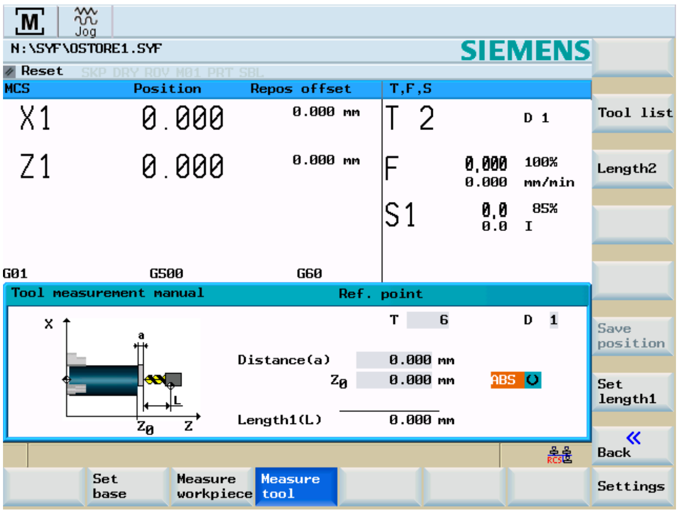
Enter the following workpiece parameters for the particular length calculation of the tool:

- The thickness of a spacer can be taken into account in the calculation in the distance field (a).
- Enter the workpiece edge in the field "Z0", if "ABS" was pre-selected in the adjacent toggle field.

Note

As the known machine coordinate you may also use the work offset already determined (e.g G54 value). This should be selected in the toggle field for the reference point.

Workpiece parameters and operating sequence to manually measure a drill "Length1"



N:\SYF\OSTORE1.SYF

Reset

MCS	Position	Repos offset
X1	0.000	0.000 mm
Z1	0.000	0.000 mm

T,F,S

T 2 D 1

F 0.000 100% 0.000 mm/min

S1 0.0 85% 0.0 I

G01 G500 G60

Tool measurement manual Ref. point

T 6 D 1

Distance(a) 0.000 mm

Z0 0.000 mm ABS

Length1(L) 0.000 mm

Figure 4-13 Tool measurement window, length1 (L) for a drill

Enter the following workpiece parameters for the particular length calculation of the tool:

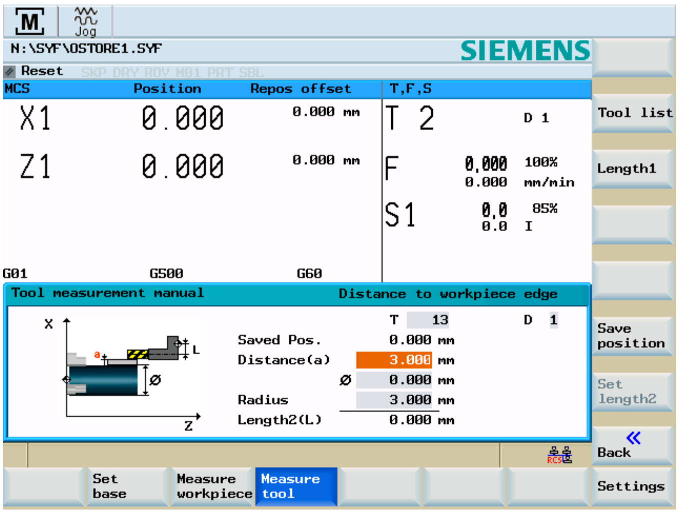
- The thickness of a spacer can be taken into account in the calculation in the distance field (a).
- Enter the workpiece edge in the field "Z0", if "ABS" was pre-selected in the adjacent toggle field.

Note

As the known machine coordinate you may also use the work offset already determined (e.g G54 value). This should be selected in the toggle field for the reference point.

With the cutting edge of the tool, move in the Z axis up to the edge of the clamped workpiece or to the spacer. Then press "Set Length1", the length value is calculated and saved in the tool offset data.

Workpiece parameters and operating sequence to manually measure a driven milling tool "Length2"



```
N:\SYF\OSTORE1.SYF
Reset
MCS   Position   Repos offset      T,F,S
X1    0.000      0.000 mm          T 2            D 1
Z1    0.000      0.000 mm          F   0.000  100%
                                       0.000  mm/min
                                   S1    0.0   85%
                                         0.0   I
G01   G500   G60

Tool measurement manual        Distance to workpiece edge
                     T   13          D  1
Saved Pos.           0.000 mm
Distance(a)          3.000 mm
                 Ø   0.000 mm
Radius               3.000 mm
Length2(L)           0.000 mm
```

Figure 4-14 Manually measuring a driven milling tool

Enter the following workpiece parameters for the particular length calculation of the tool:

- The thickness of a spacer can be taken into account in the calculation in the distance field (a).
- Enter the workpiece diameter in the "Ø" field.
- Enter the milling tool radius in the "Radius" field.

Save position

- With the cutting edge of the tool, move in the X axis up to the edge of the clamped workpiece or to the spacer. Press "Save position".

  The actual position that has been approached is taken into account in the control.

Set length2

- Press "Set Length2".

  The length value is calculated and saved in the tool offset data.

Note

The effect of the softkey is determined by display machine data MD373 MEAS_SAVE_POS_LENGTH2.

= 0 -> the softkey is only active when measuring length 1

= 1 -> the softkey is always active

4.1.4 Determining tool offsets using a probe (auto)

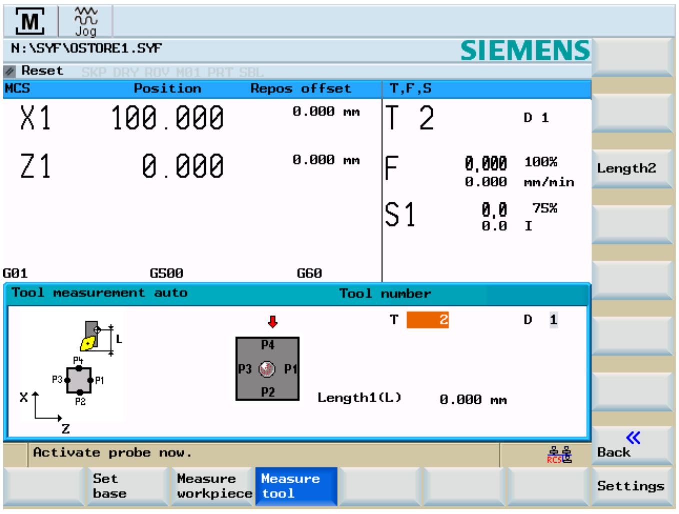
Operating sequence

Tool measurement

Press the "Measure tool" softkey.

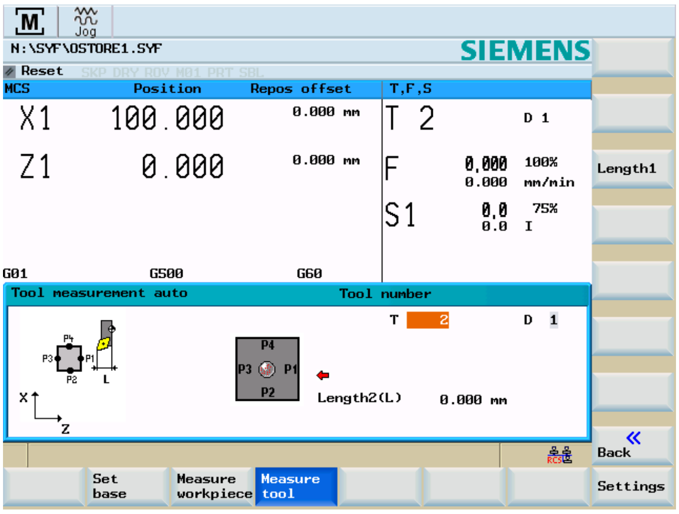
Measuring Auto

The "Measure tool automatically" window is opened.



```
N:\SYF\OSTORE1.SYF
Reset
MCS    Position     Repos offset    T,F,S
X1     100.000      0.000 mm        T 2          D 1
Z1       0.000      0.000 mm        F    0.000   100%
                                         0.000   mm/min
                                        S1   0.0  75%
                                             0.0  I
G01         G500         G60
Tool measurement auto          Tool number
                               T 2        D 1
                               Length1(L)   0.000 mm
Activate probe now.
```

Figure 4-15 "Tool measurement auto" window for Length1 (L)



```
N:\SYF\OSTORE1.SYF
Reset
MCS    Position     Repos offset    T,F,S
X1     100.000      0.000 mm        T 2          D 1
Z1       0.000      0.000 mm        F    0.000   100%
                                         0.000   mm/min
                                        S1   0.0  75%
                                             0.0  I
G01         G500         G60
Tool measurement auto          Tool number
                               T 2        D 1
                               Length2(L)   0.000 mm
```

Figure 4-16 "Tool measurement auto" window, Length2 (L)

Display machine data

The following display machine data defines the display in the "Tool measurement auto" window:

- MD290 CTM_POS_COORDINATE_SYSTEM
  - = 0 -> position of the tool after the turning center (refer to the diagrams above)
  - = 2 -> position of the tool before the turning center

"Tool measurement auto" screen form

This screen form allows you to enter the tool number and tool offset number.

After the screen form has been opened, the input fields are filled with the data of the tool currently working.

The tool can be either

- the currently active tool of the NC (loaded via a part program) or
- a tool loaded by the PLC.

If the tool was loaded by the PLC, the tool number in the input screen can be different than that in the **T,F,S** window.

If you change the tool number, no automatic tool change will be performed using this function. However, the measurement results are assigned to the entered tool.

Measuring process

The measuring probe is approached using traversing keys.

After the "Probe tripped" symbol has appeared, release the traversing key and wait until the measuring process is completed.

During the automatic measurement, a dial gauge is displayed, which symbolizes the measuring process currently active.

Note

To create the measuring program, the safety clearance parameters from the "Settings" screen form and the feedrate from the "Probe data" screen form are used (see chapter "Probe settings").

If several axes are moved simultaneously, no offset data can be calculated.

Procedure for "Probe tripped"

A solid circle on the screen indicates that the measuring probe has been tripped.

Release the axis direction key after tripping of the probe.

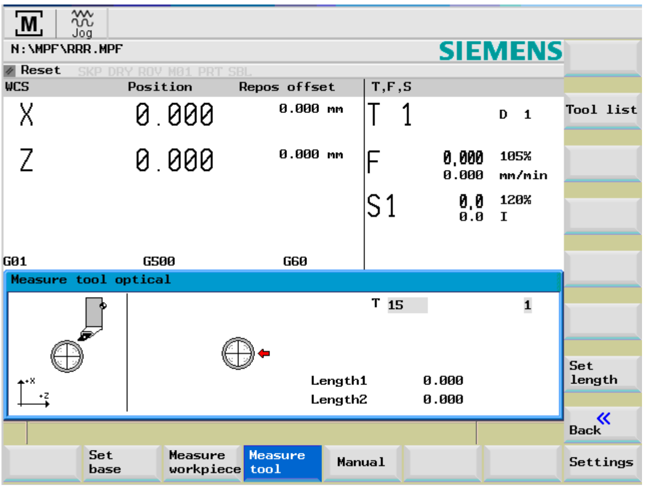
After release of the axis direction key, the control automatically creates an internal measuring program in the program memory and starts it.

This measuring program causes the probe to be approached max. three times in order to deliver measured values to the control.

If no measured value is transferred to the control after the third approach of the probe, a message will be displayed informing the operator that no measured values could be collected.

All axes involved in the measuring process must be approached this way.

4.1.5 Determining the tool offsets using optical measuring instruments



```
N:\MPF\RRR.MPF
Reset   SKP DRY ROV M01 PRT SBL
WCS      Position     Repos offset   T,F,S
X        0.000        0.000 mm       T  1            D  1
Z        0.000        0.000 mm       F      0,000   105%
                                            0.000   mm/min
                                     S1       0,0   120%
                                              0.0   I
G01      G500         G60
Measure tool optical
                                     T 15                 1
Length1   0.000
Length2   0.000
```

Figure 4-17 Measuring using an optical measuring instrument (for the T and D input fields, please refer to "Measuring using a probe")

Measuring process

For measuring, traverse the tool until its tip appears in the crosshair. With a milling tool, use the highest point of the cutting edge to determine the tool length.

Then press the "Set length" softkey to calculate the offset values.

4.1.6 Probe settings

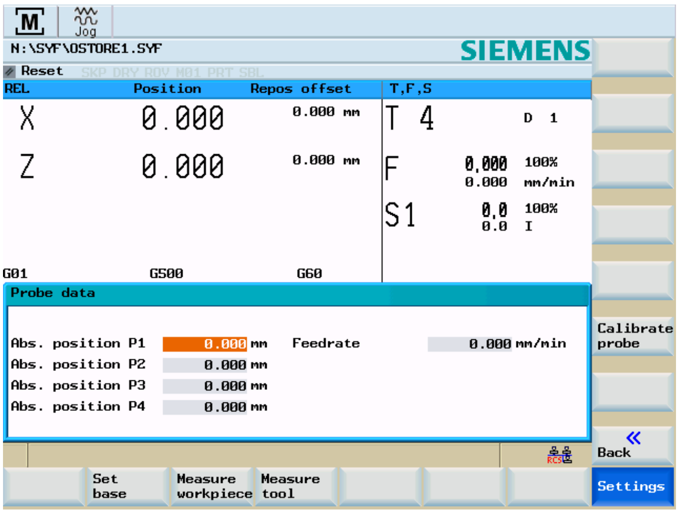
Settings

Select the "Settings" softkey.

Probe data

The screen form below is used to store the coordinates of the probe and to set the axis feedrate for the automatic measuring process.

All position values refer to the machine coordinate system.



Probe data

Abs. position P1	0.000 mm	Feedrate	0.000 mm/min
Abs. position P2	0.000 mm		
Abs. position P3	0.000 mm		
Abs. position P4	0.000 mm		

Figure 4-18 "Probe data" input screen

Parameter	Significance
Absolute position P1	Absolute position of the probe in Z direction
Absolute position P2	Absolute position of the probe in X+ direction
Absolute position P3	Absolute position of the probe in Z+ direction
Absolute position P4	Absolute position of the probe in X- direction
Feedrate	Feedrate with which the tool approaches the probe

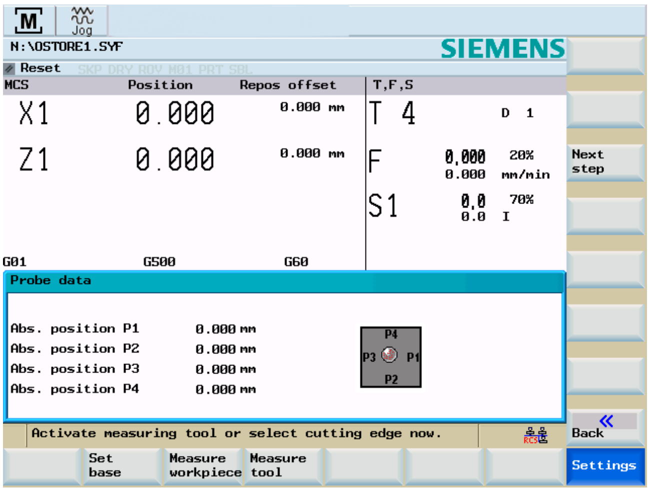
Probe calibration

Calibrate probe

The probe can be calibrated either in the "Settings" menu or in the "Measure tool" menu. The four points of the probe must be approached.

For calibration, use a tool of the type 500 with tool tip position 3 or 4.

The offset parameters required to determine the four probe positions can be written to the data records of two cutting edges.



N:\OSTORE1.SYF

Reset

MCS	Position	Repos offset
X1	0.000	0.000 mm
Z1	0.000	0.000 mm

T,F,S

T 4 D 1

F 0.000 0.000 20% mm/min

S1 0.0 0.0 70% I

G01 G500 G60

Probe data

Abs. position P1	0.000 mm
Abs. position P2	0.000 mm
Abs. position P3	0.000 mm
Abs. position P4	0.000 mm

Activate measuring tool or select cutting edge now.

Figure 4-19 Calibrating the probe

After the screen form has appeared, an animation signaling the step to be executed is displayed next to the current positions of the probe. This point must be approached with the appropriate axis.

After the "Probe tripped" symbol has appeared, release the traversing key and wait until the measuring process is completed.

During the automatic measurement, a dial gauge is displayed, which symbolizes the measuring process currently active.

The positions delivered by the measuring program serve to calculate the real probe position.

The measuring function can be quit without approaching all positions. The points already sensed are stored.

Note

To create the measuring program, the "Safety clearance" parameters from the "Settings" screen form and the feedrate from the "Probe data" screen form are used.

If several axes are moved simultaneously, no offset data can be calculated.

Use the "Next Step" function to skip a point if this is not needed for measuring.

4.2 Tool monitoring

Functionality

This function is available for SINUMERIK 802D sl plus and 802D sl pro.

The following types of active cutting edge monitoring for the active tool are possible:

- Monitoring the service life

  By activating service life monitoring, the service life during the action time of the tool (G1, G2, G3) is monitored.

- Monitoring of the workpiece count

  By activating workpiece count monitoring, the workpiece count is monitored via the program command SETPIECE( ) at the end of the part program.

See also the chapter "Language commands for tool monitoring".

Note

Set the following machine data to enable the "tool monitoring" function:

- General machine data
  MD18080 $MN_MM_TOOL_MANAGEMENT_MASK bit 1 = 1: Memory for the monitoring data (WZMO) is provided.
- Channel machine data
  MD20310 $MC_TOOL_MANAGEMENT_MASK bit 1 = 1: Tool management monitoring function is active.

After the machine data were changed, proceed as follows at the control:

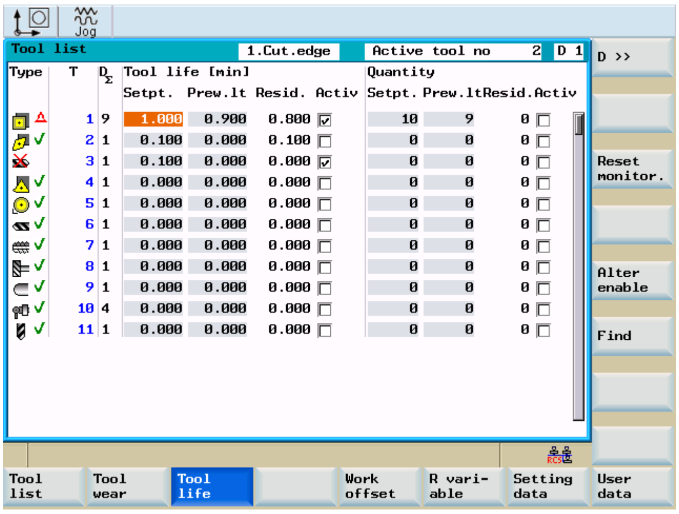
1. Backup the commissioning archive data (Drive/NC/PLC/HMI).
2. Download the commissioning archive data that was backed up.

Operating sequence

OFFSET PARAM

Tool monitoring

Monitoring is carried out in the <OFFSET PARAM> > "Tool monitoring" operating area.



Tool list | 1.Cut.edge | Active tool no 2 | D 1

Type	T	DΣ	Tool life [min]				Quantity			
			Setpt.	Prew.lt	Resid.	Activ	Setpt.	Prew.lt	Resid.	Activ
△	1	9	1.000	0.900	0.800	☑	10	9	0	□
✓	2	1	0.100	0.000	0.100	□	0	0	0	□
✗	3	1	0.100	0.000	0.000	☑	0	0	0	□
✓	4	1	0.000	0.000	0.000	□	0	0	0	□
✓	5	1	0.000	0.000	0.000	□	0	0	0	□
✓	6	1	0.000	0.000	0.000	□	0	0	0	□
✓	7	1	0.000	0.000	0.000	□	0	0	0	□
✓	8	1	0.000	0.000	0.000	□	0	0	0	□
✓	9	1	0.000	0.000	0.000	□	0	0	0	□
✓	10	4	0.000	0.000	0.000	□	0	0	0	□
✓	11	1	0.000	0.000	0.000	□	0	0	0	□

Figure 4-20 Tool monitoring

Each monitoring type is displayed in four columns.

- Setpoint
- Prewarning limit
- Residual value
- active

Using the checkbox element in the fourth column, the monitoring type can be switched to active/inactive.

The symbols in the "Type" column have the following meaning:

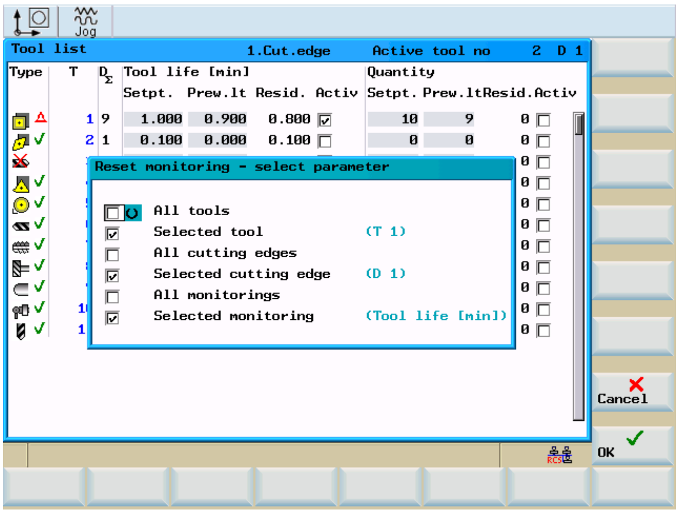
△ Prewarning limit reached

✓ The tool is released

✗ Tool disabled

Reset monitoring

Monitoring for the selected tool is reset.



Tool list 1.Cut.edge Active tool no 2 D 1

Type	T	D	Tool life [min] Setpt.	Prew.lt	Resid.	Activ	Quantity Setpt.	Prew.lt	Resid.	Activ
	1	9	1.000	0.900	0.800	☑	10	9	0	□
✓	2	1	0.100	0.000	0.100	□	0	0	0	□

Reset monitoring - select parameter

- □ All tools
- ☑ Selected tool (T 1)
- □ All cutting edges
- ☑ Selected cutting edge (D 1)
- □ All monitorings
- ☑ Selected monitoring (Tool life [min])

Figure 4-21 Reset monitoring

4.3 Entering / modifying a work offset

Functionality

After the reference point approach, the actual-value memory and thus also the actual-value display are referred to the machine zero point. A machining program, however, is always referred to the workpiece zero. This offset must be entered as the work offset.

Operating sequences

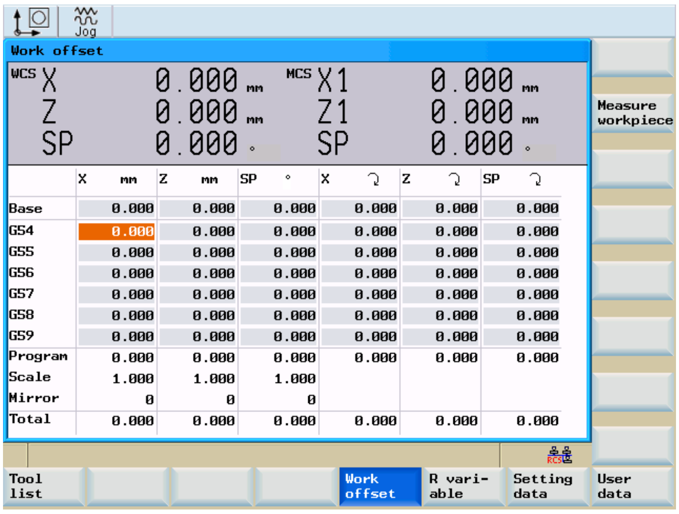
OFFSET PARAM

Press the <OFFSET PARAM> key.

Work offset

Use <OFFSET PARAM> and "Work offset" to select the work offset.

An overview of all settable work offsets will appear on the screen. The screen form additionally contains the values of the basic offset of the programmed work offset and the active scaling factors, the "Mirroring active" status display and the total of all active work offsets.



Work offset

WCS X 0.000 mm   MCS X1 0.000 mm
Z 0.000 mm   Z1 0.000 mm
SP 0.000 °   SP 0.000 °

	X mm	Z mm	SP °	X ⤵	Z ⤵	SP ⤵
Base	0.000	0.000	0.000	0.000	0.000	0.000
G54	0.000	0.000	0.000	0.000	0.000	0.000
G55	0.000	0.000	0.000	0.000	0.000	0.000
G56	0.000	0.000	0.000	0.000	0.000	0.000
G57	0.000	0.000	0.000	0.000	0.000	0.000
G58	0.000	0.000	0.000	0.000	0.000	0.000
G59	0.000	0.000	0.000	0.000	0.000	0.000
Program	0.000	0.000	0.000	0.000	0.000	0.000
Scale	1.000	1.000	1.000			
Mirror	0	0	0			
Total	0.000	0.000	0.000	0.000	0.000	0.000

Figure 4-22 The "Work offset" window

- Position the cursor bar on the input field to be changed
- Enter the value(s). Either move the cursor or press the <Input> key to accept the values from the input fields into the work offsets.

4.3.1 Determining the work offset

Prerequisite

You have selected the window with the relevant work offset (e.g. G54) and the axis you want to determine for the offset.

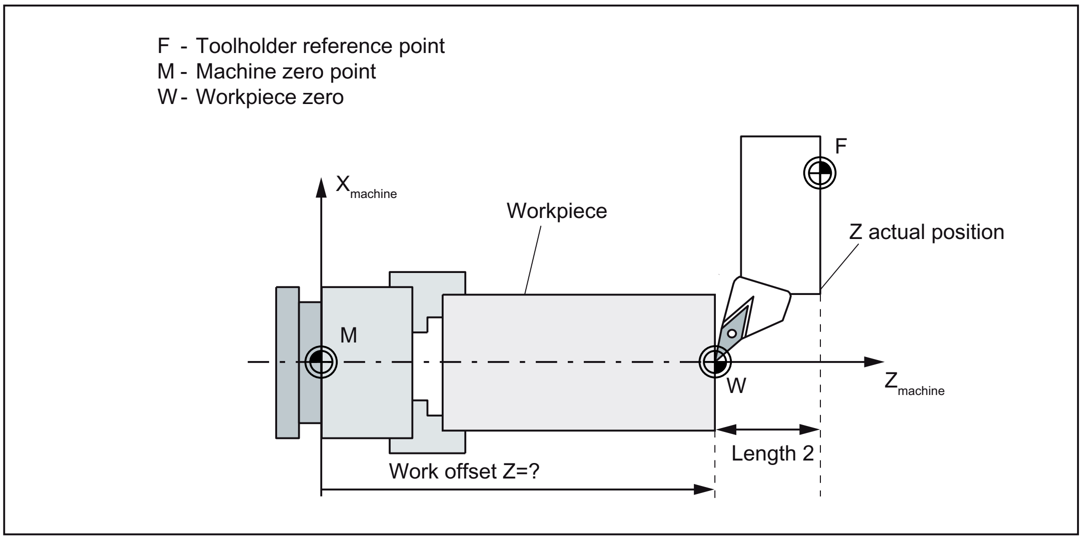


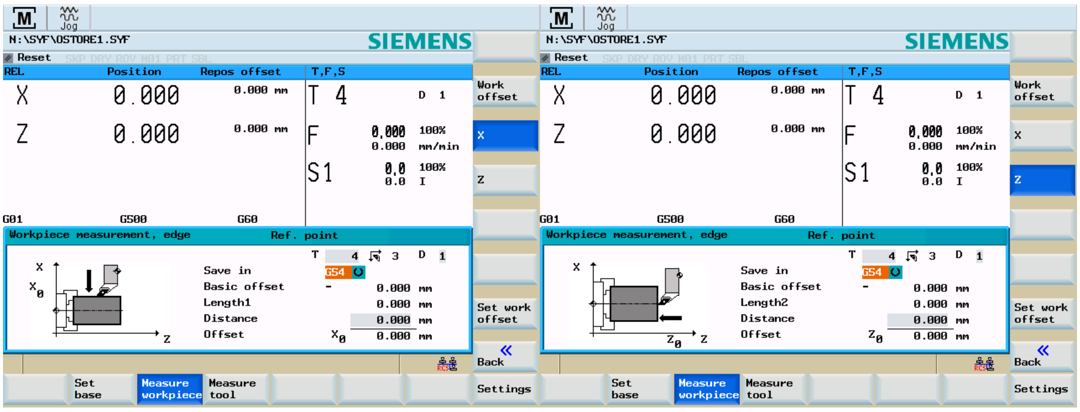
Figure 4-23 Determining the work offset Z axis

Procedure

Measure workpiece

Press the "Measure workpiece" softkey. The control system will switch to the "Position" operating area and will open the dialog box for measuring the work offsets. The selected axis is displayed as a softkey with a blue background.

Then scratch the workpiece with the tool tip.



```
N:\SYF\OSTORE1.SYF                                SIEMENS
Reset   SKP DRY ROV M01 PRT SBL
REL     Position     Repos offset   T,F,S
X       0.000        0.000 mm       T 4          D 1
Z       0.000        0.000 mm       F   0.000  100%
                                        0.000  mm/min
                                    S1    0.0  100%
                                          0.0  I
G01     G500         G60
Workpiece measurement, edge        Ref. point
                          T    4   3   D  1
            Save in       G54
            Basic offset  -    0.000 mm
            Length1            0.000 mm
            Distance           0.000 mm
            Offset        X0   0.000 mm
```

```
N:\SYF\OSTORE1.SYF                                SIEMENS
Reset   SKP DRY ROV M01 PRT SBL
REL     Position     Repos offset   T,F,S
X       0.000        0.000 mm       T 4          D 1
Z       0.000        0.000 mm       F   0.000  100%
                                        0.000  mm/min
                                    S1    0.0  100%
                                          0.0  I
G01     G500         G60
Workpiece measurement, edge        Ref. point
                          T    4   3   D  1
            Save in       G54
            Basic offset  -    0.000 mm
            Length2            0.000 mm
            Distance           0.000 mm
            Offset        Z0   0.000 mm
```

Figure 4-24 Determine work offset in X Determine work offset in Z

Set work offset

The softkey calculates the offset and displays the result in the offset field.

4.4 Program setting data

Functionality

The setting data are used to define the settings for the operating states. These can be changed as necessary.

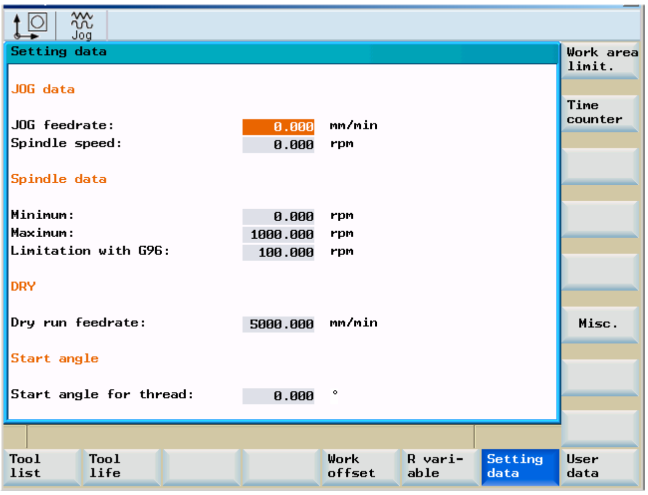
Operating sequence

OFFSET PARAM

These can be found in the <OFFSET PARAM> operating area.

Setting data

Press the "Setting data" softkey. The start screen "Setting data" is opened. Other softkey functions are available here with which you can set various control system options.



```
Setting data

JOG data

JOG feedrate:                 0.000  mm/min
Spindle speed:                0.000  rpm

Spindle data

Minimum:                      0.000  rpm
Maximum:                   1000.000  rpm
Limitation with G96:        100.000  rpm

DRY

Dry run feedrate:          5000.000  mm/min

Start angle

Start angle for thread:       0.000  °
```

Figure 4-25 Setting data start screen

- JOG feedrate

  Feedrate value in JOG mode
  If the feedrate value is zero, the control system will use the value stored in the machine data.

- Spindle

  Spindle speed

- Minimum / maximum

  A limitation of the spindle speed in the "Max." (G26) / "Min." (G25) fields can only be performed within the limit values defined in the machine data.

- Limitation using G96

  Programmable upper speed limitation (LIMS) at constant cutting rate (G96).

- Dry run feed (DRY)

  The feedrate which can be entered here will be used instead of the programmed feedrate in the AUTOMATIC mode if the "Dry run feed" function is selected.

- Starting angle for thread (SF)

  For thread cutting, a start position for the spindle is displayed as the start angle. A multiple thread can be cut by changing the angle when the thread cutting operation is repeated.

Place the cursor bar on the input field to be modified and enter the value.

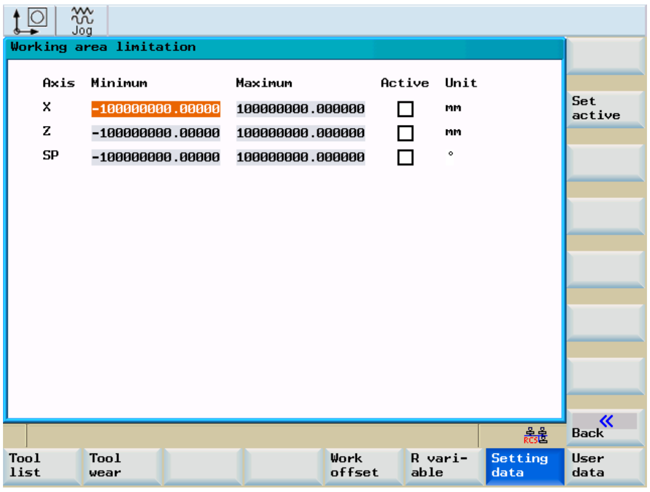
INPUT

Either press the <Input> key or move the cursor to confirm.

Softkeys

Limit working area

The working area limitation is active with geometry and additional axes. If you want to use a working area limitation, its values can be entered in this dialog box. Selecting the "Set active" softkey enables/disables the values for the axis highlighted by the cursor.



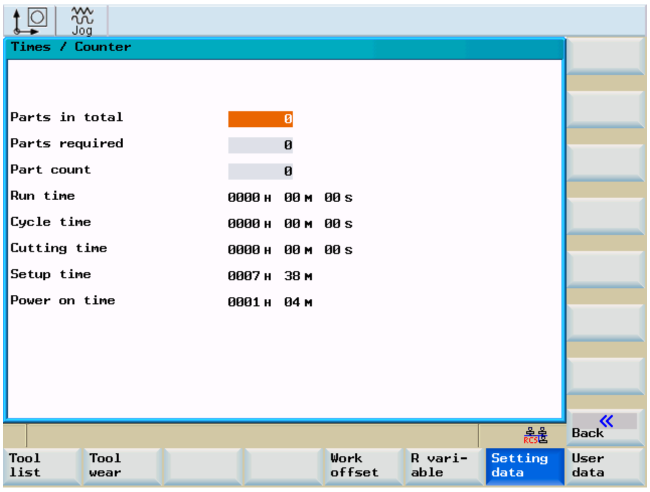
Working area limitation

Axis	Minimum	Maximum	Active	Unit
X	-100000000.00000	100000000.000000	□	mm
Z	-100000000.00000	100000000.000000	□	mm
SP	-100000000.00000	100000000.000000	□	°

Figure 4-26 Working area limitation

Times Multiplier

Times Counters



Times / Counter

Parts in total	0
Parts required	0
Part count	0
Run time	0000 H 00 M 00 s
Cycle time	0000 H 00 M 00 s
Cutting time	0000 H 00 M 00 s
Setup time	0007 H 38 M
Power on time	0001 H 04 M

Figure 4-27 Times, Counters

Meaning:

- Total parts: Total number of workpieces produced (total actual)
- Parts requested: Number of workpieces required (workpiece setpoint)
- Number of parts: This counter registers the number of all workpieces produced since the starting time.

> **Note**
>
> The counter functionality is set using the following channel-specific machine data:
>
> - MD27880 $MC_PART_COUNTER, the workpiece counter is activated
> - MD27882 $MC_PART_COUNTER_MCODE[0-2], workpiece counting with user defined M command

- Total runtime: Total runtime of NC programs in AUTOMATIC mode

  In the AUTOMATIC mode, the runtimes of all programs between NC START and end of program / RESET are summed up. The timer is zeroed with each power-up of the control system.

- Program runtime Active tool operating times

  The runtime between NC Start and End of program / Reset is measured in the selected NC program. The timer is reset with the start of a new NC program.

- Feedrate runtime

  The runtime of the path axes is measured in all NC programs between NC START and end of program / RESET without rapid traverse active and with the tool active. The measurement is interrupted when a dwell time is active.

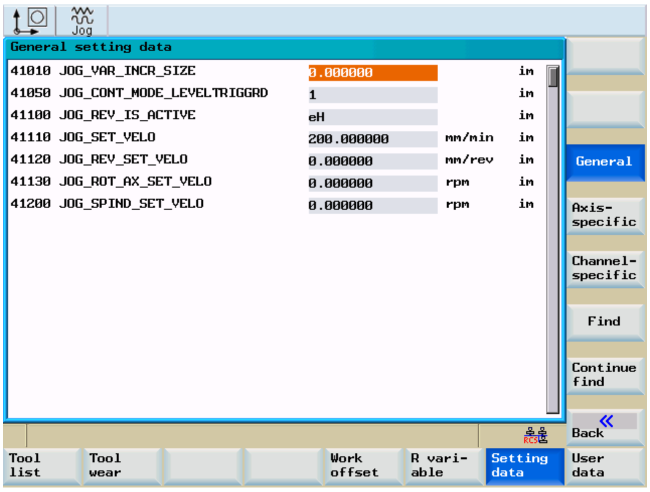
The timer is automatically reset to zero in the case of a "Control power-up with default values".

Misc.

Use this function to display all setting data for the control system in the form of a list. The setting data are divided up into general, axis-specific and channel-specific data.

They can be selected using the following softkey functions:

- "General"
- "Axis-spec."
- "Channel-spec."



```
General setting data
41010 JOG_VAR_INCR_SIZE              0.000000             im
41050 JOG_CONT_MODE_LEVELTRIGGRD     1                    im
41100 JOG_REV_IS_ACTIVE              eH                   im
41110 JOG_SET_VELO                   200.000000  mm/min   im
41120 JOG_REV_SET_VELO               0.000000    mm/rev   im
41130 JOG_ROT_AX_SET_VELO            0.000000    rpm      im
41200 JOG_SPIND_SET_VELO             0.000000    rpm      im
```

Figure 4-28 General setting data

4.5 R parameters - "Offset/Parameter"operating area

Functionality

In the "R parameters" start screen, any R parameters that exist within the control system are listed. These global parameters can be set or queried by the programmer of the part program for any purpose in the program and can be changed as required.

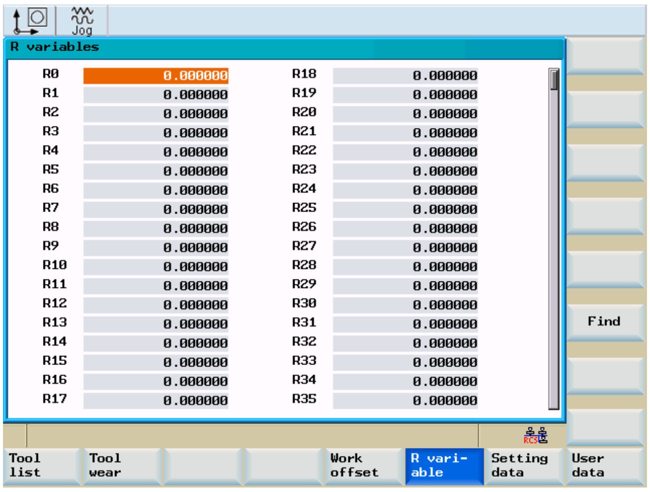
Operating sequence

OFFSET PARAM

These can be found in the <OFFSET PARAM> operating area.

R para-meters

Press the <R variable> softkey. The "R variables" start screen appears.



R variables

R0	0.000000	R18	0.000000
R1	0.000000	R19	0.000000
R2	0.000000	R20	0.000000
R3	0.000000	R21	0.000000
R4	0.000000	R22	0.000000
R5	0.000000	R23	0.000000
R6	0.000000	R24	0.000000
R7	0.000000	R25	0.000000
R8	0.000000	R26	0.000000
R9	0.000000	R27	0.000000
R10	0.000000	R28	0.000000
R11	0.000000	R29	0.000000
R12	0.000000	R30	0.000000
R13	0.000000	R31	0.000000
R14	0.000000	R32	0.000000
R15	0.000000	R33	0.000000
R16	0.000000	R34	0.000000
R17	0.000000	R35	0.000000

Figure 4-29 "R parameters" start screen

Place the cursor bar on the input field to be modified and enter the values.

INPUT

Either press the <Input> key or move the cursor to confirm the entry.

Find

Searching for R variables

Manually Controlled Mode 5

5.1 Manually Controlled Mode

The manually controlled mode is possible in JOG and MDA modes.

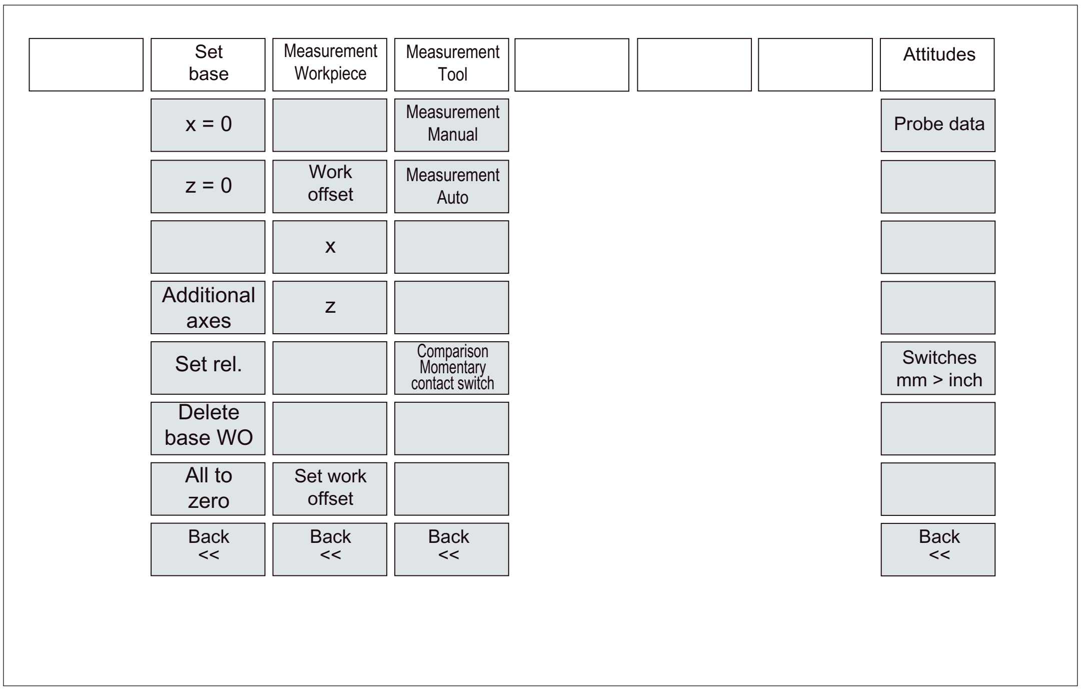


Figure 5-1 Menu tree JOG, "Position" operating area

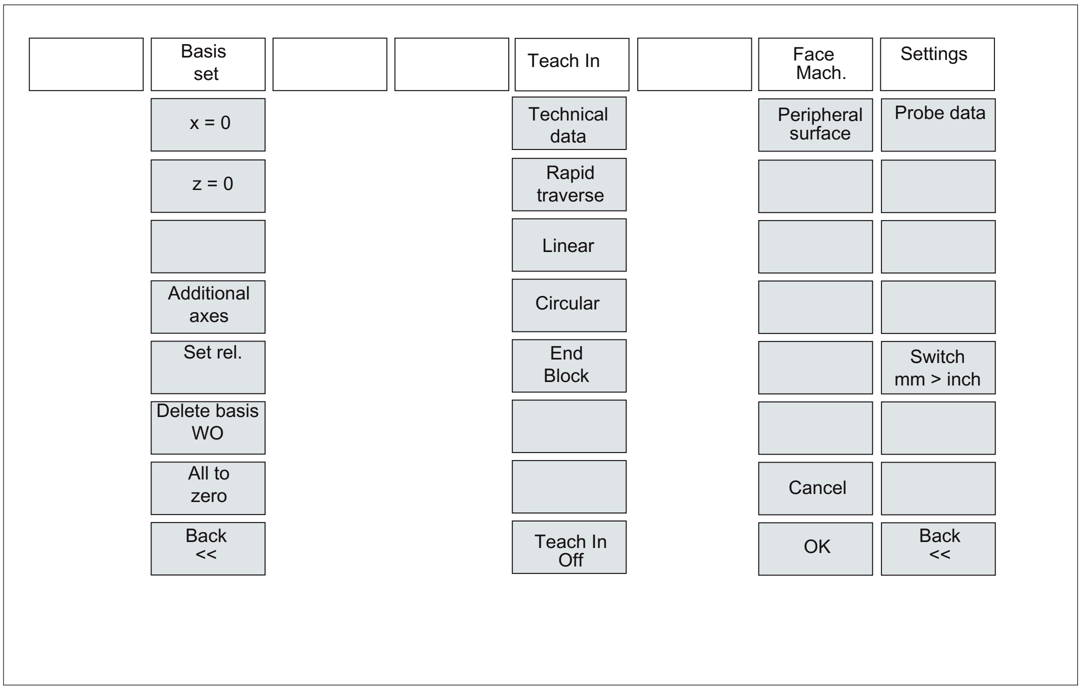


Figure 5-2 Menu tree MDA, "Position" operating area

5.2 JOG mode - "Position" operating area

Operating sequences

JOG

Use the <JOG> key on the machine control panel to select the Jog mode.

+X

To traverse the axes, press the appropriate key of the X or Z axis.

-Z

The axes will traverse continuously at the velocity stored in the setting data until the key is released. If the value of the setting data is zero, the value stored in the machine data is used.

If necessary set the velocity using the override switch.

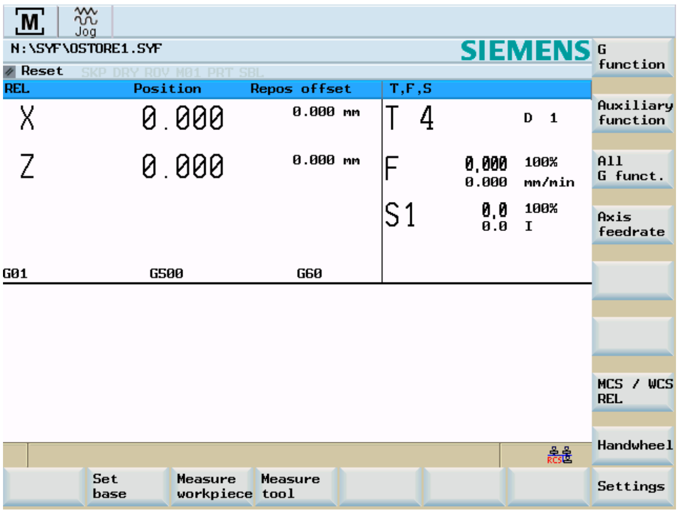
RAPID

When you additionally press the <Rapid traverse override> key, the selected axis will be traversed at rapid traverse speed until both keys are released.

[VAR]

In the <Increment> mode, you can traverse by adjustable increments using the same operating sequence. The set number of increments is displayed in the status area. To deselect, press <JOG> again.

The JOG start screen displays the position, feedrate and spindle values, as well as the current tool.



```
N:\SYF\OSTORE1.SYF
Reset   SKP DRY ROV M01 PRT SBL
REL         Position     Repos offset   T,F,S
X           0.000        0.000 mm       T 4            D 1
Z           0.000        0.000 mm       F      0.000  100%
                                               0.000  mm/min
                                        S1       0.0  100%
                                                 0.0  I
G01         G500         G60
```

Figure 5-3 "JOG" start screen

Parameter

Table 5- 1 Description of the parameters in the "JOG" start screen

Parameter	Explanation
MCS X Z	Displays the axes existing in the machine coordinate system (MCS) or in the workpiece coordinate system (WCS)
+ X -Z	If you traverse an axis in the positive (+) or negative () direction, a plus or minus sign will appear in the relevant field. If the axis is already in the required position, no sign is displayed.
Position mm	These fields display the current position of the axes in the MCS or WCS.
Repos. offset	If the axes are traversed in the "Program interrupted" condition in the <JOG> mode, the distance traversed by each axis is displayed referred to the interruption point.
G function	Displays important G functions
Spindle S r.p.m.	Displays the actual value and the setpoint of the spindle speed
Feed F mm/min	Displays the path feedrate actual value and setpoint
Tool	Displays the currently active tool with the current edge number

Note

If a second spindle is integrated into the system, the work spindle will be displayed using a smaller font. The window will always display the data of only one spindle.

The control system displays the spindle data according to the following aspects:

The master spindle (large display) is displayed:

- Idle,
- at spindle start
- with both spindles active

The work spindle (small display) is displayed:

- when starting the work spindle

The power bar applies to the spindle currently active. With both master spindle and work spindle active, the master spindle performance bar is displayed.

Softkeys

Set basis

This softkey is used to set the base work offset or a temporary reference point in the relative coordinate system. After opening, this function can be used to set the base work offset.

The following subfunctions are provided:

- Direct input of the desired axis position

  In the input window, position the input cursor on the desired axis; thereafter, enter the new position. Then, press "Input" or move the cursor to confirm your input.
- Setting all axes to zero

  The softkey function "All to zero" overwrites the current position of the appropriate axis with zero.
- Setting individual axes to zero

  Selecting the softkey "X=0" or "Z=0" overwrites the current position with zero.

Note

A changed base work offset acts independently of any other work offsets.

Use the "Set rel." softkey to switch the display to the relative coordinate system.

Any subsequent inputs will change the reference point in this coordinate system.

The value of the axis position shown can be specified as a reference point for the relative coordinate system.

Here it is useful to set a reference point "X=0" or "Z=0", or to directly enter a reference point for the axes in the display.

Measure workpiece

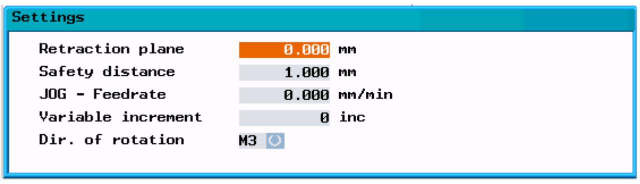
Use this softkey to determine the work offset (see section "Set up")

Tool measurement

Use this softkey to measure the tool offsets (see section "Set up")

Settings

The input screen shown below is intended to set the retraction plane, the safety clearance and the direction of rotation of the spindle for automatically generated part programs in the MDA mode. Furthermore, the values for the JOG feedrate and the variable size of increments can be set.



Settings

Retraction plane	0.000	mm
Safety distance	1.000	mm
JOG - Feedrate	0.000	mm/min
Variable increment	0	inc
Dir. of rotation	M3	

Figure 5-4 Settings

- Retraction plane

  The "Face" function retracts the tool to the specified position (Z position) after the function has been executed.

- Safety clearance

  Safety distance to the workpiece surface

  This value defines the minimum distance between the workpiece surface and the workpiece. It is used by the "Face" and "Automatic tool gauging" functions.

- JOG feedrate

  Feedrate value in JOG mode

- Direction of rotation

  Direction of rotation of the spindle for automatically generated programs in the JOG and MDA modes.

Use this softkey to switch between the metric and the inch dimension systems

5.2.1 Assigning handwheels

Operating sequence

JOG

Select the <JOG> operating mode.

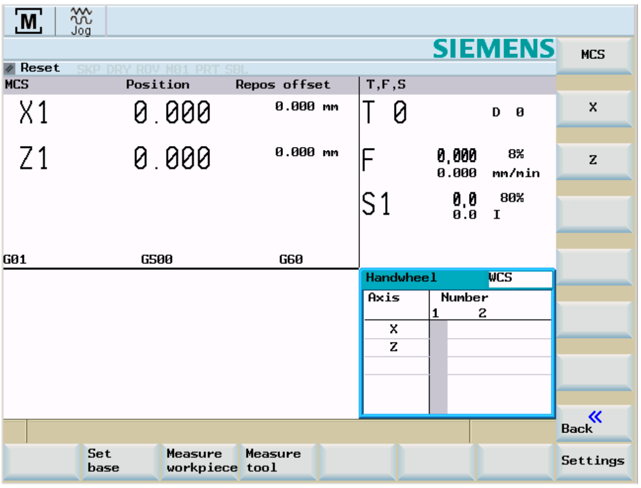
Handwheel

Press the "Handwheel" softkey. The "Handwheel" window appears on the screen.

After the window has been opened, all axis identifiers are displayed in the "Axis" column, which simultaneously appear in the softkey bar.

Select the desired handwheel using the cursor. Thereafter, press the relevant axis softkey for the desired axis for assignment or deselection

The ☑ symbol appears in the window.



MCS	Position	Repos offset
X1	0.000	0.000 mm
Z1	0.000	0.000 mm

G01 G500 G60

T,F,S

T 0 D 0

F 0.000 8%
0.000 mm/min

S1 0.0 80%
0.0 I

Handwheel WCS

Axis	Number 1	2
X		
Z		

Figure 5-5 "Handwheel" menu screen

MCS

Use the "MCS" softkey to select the axes from the machine or workpiece coordinate system for handwheel assignment. The current setting is displayed in the window.

5.3 MDA mode (manual input) "Position" operating area

Functionality

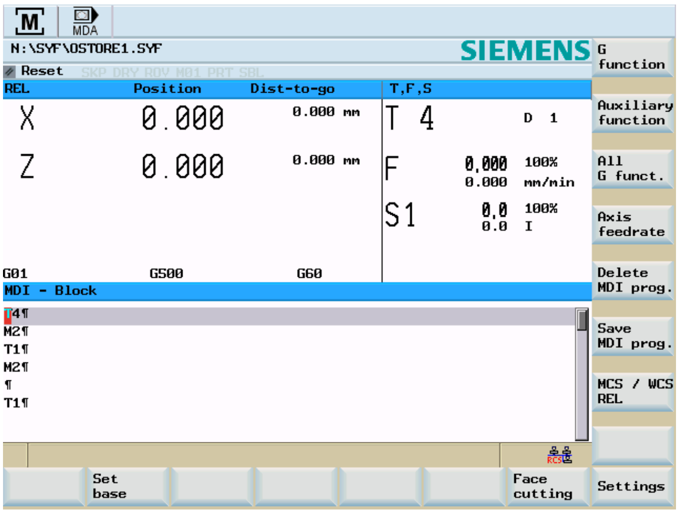
In the MDA mode, you can create or execute a part program.

> ⚠ CAUTION
>
> The Manual mode is subject to the same safety interlocks as the fully automatic mode.
>
> Furthermore, the same prerequisites are required as in the fully automatic mode.

Operating sequence

MDA

Select the <MDA> mode through the machine control panel.



```
N:\SYF\OSTORE1.SYF                                    SIEMENS
Reset  SKP DRY ROV M01 PRT SBL
REL            Position      Dist-to-go     T,F,S
X              0.000         0.000 mm       T  4            D  1
Z              0.000         0.000 mm       F      0,000   100%
                                                   0.000   mm/min
                                            S1       0,0   100%
                                                     0.0   I
G01            G500          G60
MDI - Block
T4¶
M2¶
T1¶
M2¶
¶
T1¶
```

Figure 5-6 "MDA" start screen

Enter one or several blocks using the keyboard.

CYCLE START

Press <NC START> to start machining. During machining, editing of the blocks is no longer possible.

After machining, the contents are preserved so that the machining can be repeated by pressing <NC START> once again.

Parameters

Table 5- 2 Description of the parameters in the "MDA" working window

Parameter	Explanation
MCS X Z	Displays the existing axes in the MCS or WCS
+X -Z	If you traverse an axis in the positive (+) or negative () direction, a plus or minus sign will appear in the relevant field. If the axis is already in the required position, no sign is displayed.
Position mm	These fields display the current position of the axes in the MCS or WCS.
Distance-to-go	This field displays the distance to go of the axes in the MCS or WCS.
G function	Displays important G functions
Spindle S r.p.m.	Displays the actual value and the setpoint of the spindle speed
Feedrate F	Displays the path feedrate actual value and setpoint in mm/min or mm/rev.
Tool	Displays the currently active tool with the current edge number (T..., D...).
Editing window	In the "Stop" or "Reset" program state, an editing window serves to input a part program block.

Note

If a second spindle is integrated into the system, the work spindle will be displayed using a smaller font. The window will always display the data of only one spindle.

The control system displays the spindle data according to the following aspects:

- The master spindle is displayed:
  - in the idle state,
  - when the spindle starts,
  - if both spindles are active.
- The work spindle is displayed:
  - when the work spindle starts.

The power bar applies to the spindle currently active.

Softkeys

The horizontal softkeys are explained in the chapter titled "JOG mode - 'Position' operating area" (Page 67).

G-function

The G function window displays G functions whereby each G function is assigned to a group and has a fixed position in the window.
Use the <Page Up> or <Page Down> keys to display additional G functions. Selecting the softkey repeatedly will close the window.

Auxiliary function

This window displays the auxiliary and M functions currently active. Selecting the softkey repeatedly will close the window.

All G-functions

All the G functions are displayed.

Axis feedrate

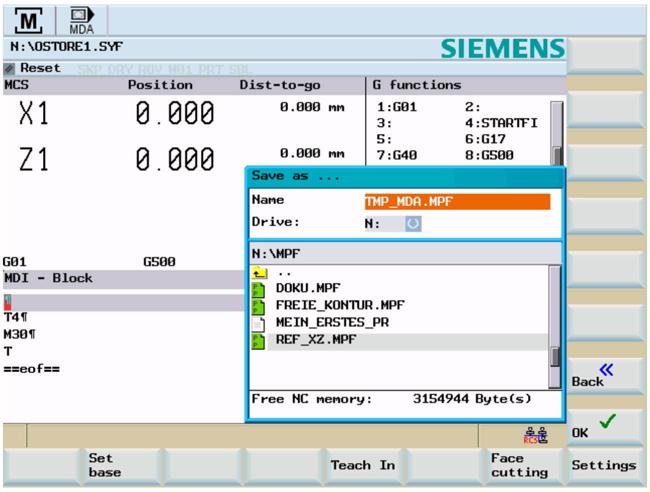
Use this softkey to display the "Axis feedrate" window.
Pressing the softkey repeatedly will close the window.

Delete MDA progr.

Use this function to delete blocks from the program window.

Save MDA prog.

Enter a name in the input field with which you wish the MDA program to be saved in the program directory. Alternatively, you may select an existing program from the list.
Use the TAB key to change between input field and program list.



N:\OSTORE1.SYF

MCS	Position	Dist-to-go
X1	0.000	0.000 mm
Z1	0.000	0.000 mm

G functions

1:G01	2:
3:	4:STARTFI
5:	6:G17
7:G40	8:G500

```
G01                 G500
MDI - Block
¶
T4¶
M30¶
T
==eof==
```

Save as ...

Name TMP_MDA.MPF

Drive: N:

N:\MPF

```
..
DOKU.MPF
FREIE_KONTUR.MPF
MEIN_ERSTES_PR
REF_XZ.MPF
```

Free NC memory: 3154944 Byte(s)

Figure 5-7 Saving an MDA program

MKS/WKS REL

The actual values for the MDA mode are displayed depending on the selected coordinate system. Use this softkey to switch between the two coordinate systems.

5.3.1 Teach-in

Functionality

You can use the "Teach In" function to create and change simple traversing blocks. You can transfer axis position values directly into a newly generated or changed part program record.

The axis positions are reached by traversing with the axis direction keys and transferred into the part program.

Operating sequence

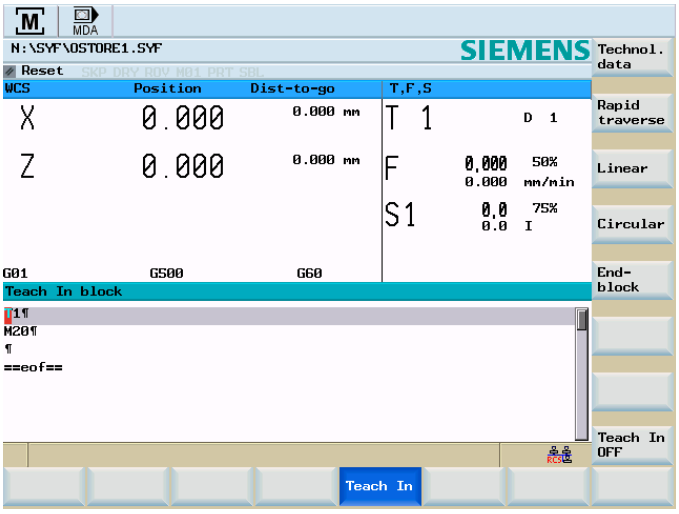
MDA

In the <POSITION> operating area, use the machine control panel to select <MDA> mode.

Teach In

Press the "Teach In" softkey.

In the "Teach In" submode, assume the following start screen:



```
T1¶
M20¶
¶
==eof==
```

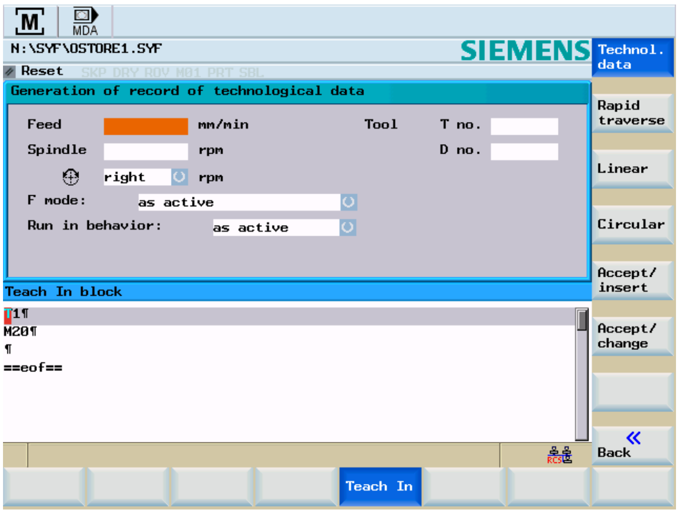
Figure 5-8 Main screen

General sequence

1. Use the arrow keys to select the program block that you want to edit or that is to have the new traversing block inserted in front of it.
2. Select the appropriate softkey.

Technol. data

– "Technological data"



N:\SYF\OSTORE1.SYF

Reset

Generation of record of technological data

Feed mm/min Tool T no.

Spindle rpm D no.

right rpm

F mode: as active

Run in behavior: as active

Teach In block

```
T1¶
M20¶
¶
==eof==
```

Figure 5-9 Technological data

Insert transfer

Change transfer

<< Back

Enter the appropriate technological data (e.g. feedrate: 1000).

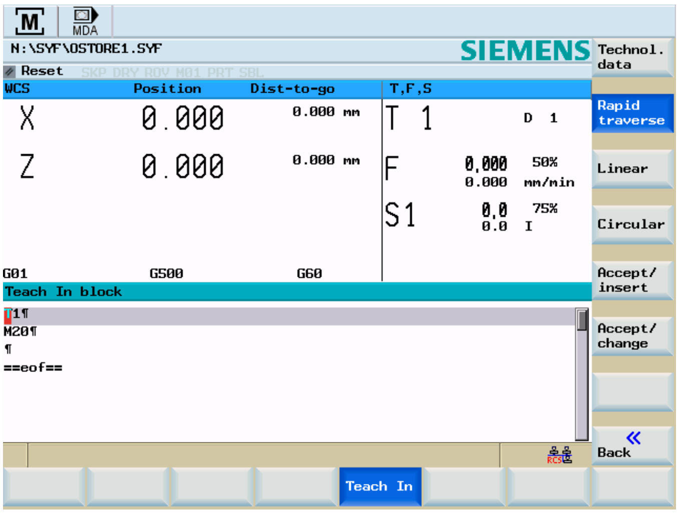
Click "Insert transfer" to add a new part program block. The new part program block will be added in front of the block selected with the cursor.

Click "Change transfer" to change the selected part program block.

Use "<<Back" to return to the "Teach In" start screen.

Rapid traverse

– "Rapid feed"



N:\SYF\OSTORE1.SYF

Reset

WCS	Position	Dist-to-go	T,F,S		
X	0.000	0.000 mm	T 1		D 1
Z	0.000	0.000 mm	F	0.000 0.000	50% mm/min
			S1	0.0 0.0	75% I

G01 G500 G60

Teach In block

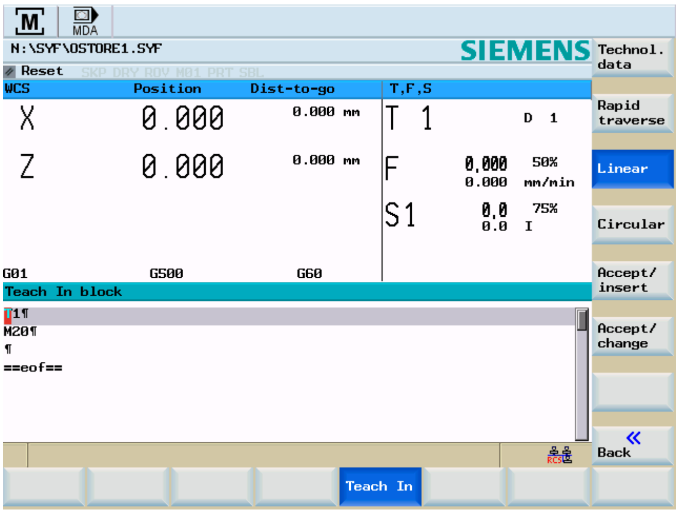
```
T1¶
M20¶
¶
==eof==
```

Figure 5-10 Rapid traverse

You traverse the axes and teach-in a rapid traverse block with the approached positions.

Linear

– "Linear"



N:\SYF\OSTORE1.SYF

Reset

WCS	Position	Dist-to-go	T,F,S		
X	0.000	0.000 mm	T 1		D 1
Z	0.000	0.000 mm	F	0.000 0.000	50% mm/min
			S1	0.0 0.0	75% I

G01 G500 G60

Teach In block

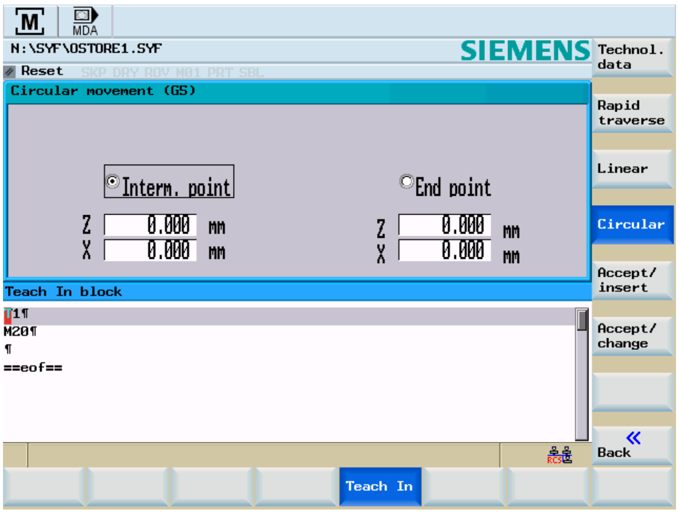
```
T1¶
M20¶
¶
==eof==
```

Figure 5-11 Linear

You traverse the axes and teach in a linear block with the approached positions.

Circular

– "Circular"



N:\SYF\OSTORE1.SYF

Reset

Circular movement (G5)

◉ Intern. point ○ End point

	Intern. point			End point	
Z	0.000	mm	Z	0.000	mm
X	0.000	mm	X	0.000	mm

Teach In block

```
T1¶
M20¶
¶
==eof==
```

Figure 5-12 Circular

You teach in an intermediate point and an end point for a circle.

Operation in the "Rapid traverse", "Linear" and "Circular" dialogs

+X

-Z

1. Use the axis keys to traverse the axes to the required position that you want to add/change in the part program.

Insert transfer

2. Click "Insert transfer" to add a new part program block. The new part program block will be added in front of the block selected with the cursor.

Change transfer

3. Click "Change transfer" to change the selected part program block.

<< Back

Use "<<Back" to return to the "Teach In" start screen.

Exit Teach In

Use "Exit Teach In" (see "Start screen") to leave the "Teach In" submode.

5.3.2 Face turning

Functionality

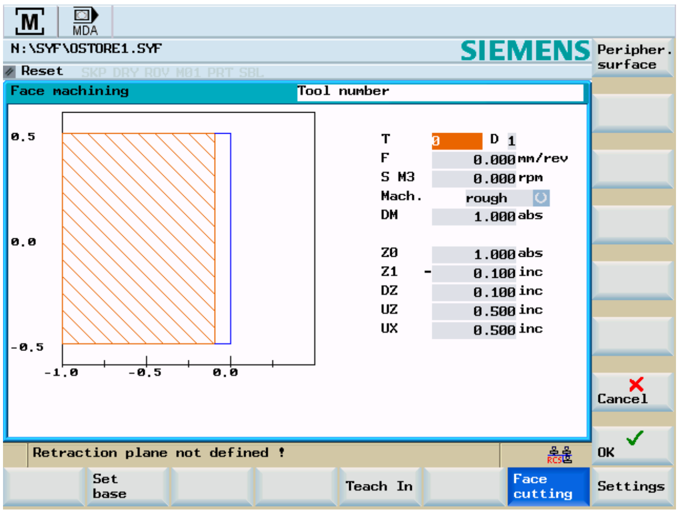
Use this function to prepare a blank for the subsequent machining without creating a special part program.

Operating sequence

MDA

Planned machining

Select the <MDA> mode and open the input screen "Machining of end face" by pressing the "Face turning" softkey.



N:\SYF\OSTORE1.SYF

Reset SKP DRY ROV M01 PRT SBL

Face machining | Tool number

```
T      0        D 1
F        0.000 mm/rev
S M3     0.000 rpm
Mach.    rough
DM       1.000 abs

Z0       1.000 abs
Z1   -   0.100 inc
DZ       0.100 inc
UZ       0.500 inc
UX       0.500 inc
```

Retraction plane not defined !

Figure 5-13 Machining of end face

OK ✓

After you have filled out the screen form completely and pressed "OK", the function will create a part program.

The input screen will be closed and the HMI will return to the machine start screen.

CYCLE START

The part program can be started with <NC START>.

In the machine start screen you can observe the program progress.

Note

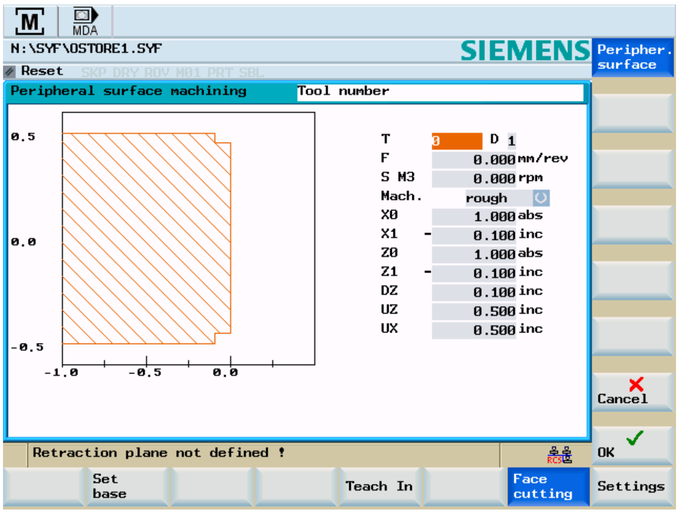
The retraction plane and the safety clearance must be defined beforehand in the "Settings" menu.

Table 5- 3 Description of the parameters in the "Machining end face" working window

Parameter	Explanation
Tool T	Input of the tool to be used The tool is loaded prior to machining. To this end, the function calls a working cycle performing all steps required. This cycle is provided by the machine manufacturer.
Feedrate F	Input of the path feedrate, in mm/min or mm/rev.
Spindle S rpm	Input of the spindle speed
Contact:	Definition of the surface quality. You can select between roughing and finishing.
Diameter DM	Input of the blank diameter of the part
Z0 Blank dimension	Input of the Z position
Z1 Cutting dimension	Cutting dimension, incremental
DZ Cutting dimension	Input of the cutting length in the z-direction. Specified in increments relative to the workpiece edge.
UZ Max. infeed	Stock allowance in the Z direction
UX Max. infeed	Stock allowance in the X direction

Peripheral surface

"Peripheral surface"



Peripheral surface machining

Tool number

```
T        0         D 1
F            0.000 mm/rev
S M3         0.000 rpm
Mach.        rough
X0           1.000 abs
X1     -     0.100 inc
Z0           1.000 abs
Z1     -     0.100 inc
DZ           0.100 inc
UZ           0.500 inc
UX           0.500 inc
```

Retraction plane not defined !

Figure 5-14 Machining of peripheral surface

Table 5- 4 Description of the parameters in the "Machining peripheral surface" working window

Parameter	Explanation
Tool T	Input of the tool to be used The tool is loaded prior to machining. To this end, the function calls a working cycle performing all steps required. This cycle is provided by the machine manufacturer.
Feedrate F	Input of the path feedrate, in mm/min or mm/rev.
Spindle S rpm	Input of the spindle speed
Contact:	Use this softkey to define the surface quality. You can select between roughing and finishing.
X0 Blank diameter	Input of the blank diameter
X1 Cutting length	Cutting length, incremental, in the X direction
Z0 Position	Input of the position of the workpiece edge in the Z direction
Z1 Cutting length	Cutting length, incremental, in the Z direction
DZ Max. infeed	Input of the infeed dimension in the X direction
UZ	Input field for the stock allowance when roughing
UX	Stock allowance

Automatic mode

6

6.1 AUTOMATIC mode

Menu tree

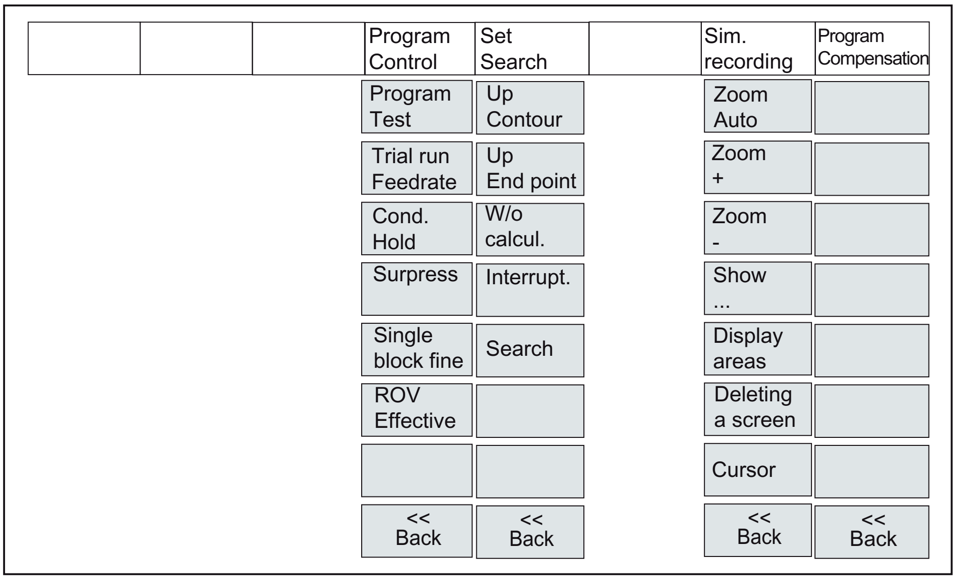


Figure 6-1 "AUTOMATIC" menu tree

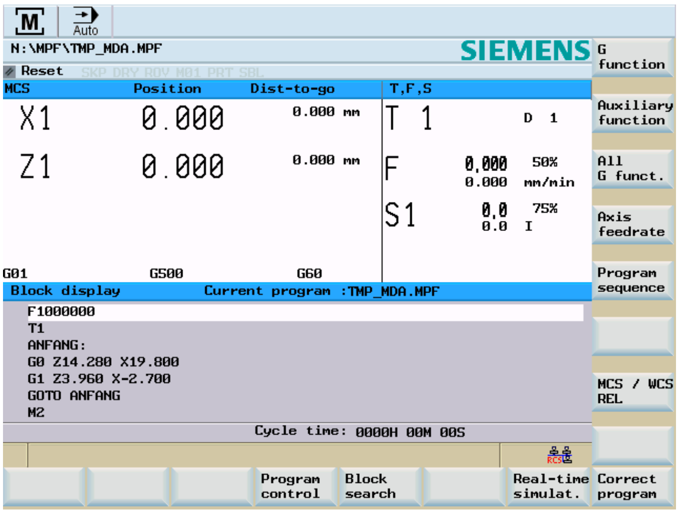
Preconditions

The machine is set up for the AUTOMATIC mode according to the specifications of the machine manufacturer.

Operating sequence

Select AUTOMATIC mode by pressing the <AUTOMATIC> key on the machine control panel.

The "AUTOMATIC" start screen appears, displaying the position, feedrate, spindle, and tool values, as well as the block currently active.



Block display Current program :TMP_MDA.MPF

```
F1000000
T1
ANFANG:
G0 Z14.280 X19.800
G1 Z3.960 X-2.700
GOTO ANFANG
M2
```

Cycle time: 0000H 00M 00S

Figure 6-2 "AUTOMATIC" start screen

Parameter

Table 6- 1 Description of the parameters in the working window

Parameter	Explanation
MCS X Z	Displays the existing axes in the MCS or WCS
+X -Z	If you traverse an axis in the positive (+) or negative () direction, a plus or minus sign will appear in the relevant field. If the axis is already in the required position, no sign is displayed.
Position mm	These fields display the current position of the axes in the MCS or WCS.
Distance-to-go	These fields display the current position of the axes in the MCS or WCS.
G function	Displays important G functions
Spindle S r.p.m.	Displays the actual value and the setpoint of the spindle speed
Feed F mm/min or mm/rev	Displays the path feedrate actual value and setpoint

Parameter	Explanation
Tool	Displays the currently active tool with the current edge number (T..., D...).
Current block	The block display displays seven subsequent blocks of the currently active part program. The display of one block is limited to the width of the window. If several blocks are executed quickly one after the other, it is recommended to switch to the "Program progress" window. To switch back to the seven-block display, use the "Program sequence" softkey.

Note

If a second spindle is integrated into the system, the work spindle will be displayed using a smaller font. The window will always display the data of only one spindle.

The control system displays the spindle data according to the following aspects:

The master spindle is displayed:

- Idle,
- at spindle start
- with both spindles active

The work spindle is displayed:

- when starting the work spindle

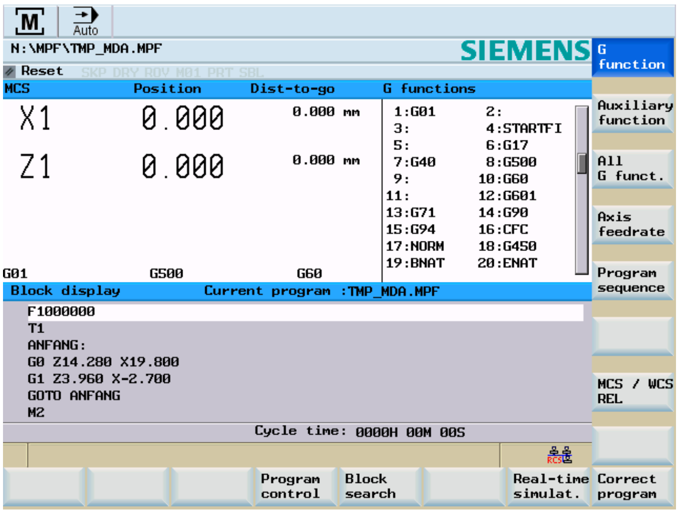
The power bar applies to the spindle currently active. With both master spindle and work spindle active, the master spindle performance bar is displayed.

Softkeys

G-function

Opens the "G functions" window to display all G functions currently active.

The window displays all G functions currently active whereby each G function is assigned to a group and has a fixed position in the window.



```
N:\MPF\TMP_MDA.MPF
Reset   SKP DRY ROV M01 PRT SBL
MCS        Position      Dist-to-go      G functions
X1         0.000         0.000 mm        1:G01       2:
                                         3:          4:STARTFI
Z1         0.000         0.000 mm        5:          6:G17
                                         7:G40       8:G500
                                         9:         10:G60
                                        11:         12:G601
                                        13:G71      14:G90
                                        15:G94      16:CFC
                                        17:NORM     18:G450
                                        19:BNAT     20:ENAT
G01             G500          G60
Block display          Current program :TMP_MDA.MPF
  F1000000
  T1
  ANFANG:
  G0 Z14.280 X19.800
  G1 Z3.960 X-2.700
  GOTO ANFANG
  M2
                    Cycle time: 0000H 00M 00S
```

Figure 6-3 "G functions" window

Use the <Page Up> or <Page Down> keys to display additional G functions.

Auxiliary function

This window displays the auxiliary and M functions currently active.
Selecting the softkey repeatedly will close the window.

All G-functions

All G functions are displayed (also see chapter "Programming").

Axis feedrate

Use this softkey to display the "Axis feedrate" window.
Pressing the softkey repeatedly will close the window.

MKS/WKS REL

Switches the axis value display between the machine, workpiece and relative coordinate systems.

Program control

The program control softkeys are displayed (e.g. "Skip block", "Program test").

- "Program test": If "Program test" is selected, the output of setpoints to axes and spindles is disabled. The set point display "simulates" the traverse movements.
- "Dry run feedrate": If you select this softkey, all traversing motions will be performed with the feedrate setpoint specified via the "Dry run feed" setting data. Instead of the programmed motion commands, the dry run feedrate will be effective.
- "Conditional stop": When this function is active, processing of the program is stopped at every block in which miscellaneous function M01 is programmed.
- "Skip": Program blocks that are identified with a slash in front of the block number are skipped when the program starts (e.g. "/N100").
- "Single block, fine": If this function is active, the part program blocks are executed as follows: Each block is decoded separately, and a stop is performed at each block; an exception are only the thread blocks without dry run feedrate. In such blocks, a stop is only performed at the end of the current thread block. "Single Block fine" can only be selected in the RESET state.
- "ROV active": The feedrate override switch will also act on the rapid traverse override.

<< Back

Use this softkey to close the screen.

Block search

Use the "Block search" function to go to the desired program location.

To contour

Forward block search with calculation
During the block search, the same calculations are carried out as during normal program operation, but the axes do not move.

To
end point

Forward block search with calculation to the block end point
During the block search, the same calculations are carried out as during normal program operation, but the axes do not move.

Without
calculat.

Block search without calculation
During the block search, no calculation is carried out.

Interr.
point

The cursor is placed on the main program block of the interrupt point.

Find

The "Find" softkey provides the functions "Find line", "Find text" etc.

Simultaneous
recording

It is possible to simultaneously record when the part program is executed (see Chapter "Simultaneously recording").

Correct
program

It is possible to correct a program passage. Any changes will be stored immediately.

6.2 Select and start a part program

Functionality

Before starting the program, make sure that both the control system and the machine are set up. Observe the relevant safety notes of the machine manufacturer.

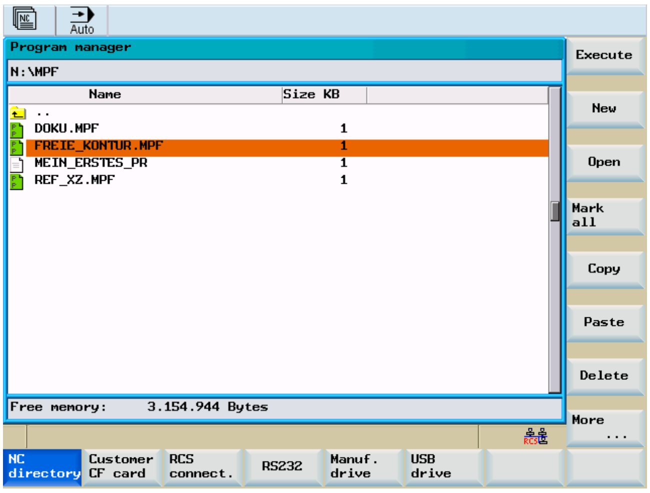
Operating sequence

AUTO

Select AUTOMATIC mode by pressing the <AUTOMATIC> key on the machine control panel.

PROGRAM MANAGER

The Program Manager is opened. Use the "NC directory" (default selection), "Customer CF card" or "USB drive" softkeys to go to the the appropriate directories.



Program manager

N:\MPF

Name	Size KB
..	
DOKU.MPF	1
FREIE_KONTUR.MPF	1
MEIN_ERSTES_PR	1
REF_XZ.MPF	1

Free memory: 3.154.944 Bytes

Figure 6-4 The "Program Manager" start screen

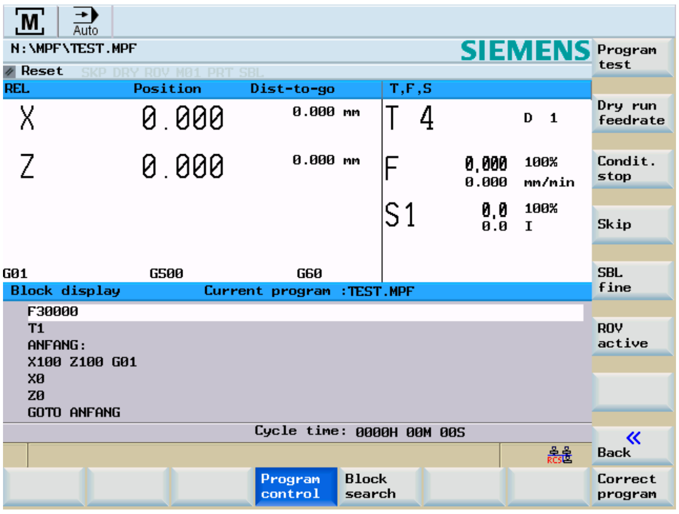
Place the cursor bar on the desired program.

Exe-cute

Use the "Execute" softkey to select the program to be executed (also see Execute from external (Page 98)). The name of the selected program will appear in the "Program name" screen line.

Program control

If desired, here you can specify how you want the program to be executed.



```
N:\MPF\TEST.MPF
Reset   SKP DRY ROV M01 PRT SBL
REL        Position     Dist-to-go      T,F,S
X          0.000        0.000 mm        T 4            D  1
Z          0.000        0.000 mm        F      0.000   100%
                                               0.000   mm/min
                                        S1       0.0   100%
                                                 0.0   I
G01             G500            G60
Block display          Current program :TEST.MPF
   F30000
   T1
   ANFANG:
   X100 Z100 G01
   X0
   Z0
   GOTO ANFANG
                        Cycle time: 0000H 00M 00S
```

Figure 6-5 Program control

CYCLE START

Press <NC START> to start execution of the part program.

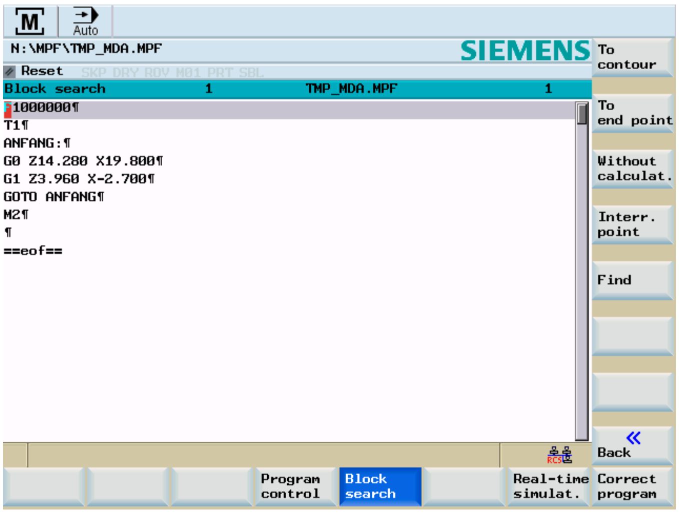
6.3 Block search

Operating sequence

Requirement: The desired program has already been selected and the control system is in the RESET state.

Block search

The block search function provides advance of the program to the required block in the part program. The search target is set by positioning the cursor bar directly on the required block in the part program.



N:\MPF\TMP_MDA.MPF

Reset

Block search 1 TMP_MDA.MPF 1

```
F1000000¶
T1¶
ANFANG:¶
G0 Z14.280 X19.800¶
G1 Z3.960 X-2.700¶
GOTO ANFANG¶
M2¶
¶
==eof==
```

Figure 6-6 Block search

To contour

Block search to block start

To end point

Block search to block end

Without calculat.

Block search without calculation

Interr. point

The interruption location is loaded.

Find

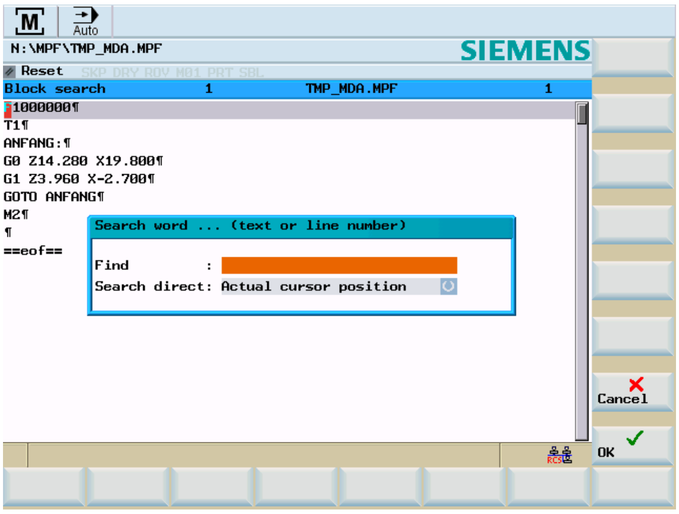
Use this softkey to perform the block search by entering a term you are looking for.

The block search varies as follows:

- Numerical value (e. g. "100")

  The system jumps to the corresponding line in the program.
- Alphanumeric text (e. g. "N100")

  The systems jumps to the line with the corresponding text.



N:\MPF\TMP_MDA.MPF

Reset SKP DRY ROV M01 PRT SBL

Block search 1 TMP_MDA.MPF 1

```
F1000000¶
T1¶
ANFANG:¶
G0 Z14.280 X19.800¶
G1 Z3.960 X-2.700¶
GOTO ANFANG¶
M2¶
¶
==eof==
```

Search word ... (text or line number)

Find :

Search direct: Actual cursor position

Figure 6-7 Enter search term

A toggle field is provided to define from which position you will search for the term.

Search result

The required block is displayed in the "Current block" window.

Note

For "Execute externally", no block search is possible.

6.4 Simultaneous recording

Operating sequence

CYCLE START

You have selected a part program to be executed and have pressed <NC START>.

Simultaneous recording

Execution of the part program is simultaneously recorded on the HMI using the "Simultaneous recording" function.

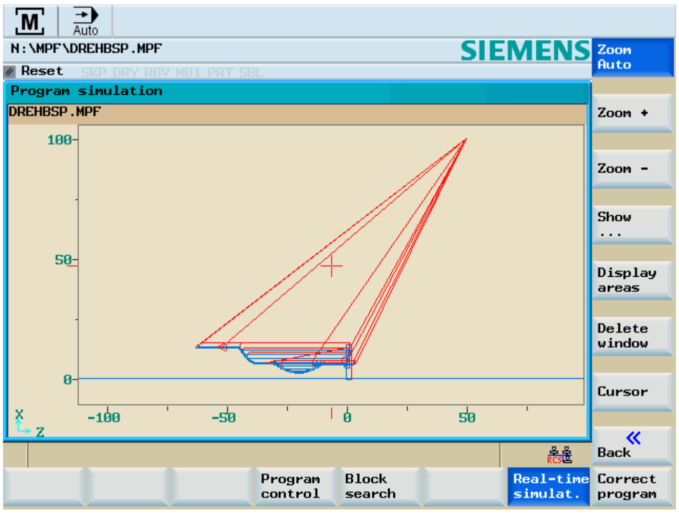


Figure 6-8 "Simultaneous recording" start screen

You can influence how the simultaneous recording function is displayed on the HMI using the following vertical softkeys:

- "Zoom Auto"
- "Zoom +"
- "Zoom -"
- "Show ..."
  - "All G17 blocks"
  - "All G18 blocks"
  - "All G19 blocks"
- "Display areas"

  See the following page for a description.

- "Delete window"
- "Cursor"
  - "Set cursor"
  - "Cursor fine", "Cursor coarse", "Cursor very coarse"

    When the cursor keys are pressed, the cross hair moves in small, average or large steps.

<< Back

Exit the "Simultaneous recording" function.

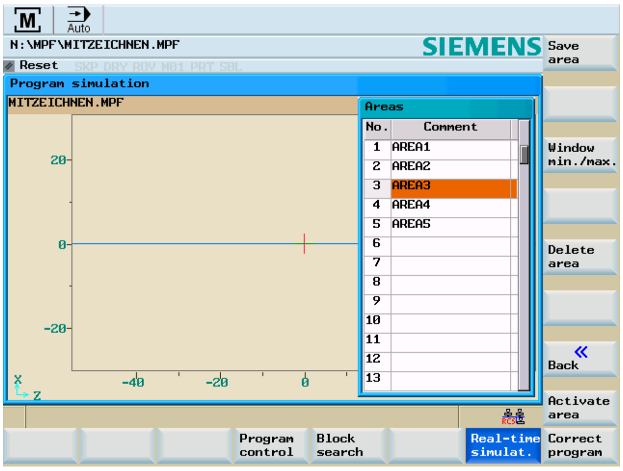
"Display areas"

Display area

Window, min/max

Using the "Display areas" function, you have the possibility of saving a previously selected area from the simulation display.

The menu for the display area can be selected using the "Window min/max" function.



N:\MPF\MITZEICHNEN.MPF

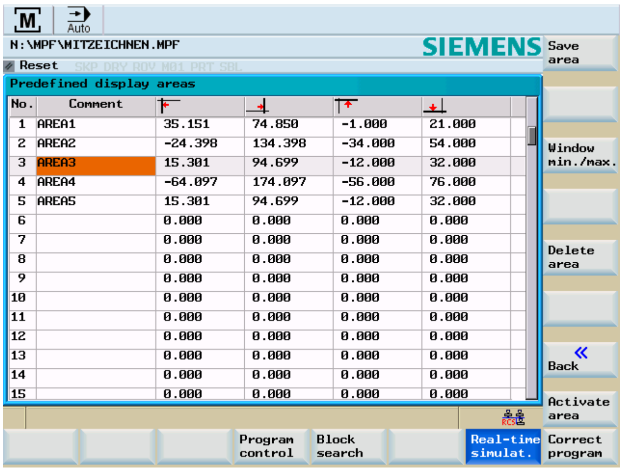
Program simulation

MITZEICHNEN.MPF

Areas

No.	Comment
1	AREA1
2	AREA2
3	AREA3
4	AREA4
5	AREA5
6	
7	
8	
9	
10	
11	
12	
13	

Figure 6-9 Display area "Window min"



N:\MPF\MITZEICHNEN.MPF

Predefined display areas

No.	Comment				
1	AREA1	35.151	74.850	-1.000	21.000
2	AREA2	-24.398	134.398	-34.000	54.000
3	AREA3	15.301	94.699	-12.000	32.000
4	AREA4	-64.097	174.097	-56.000	76.000
5	AREA5	15.301	94.699	-12.000	32.000
6		0.000	0.000	0.000	0.000
7		0.000	0.000	0.000	0.000
8		0.000	0.000	0.000	0.000
9		0.000	0.000	0.000	0.000
10		0.000	0.000	0.000	0.000
11		0.000	0.000	0.000	0.000
12		0.000	0.000	0.000	0.000
13		0.000	0.000	0.000	0.000
14		0.000	0.000	0.000	0.000
15		0.000	0.000	0.000	0.000

Figure 6-10 Display area "Window max"

Operating sequence to set and save the display area

1. You have selected an area in the simulation view.

Display area

2. Press the "Display areas" function.

Window, min/max

3. Press the "Window min/max" so that a maximum display can be see according to the screen "Display areas "Window max".
4. In the "Comment field", you can assign a name to the area.

INPUT

5. Complete the entry with <Input>.

Save area

6. Press "Save area".

Activating or deleting an area

Display area

You have selected a display area.

▲▼

Using the cursor keys, select the area that you wish to either activate or delete.

Activate area

Delete area

Press "Activate area" or "Delete area".

6.5 Stop / cancel a part program

Operating sequence

CYCLE STOP

With <NC STOP> the execution of a part program is interrupted.
The interrupted machining can be continued with <NC START>.

RESET

Use <RESET> to abort the program currently running.
By pressing <NC START> once again, the aborted program is restarted and executed from the beginning.

6.6 Reapproach after cancellation

After a program cancellation (RESET), you can retract the tool from the contour in manual mode (JOG).

Operating sequence

AUTO	Select mode <AUTOMATIC> mode.
Block search	Opening the "Block search" window for loading the interruption point.
Interr. point	The interruption point is loaded.
To contour	The block search to the interruption point will start. An adjustment to the start position of the interrupted block will be carried out.
CYCLE START	Press <NC START> to continue machining.

6.7 Repositioning after interruption

After interrupting the program (<NC STOP>), you can retract the tool from the contour in manual mode (JOG). The control saves the coordinates of the point of interruption. The distances traversed are displayed.

Operating sequence

AUTO

Select <AUTOMATIC> mode.

CYCLE START

Press <NC START> to continue machining.

⚠ CAUTION

When reapproaching the interruption point, **all axes will traverse at the same time.** Make sure that the traversing area is not obstructed.

6.8 Execute from external

Functionality

AUTO

PROGRAM MANAGER

In <AUTOMATIC> mode > <PROGRAM MANAGER> operating area, the following interfaces are available for external execution of programs:

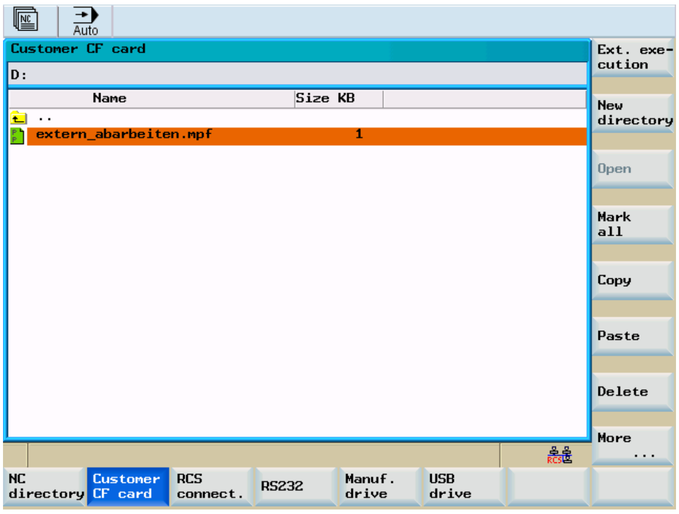
Customer CF card — Customer CompactFlash card

RCS connect. — RCS connection for external execution via network (only for SINUMERIK 802D sl pro)

Manufac-turer drive — Manufacturer's drive

USB drive — USB FlashDrive

Start in the following start screen of the Program Manager:



Name	Size KB
..	
extern_abarbeiten.mpf	1

Figure 6-11 The "Program Manager" start screen

Use vertical softkey "Ext. execution" to transmit the selected external program to the control system; to execute this program, press <NC START>.

While the contents of the buffer memory are being processed, the blocks are reloaded automatically.

Operating sequence, execution from customer CompactFlash Card or USB FlashDrive

Requirement: The control system is in the "Reset" state.

AUTO

Select the <AUTOMATIC> mode key .

PROGRAM MANAGER

Press the <PROGRAM MANAGER> key on the machine control panel.

Customer CF card

USB drive

Press the "Customer CF card" or "USB drive".
You can thus access the directories of the "Customer CF Card / USB FlashDrive".

Place the cursor bar on the desired program.

Ext. execution

Press "Ext. execution".

The program is transferred into the buffer memory and selected and displayed in the program selection automatically.

CYCLE START

Press the <NC START> key.

Machining starts. The program is reloaded continuously.

At the end of the program or in case of <RESET>, the program is automatically removed from the control system.

Note

For "Execute externally", **no** block search is possible.

Requirements for external execution via network

- The control system and the external programming device/PC are connected via Ethernet.
- The RCS tool is installed on the programming device/PC.

The following conditions are required on the the devices:

1. Control: (see "User Management")
   - Create an authorization for using the network using the following dialog:

     Operating area <SYSTEM> > "Service Display" > "Service Control" > "Service Network" > "Authorization" > "Create"
2. Control: (see "User log in - RCS log in")
   - Log in for the RCS connection using the following dialog:

     Operating area <SYSTEM> > vertical softkey "RCS log in" > "Log in"
3. Programming device/PC:
   - Start the RCS tool.

4. Programming device/PC:
   – Activate the drive/directory for network operation.
5. Programming device/PC:
   – Establish an Ethernet connection to the control.
6. Control: (see "Connecting / disconnecting a network drive")
   – Connect to the directory activated on the programming device/PC using the following dialog:

     Operating area <SYSTEM> > "Service Display" > "Service Control" > "Service Network" > > "Connect" > "RCS Network" (Select a free drive of the control > Enter the server name and and activated directory of the programming device/PC, for example: "\\123.456.789.0\External Program")

Operating sequences for external execution via network

AUTO	Select the <AUTOMATIC> mode key .
PROGRAM MANAGER	Press the <PROGRAM MANAGER> key on the machine control panel.
RCS connect.	Press "RCS connect."". You go to the directories of the PG/PC.
	Place the cursor bar on the desired program.
Ext. execution	Press "Ext. execution".
	The program is transferred into the buffer memory and selected and displayed in the program selection automatically.
CYCLE START	Press the <NC START> key.
	Machining starts. The program is reloaded continuously.
	At the end of the program or in case of <RESET>, the program is automatically removed from the control system.

Note

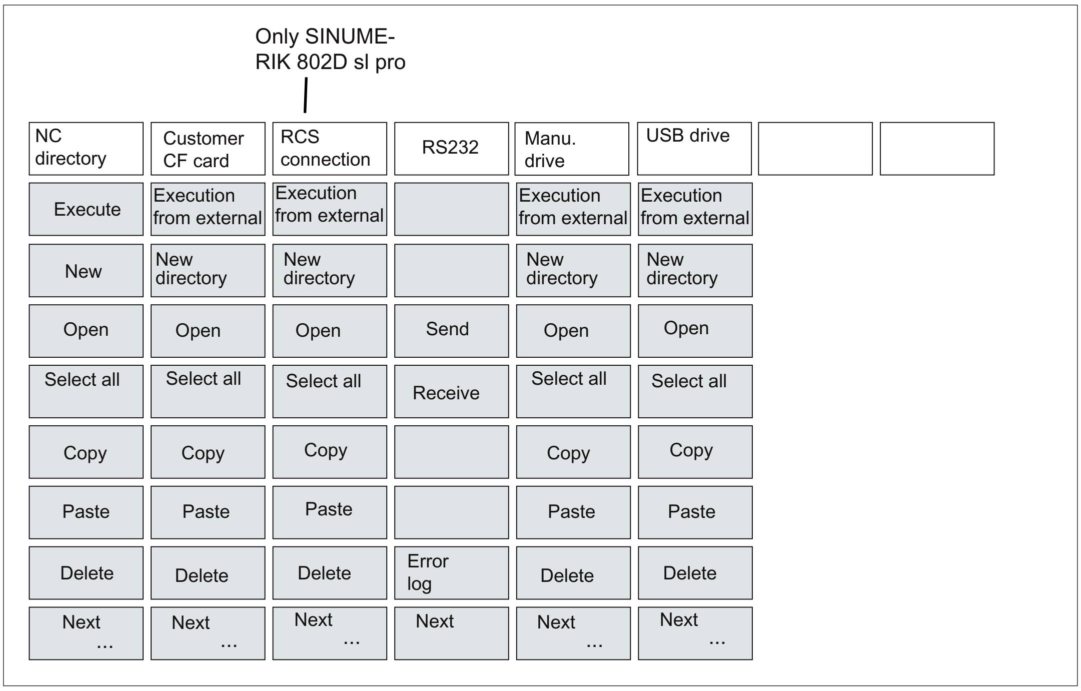
The program can only be executed. Program correction is not possible at the control.

Part Programming

7

7.1 Part programming overview

Menu tree



Only SINUME-RIK 802D sl pro

NC directory	Customer CF card	RCS connection	RS232	Manu. drive	USB drive		
Execute	Execution from external	Execution from external		Execution from external	Execution from external		
New	New directory	New directory		New directory	New directory		
Open	Open	Open	Send	Open	Open		
Select all	Select all	Select all	Receive	Select all	Select all		
Copy	Copy	Copy		Copy	Copy		
Paste	Paste	Paste		Paste	Paste		
Delete	Delete	Delete	Error log	Delete	Delete		
Next ...	Next ...	Next ...	Next	Next ...	Next ...		

Figure 7-1 "Program Manager" menu tree

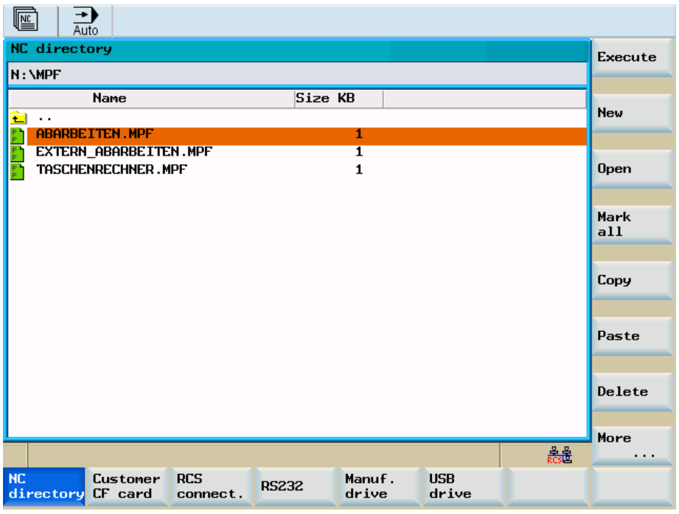
Functionality

The PROGRAM MANAGER operating area is the management area for workpiece programs in the control system. In this area, programs can be created, opened for modification, selected for execution, copied, and inserted.

Operating sequence

PROGRAM MANAGER

Press the <PROGRAM MANAGER> key to open the program directory.



NC directory

N:\MPF

Name	Size KB
..	
ABARBEITEN.MPF	1
EXTERN_ABARBEITEN.MPF	1
TASCHENRECHNER.MPF	1

Figure 7-2 The "Program Manager" start screen

Use the cursor keys to navigate in the program directory. To find program names quickly, simply type the initial letter of the program name. The control system will automatically position the cursor on a program with matching characters.

Softkeys

NC directories

Use this softkey to display the directories of the NC.

Exe-cute

Use this softkey to select the program on which the cursor is placed for execution. The control system will switch to the position display. Use <NC START> to start this program.

New

Use the "New" softkey to create a new program.

Open

Use the "Open" softkey to open the file highlighted by the cursor for processing.

Mark all

Use this softkey to select all files for the subsequent operations. The selection can be canceled by pressing the softkey once more.

SELECT

Note

Selecting individual files:

Position the cursor on the corresponding file and press the <Select> key. The selected line will change its color. If you press the <Select> key once more, the selection is canceled.

Copy

This function will enter one or several files in a list of files (called 'clipboard') to be copied.

Paste

This function will paste files or directories from the clipboard to the current directory.

Delete

When selecting the "Delete" softkey, the file selected by the cursor is deleted after a confirmation warning. If several files have been selected, all these files will be deleted after a confirmation warning.

Use the "OK" softkey to execute the deletion request and "Abort" to discard.

More ...

Use this softkey to branch to further functions.

Renaming

A window opens where you can rename the program you have selected beforehand using the cursor.

After you have entered the new name, either press "OK" to confirm or "Abort" to cancel.

Preview window

This function opens a window displaying the first seven lines of a file if the cursor has been positioned on the program name for a certain time.

Find

A window opens up where you can enter a file name you are looking for.

After you have entered the name, either press "OK" to confirm or "Abort" to cancel.

Enables

A selected directory can be released for network operation.

Split window

The function splits the window on the HMI. You can use the <Tab> key to switch over between windows.

Proper-ties

The function gives information on the properties of the memory of the selected directory and of the selected file.

Error log

The function gives information in a logfile on the executed functions (e.g. copying a file) as well as on wrongly executed functions of the PROGRAM MANAGER. The logfile will be deleted after cold restart of the control.

Customer CF card

Selecting this softkey provides the functions required to read out / read in files via the customer CompactFlash card and the function "Program execution from external". When the function is selected, the directories of the customer CompactFlash card are displayed.

Ext. execution

Use this softkey to select the program on which the cursor is placed for execution. If the CF card is selected, the program is executed by the NC as an external program. This program must not contain any program calls of part programs which are not stored in the directory of the NC.

RCS connect.

This softkey is needed in connection with the work in the network. Additional information is provided in Chapter, network operation (only for SINUMERIK 802D sl pro).

RS232

The functions required for reading out/reading in files are provided via the RS232 interface.

Send

Use this function to transmit files from the clipboard to a PC connected to the RS232.

Receive

Load files via the RS232 interface.

For the settings of the interface, please refer to the "System" operating area. The part programs must be transmitted using the text format.

Error log

Error log

Manufac-turer drive

Selecting this softkey provides the functions required to read out / read in files via the manufacturer drive and the function "Program execution from external". When the function is selected, the directories of the manufacturer's drive are displayed.

USB drive

Selecting this softkey provides the functions required to read out / read in files via USB FlashDrive and the function "Program execution from external". When the function is selected, the directories of the USB FlashDrive are displayed.

7.2 Enter new program

Operating sequences

PROGRAM MANAGER

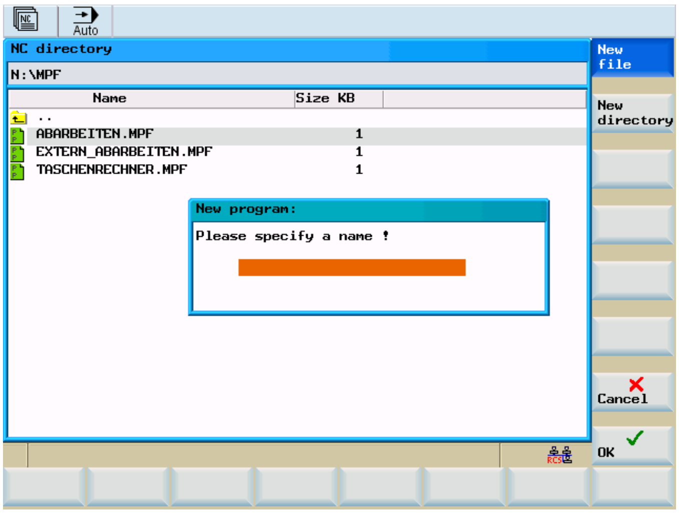
You have selected the PROGRAM MANAGER operating area.

NC directories

New

Use the "NC directory" softkeys to select the storage location for the new program.

Press "New". You have the choice of the following options:



NC directory

N:\MPF

Name	Size KB
..	
ABARBEITEN.MPF	1
EXTERN_ABARBEITEN.MPF	1
TASCHENRECHNER.MPF	1

New program:

Please specify a name !

Figure 7-3 New program

New directory

After presssing the softkey "New directory" a dialog window will open up for setting up a new file.

Enter a name and confirm with "OK."

New file

After presssing the softkey "New file" a dialog window will open up for setting up a new program file. in which you can enter the names of the new main programs and subprograms. The .MPF extension for main programs is entered automatically. The .SPF extension for subprograms must be entered along with the program name.

OK ✓

Conclude your entry with "OK". The new part program file will be created, and the editor window is opened automatically.

✗ Abort

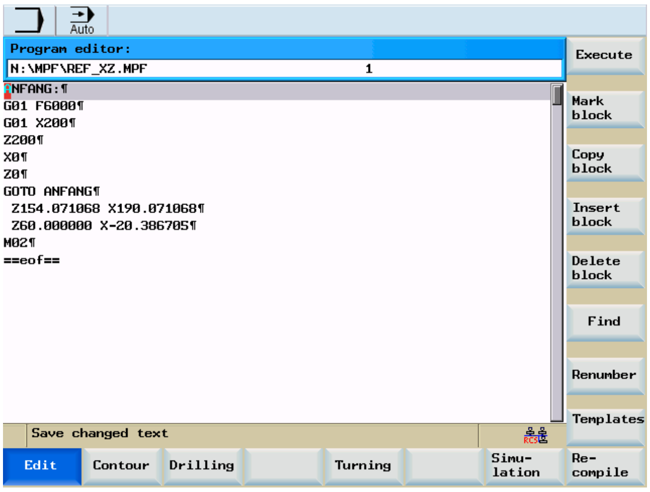
Use "Cancel" to cancel the creation of the program. the window is closed.

7.3 Edit the part program

Functionality

A part program or sections of a part program can only be edited if currently not being executed.

Any modifications to the part program are stored immediately.



Program editor:
N:\MPF\REF_XZ.MPF 1

```
ANFANG:¶
G01 F6000¶
G01 X200¶
Z200¶
X0¶
Z0¶
GOTO ANFANG¶
 Z154.071068 X190.071068¶
 Z60.000000 X-20.386705¶
M02¶
==eof==
```

Save changed text

Figure 7-4 "Program editor" start screen

Menu tree

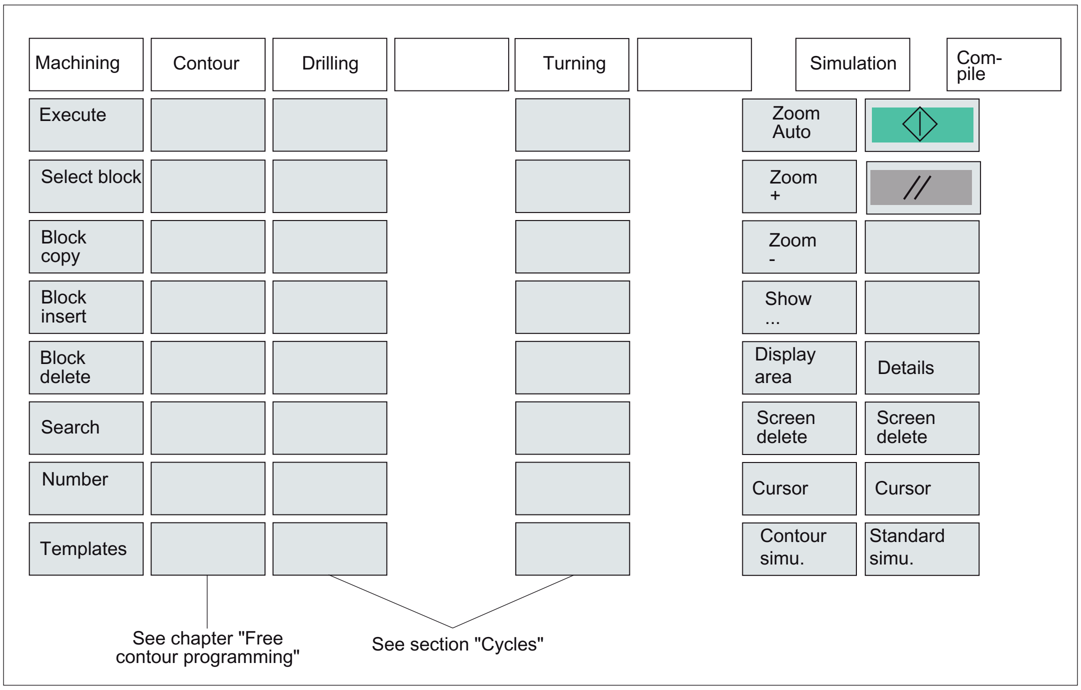


Figure 7-5 "Program" menu tree

Operating sequence

PROGRAM MANAGER

Select the program to be edited in the PROGRAM MANAGER operating area

Open

and press "Open".

The program is opened and displayed for editing. Additional softkey functions are provided.

Program changes are automatically applied.

Softkeys

Machining

Use this function to edit text segments.

Execute

Use this softkey to execute the selected file.

Mark block

Use this softkey to select a text segment up to the current cursor position (alternatively: <CTRL+B>)

Copy block

Use this softkey to copy a selected block to the clipboard (alternatively: <CTRL+C>)

Insert block

Use this softkey to paste a text from the clipboard at the current cursor position (alternatively: <CTRL+V>)

Delete block

Use this softkey to delete a selected text (alternatively: <CTRL+X>)

Find

Use the "Find" softkey to search for a string in the program file displayed.

Type the term you are looking for in the input line and use the "OK" softkey to start the search.

Use "Abort" to close the dialog box without starting the search process.

Renumber

Use this softkey to replace the block numbers from the current cursor position up to the program end.

Templates

With this softkey function, parts of programs that can be added to other programs are stored.

Contour

For information about free contour programming see section "Free Contour Programming"

Bore

See chapter "Cycles"

Turning

See chapter "Cycles"

Note

For information about the "Milling" softkey see chapter "Cycles" (with the options "Transmit" and "Tracyl")

Simulation

The simulation is described in chapter "Simulation".

Recomp.

For recompilation, position the cursor on the cycle calling line in the program.

Using the "Recompile" function, the cycle screen form is re-called for a cycle that was parameterized using a softkey function (e.g. "Drill" > "Drilling centering" -> CYCLE81). This function decodes the cycle name and prepares the screen form with the relevant parameters. If there are any parameters beyond the range of validity, the function will automatically use the default values. After closing the screen form, the original parameter block is replaced by the corrected block.

Note

Only automatically generated blocks can be recompiled.

7.4 Simulation

Functionality

By using broken-line graphics, the programmed tool path can be traced.

Operating sequence

PROGRAM

Using the operating area key <PROGRAM> or by opening a part program the displayed part program can be simulated.

Simu-
lation

The start screen is opened.

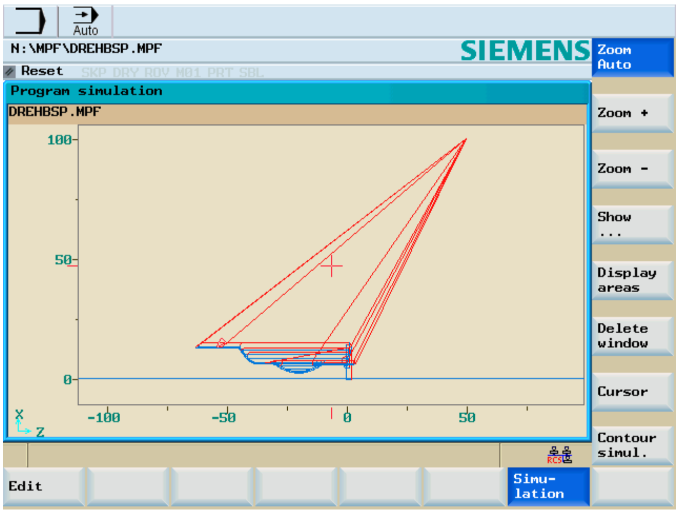


Figure 7-6 Standard simulation

The simulation of the part program can be reproduced on the HMI using the following two functions:

- Standard simulation

  The execution of the part program is simulated on the HMI by considering the axis feedrates. In the case of more extensive NC programs, the simulation may, therefore, take more time.

- Contour simulation

  Execution of the part program is simulated on the HMI. The simulation is based on pure calculations and is, therefore, faster in the case of more extensive NC programs.

Standard simulation

Standard simulation

CYCLE START

The execution of the part program is simulated with this function on the HMI, taking into consideration the axis feedrates.

Press <NC START> to start the standard simulation for the selected part program.

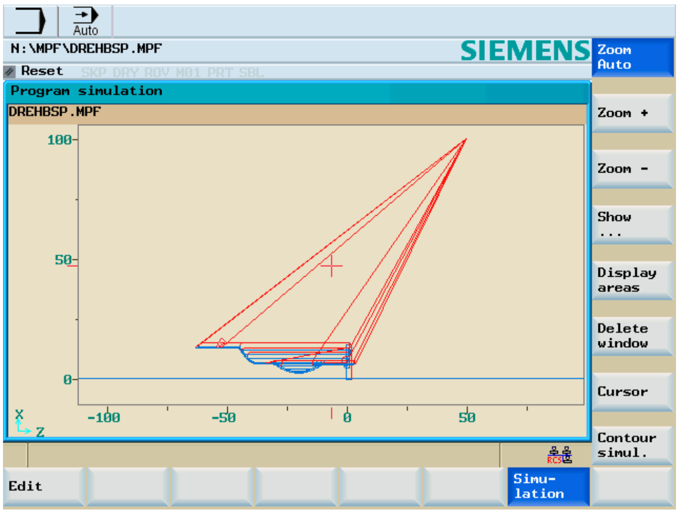


Figure 7-7 Standard simulation

Softkeys for the standard simulation

You can influence how the standard simulation is displayed on the HMI using the following vertical softkeys:

- "Zoom Auto"
- "Zoom +"
- "Zoom -"
- "Show ..."
  - "All G17 blocks"
  - "All G18 blocks"
  - "All G19 blocks"
- "Display areas"

  Defines the simulation displayed on the HMI to specific areas (see Chapter "Simultaneous recording" (Page 92)).
- "Delete window"
- "Cursor"
  - "Set cursor"
  - "Cursor fine", "Cursor coarse", "Cursor very coarse"

    When the cursor keys are pressed, the cross hair moves in small, average or large steps.

Contour simulation

Switches to "Contour simulation".

Contour simulation

Contour simulation

Execution of the part program is simulated on the HMI using this function.

The machine does not move.

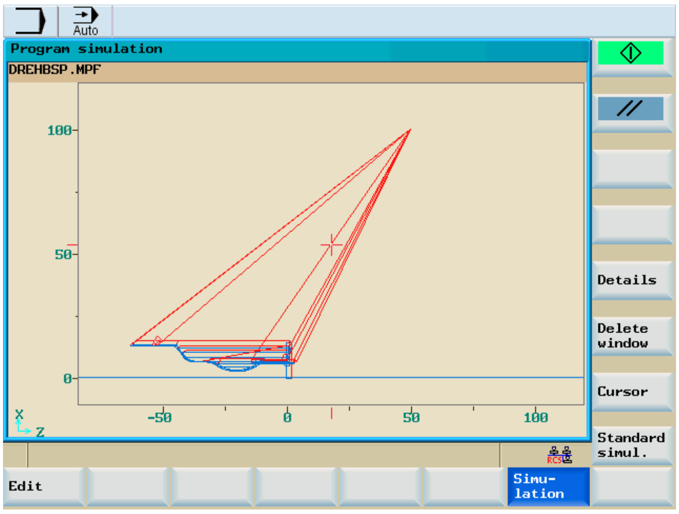


Figure 7-8 Contour simulation

Softkeys for the contour simulation

◇

The selected part program is started for the contour simulation.

//

Initiates a RESET at the HMI.

Details

The following functions can be activated:

- "Zoom Auto"
- "Zoom +"
- "Zoom -"
- "Display areas"

  Defines the simulation displayed on the HMI to specific areas (see Chapter "Simultaneous recording").

Delete window

Use this softkey to delete the visible screen.

Cursor

The type of motion of the cross-hair can be set using the following functions:

- "Set cursor"
- "Cursor fine", "Cursor coarse", "Cursor very coarse"

  When the cursor keys are pressed, the cross hair moves in small, average or large steps.

Standard simulation

Switches into the "standard simulation".

See also

Simultaneous recording (Page 92)

7.5 Calculate contour elements

When you open the calculator, softkeys for editing contour elements appear. You enter the values for the contour element in the respective input screens. Press "Accept" to perform the calculation.

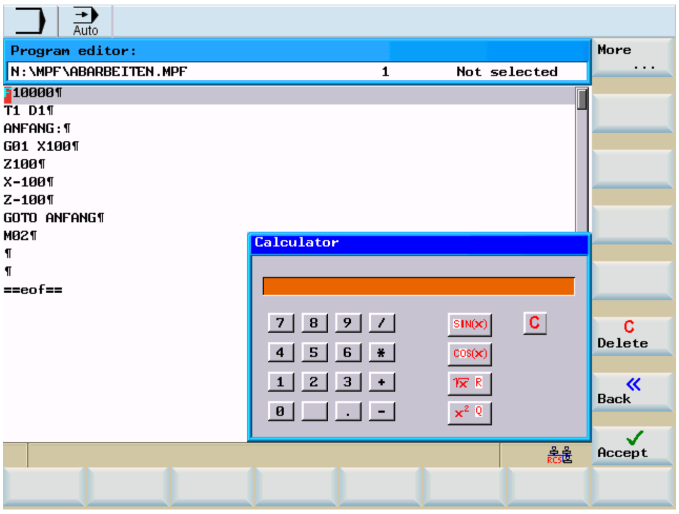
SHIFT

=

The <SHIFT> and <=> or <CTRL> and <A> key combination activates the calculator (Page 487).

Note

The <CTRL> and <A> key combination opens the calculator in the part program editor.



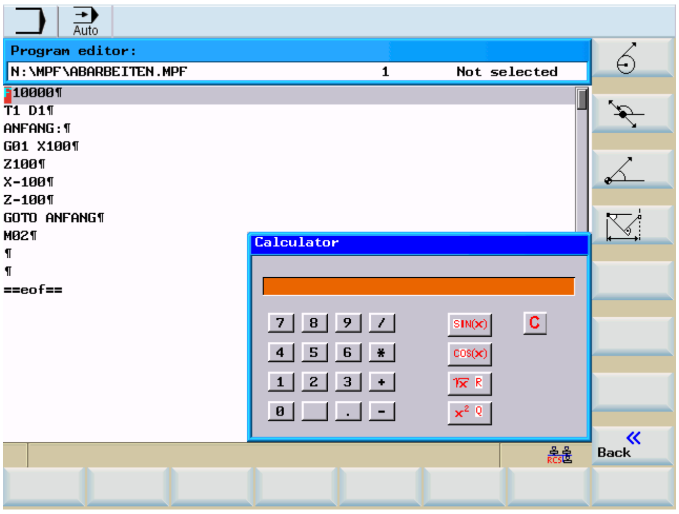
N:\MPF\ABARBEITEN.MPF 1 Not selected

```
F10000¶
T1 D1¶
ANFANG:¶
G01 X100¶
Z100¶
X-100¶
Z-100¶
GOTO ANFANG¶
M02¶
¶
¶
==eof==
```

Figure 7-9 Pocket calculator

More ...

The functions for editing contour elements are available via "Next ...".



Program editor:
N:\MPF\ABARBEITEN.MPF 1 Not selected

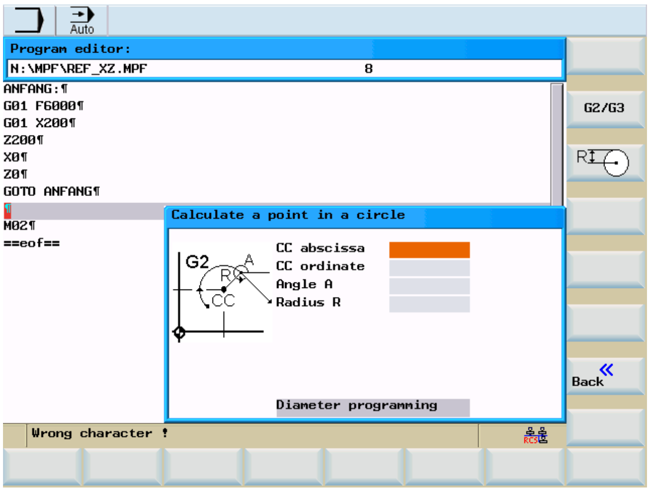
```
F10000¶
T1 D1¶
ANFANG:¶
G01 X100¶
Z100¶
X-100¶
Z-100¶
GOTO ANFANG¶
M02¶
¶
¶
==eof==
```

Calculator

Figure 7-10 Pocket calculator > Next ...

Softkeys

This function is used to calculate a point on a circle. The point results from the angle of the tangent created, as well as from the radius and the direction of rotation of the circle.



Program editor:
N:\MPF\REF_XZ.MPF 8

```
ANFANG:¶
G01 F6000¶
G01 X200¶
Z200¶
X0¶
Z0¶
GOTO ANFANG¶
¶
M02¶
==eof==
```

Calculate a point in a circle

CC abscissa
CC ordinate
Angle A
Radius R

Diameter programming

Wrong character !

Figure 7-11 Calculation: Point on circle

Enter the circle center, the angle of the tangent and the circle radius.

G2/G3

Use the "G2 / G3" softkey to define the direction of rotation of the circle.

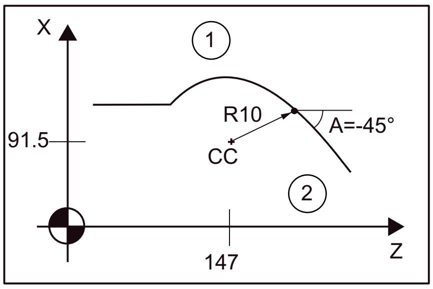
Either radius or diameter programming is slected using the softkey.

✓ Accept

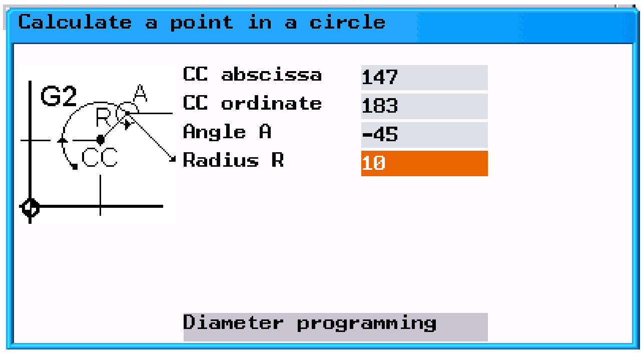
This softkey is displayed if the required parameters were entered.

Use this softkey to calculate the abscissa and ordinate values. The abscissa is the first axis, and the ordinate is the second axis of the plane. The abscissa value is copied into the input field from which the calculator function has been called, and the value of the ordinate is copied into the next following input field. If the function has been called from the part program editor, the coordinates are saved with the axis names of the selected basic plane.

Example: Calculating the point of intersection between the circle sector ① and the straight line ② in plane G18.



Given: Radius : 10
Circle center point CC: Z=147 X=183 (diameter progr.)
Connection angle for straight lines: -45°



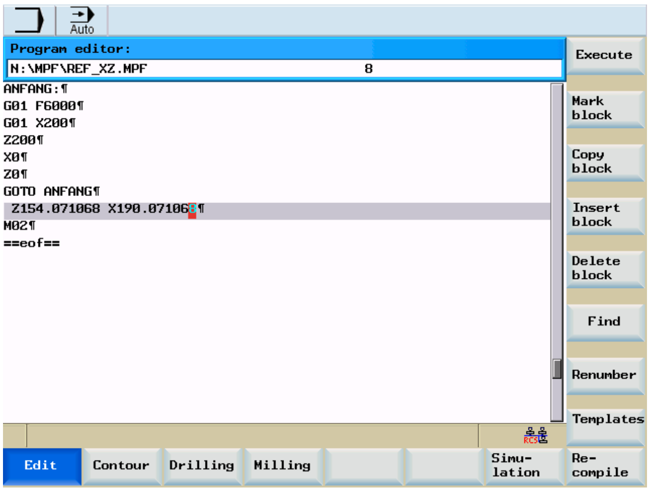
Calculate a point in a circle

CC abscissa	147
CC ordinate	183
Angle A	-45
Radius R	10

Diameter programming

Figure 7-12 Screen form

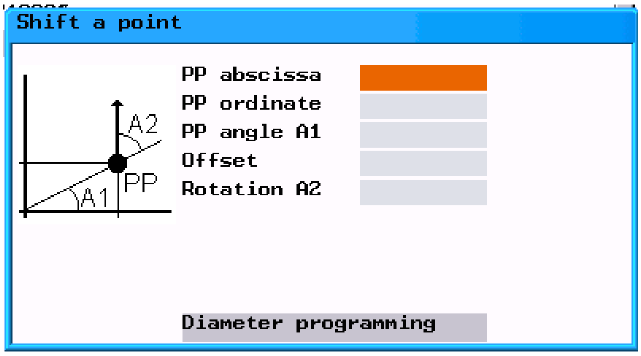
Result: Z = 154.071
X = 190.071



```
Program editor:
N:\MPF\REF_XZ.MPF                    8
ANFANG:¶
G01 F6000¶
G01 X200¶
Z200¶
X0¶
Z0¶
GOTO ANFANG¶
 Z154.071068 X190.071068¶
M02¶
==eof==
```

Figure 7-13 Programming result

This function calculates the Cartesian coordinates of a point in the plane, which is to be connected to a point (PP) on a straight line. For calculation, the distance between the points and the slope angle (A2) of the new straight line to be created with reference to the slope (A1) of the given straight line must be known.



Shift a point

PP abscissa
PP ordinate
PP angle A1
Offset
Rotation A2

Diameter programming

Figure 7-14 Calculation: Point in the plane

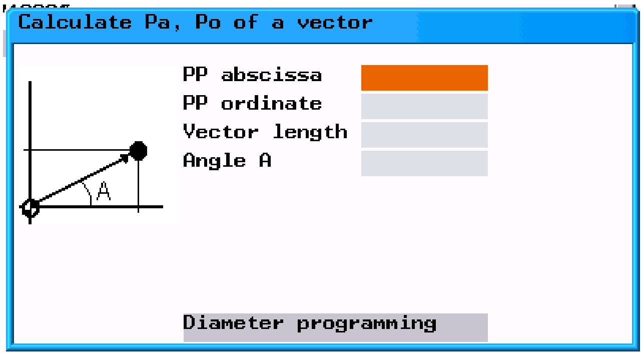
Enter the following coordinates or angles:

- the coordinates of the given point (PP)
- the slope angle of the straight line (A1)
- the distance of the new point with reference to PP
- the slope angle of the connecting straight line (A2) with reference to A1

✓ Accept

Use this softkey to calculate the Cartesian coordinates which are subsequently copied into two input fields following one after another. Copy the abscissa value into the input field from which you have called the calculator function. Copy the ordinate value into the next input field.
If the function has been called from the part program editor, the coordinates are saved with the axis names of the base plane.

This function converts the given polar coordinates into Cartesian coordinates.



Calculate Pa, Po of a vector

PP abscissa
PP ordinate
Vector length
Angle A

Diameter programming

Figure 7-15 Converting polar coordinates to Cartesian coordinates

Enter the reference point, the vector length and the slope angle.

✓ Accept

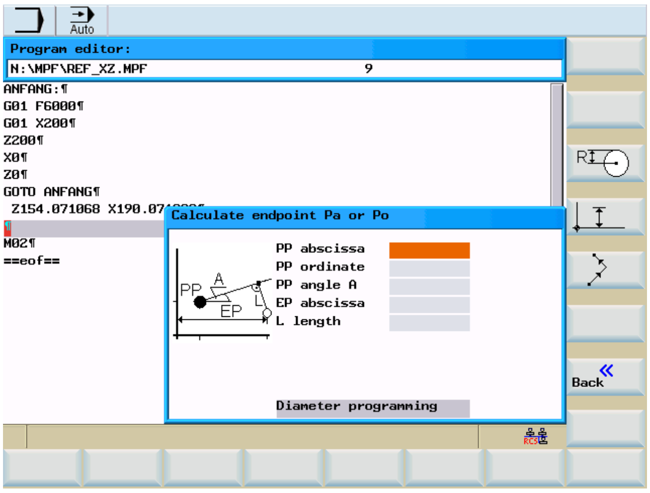
Use this softkey to calculate the Cartesian coordinates which are subsequently copied into two input fields following one after another. Copy the abscissa value into the input field from which you have called the calculator function. Copy the ordinate value into the next input field.
If the function has been called from the part program editor, the coordinates are saved with the axis names of the base plane.

Use this function to calculate the missing end point of the straight line/straight line contour section whereby the second straight line stands vertically on the first straight line.

The following values of the straight line are known:

Straight line 1: Starting point and slope angle

Straight line 2: Length and one end point in the Cartesian coordinate system



Program editor:
N:\MPF\REF_XZ.MPF 9

```
ANFANG:¶
G01 F6000¶
G01 X200¶
Z200¶
X0¶
Z0¶
GOTO ANFANG¶
 Z154.071068 X190.07
¶
M02¶
==eof==
```

Calculate endpoint Pa or Po

PP abscissa
PP ordinate
PP angle A
EP abscissa
L length

Diameter programming

Figure 7-16 Calculation: Missing end point

This function is used to select the given coordinate of the end point.

The ordinate value or the abscissa value is given.

The second straight line is rotated clockwise or

counter-clockwise by 90 degrees against the first straight line.

✓ Accept

The missing end point is calculated. Copy the abscissa value into the input field from which you have called the calculator function. Copy the ordinate value into the next input field. If the function has been called from the part program editor, the coordinates are saved with the axis names of the base plane.

Example

The following drawing must be supplemented by the value of the center circle point in order to be able to calculate the point of intersection between the circle sector of the straight lines.

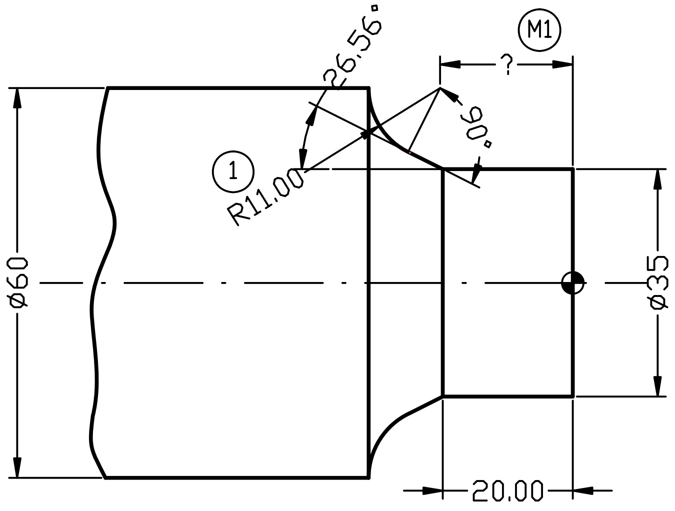


Figure 7-17 Calculation of M1

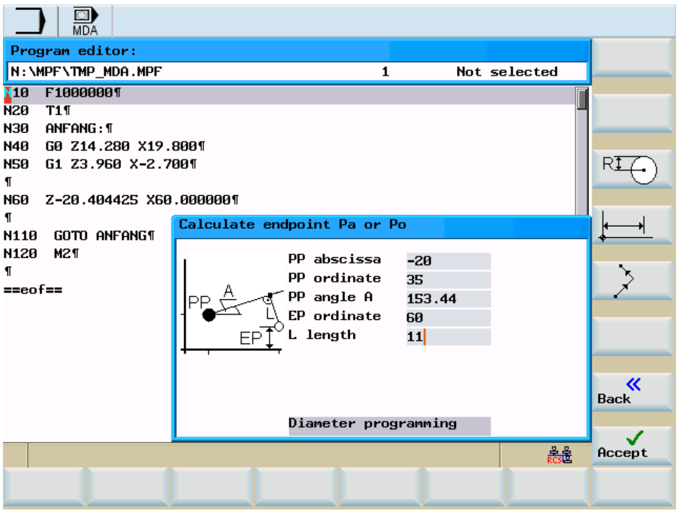
The missing center point coordinate is calculated using the calculator function, as the radius at the tangential transition is perpendicular to the straight line.

The radius is located at an angle of 90° clockwise to the straight-line defined by the angle.

Use this softkey to select the appropriate direction of rotation.

Use the softkey to define the given end point.

Enter the coordinates of the pole, the slope angle of the straight line, the ordinate angle of the end point and the circle radius as the length.



Program editor:

N:\MPF\TMP_MDA.MPF 1 Not selected

```
N10  F1000000¶
N20  T1¶
N30  ANFANG:¶
N40  G0 Z14.280 X19.800¶
N50  G1 Z3.960 X-2.700¶
¶
N60  Z-20.404425 X60.000000¶
¶
N110  GOTO ANFANG¶
N120  M2¶
¶
==eof==
```

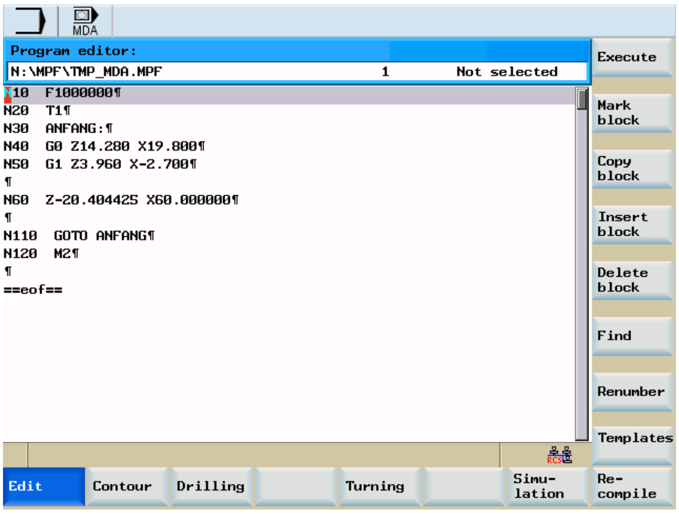
Calculate endpoint Pa or Po

PP abscissa	-20
PP ordinate	35
PP angle A	153.44
EP ordinate	60
L length	11

Diameter programming

Figure 7-18 Parameters for the example

Result: Z = -20.404425
X = 60



Program editor:

N:\MPF\TMP_MDA.MPF 1 Not selected

```
N10  F1000000¶
N20  T1¶
N30  ANFANG:¶
N40  G0 Z14.280 X19.800¶
N50  G1 Z3.960 X-2.700¶
¶
N60  Z-20.404425 X60.000000¶
¶
N110  GOTO ANFANG¶
N120  M2¶
¶
==eof==
```

Figure 7-19 Result in block N60

7.6 Free contour programming

Functionality

Free contour programming is a support tool for the editor. The contour programming function enables you to create simple and complex contours.

A contour editor (FKE) calculates any missing parameters for you as soon as they can be obtained from other parameters. You can link together contour elements. Additional contour transition elements are also available.

The programmed contours are transferred to the edited part program.

Technology

The contour calculator for turning technology provides the following functions for this purpose:

- Toggling between radius/diameter programming (DIAMON, DIAMOF, DIAM90)
- Chamfer/radius at the start and end of the contour
- Undercuts as transition elements between two axially parallel straight lines, where one runs horizontally and the other vertically (Form E, Form F, thread undercut acc. to DIN, general undercut)

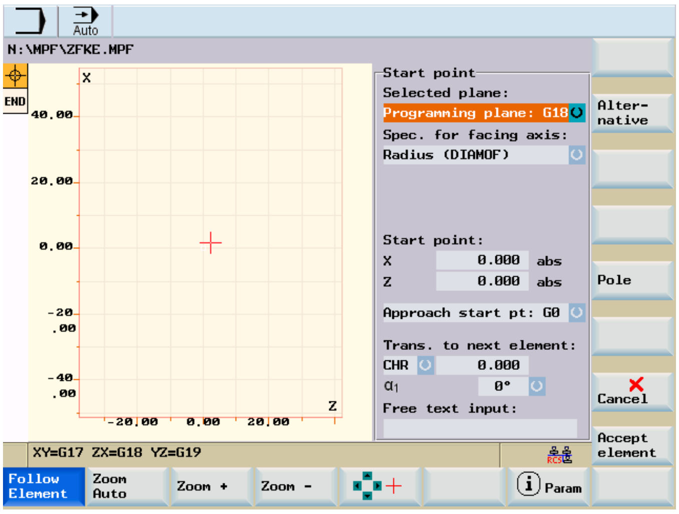
Start screen of the contour editor (FKE)

PROGRAM MANAGER

You have opened a part program in the <PROGRAM MANAGER> operating area.

Contour

Select the contour editor using the horizontal "Contour" softkey.



N:\MPF\ZFKE.MPF

Start point
Selected plane:
Programming plane: G18
Spec. for facing axis:
Radius (DIAMOF)

Start point:
X 0.000 abs
Z 0.000 abs

Approach start pt: G0

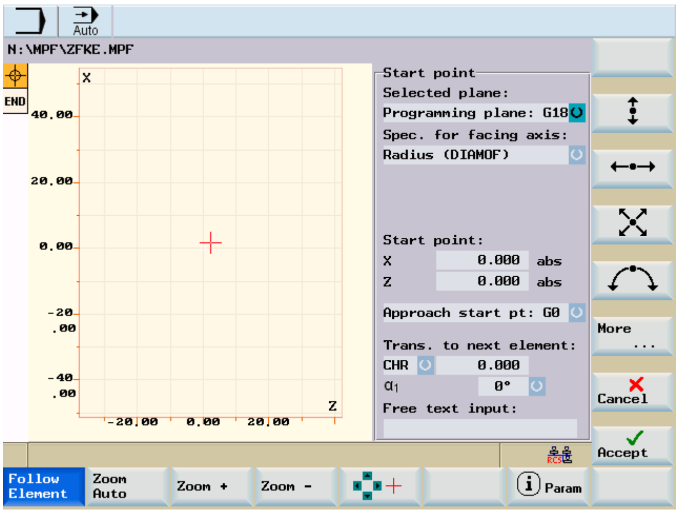
Trans. to next element:
CHR 0.000
α1 0°
Free text input:

XY=G17 ZX=G18 YZ=G19

Figure 7-20 Defining a starting point

Initially, you define a contour starting point (see Chapter "Defining a starting point (Page 124)").

The contour is then programmed step-by-step (see Chapter "Programming example turning (Page 143)").



N:\MPF\ZFKE.MPF

Start point
Selected plane:
Programming plane: G18
Spec. for facing axis:
Radius (DIAMOF)

Start point:
X 0.000 abs
Z 0.000 abs

Approach start pt: G0

Trans. to next element:
CHR 0.000
α_1 0°
Free text input:

Figure 7-21 Edit contour elements

Softkeys for the contour elements

The following are contour elements:

- Starting point

- Straight line in the vertical direction (transverse)
- Straight line in the horizontal direction (longitudinal)

- Oblique straight line

- Circular arc

A pole is a theoretical contour element. Straight lines and circular arcs can also be defined by polar coordinates in reference to a pole.

Further notes

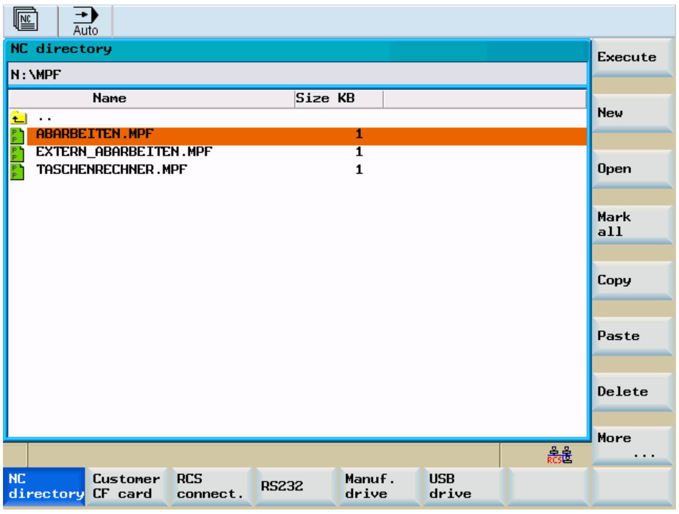
1. The valid geometry axes are determined and used in the part program.
2. For the contour allowance you must also specify the side to which the allowance applies (e.g. "right" or "left").

7.6.1 Program a contour

Operating sequences

The sequence of operations for programming the contour for a turned part in a part program are as follows:

1. Select softkey "NC directory" in the "Program Manager" operating area.
2. Select a directory with the cursor keys, e.g. "MPF Main programs" (see screenshot below).



NC directory

N:\MPF

Name	Size KB
..	
ABARBEITEN.MPF	1
EXTERN_ABARBEITEN.MPF	1
TASCHENRECHNER.MPF	1

Figure 7-22 The "Program Manager" start screen

3. Press the <Input> key to open the directory.

   You can edit an existing part program by selecting softkey "Open", or create a new program.

New OK ✓

4. To create a new part program, select softkey "New", enter a name and confirm with "OK". You are now in the ASCII editor.

Contour

5. Press softkey "Contour".

   The input screen for "Define a start point" is displayed.

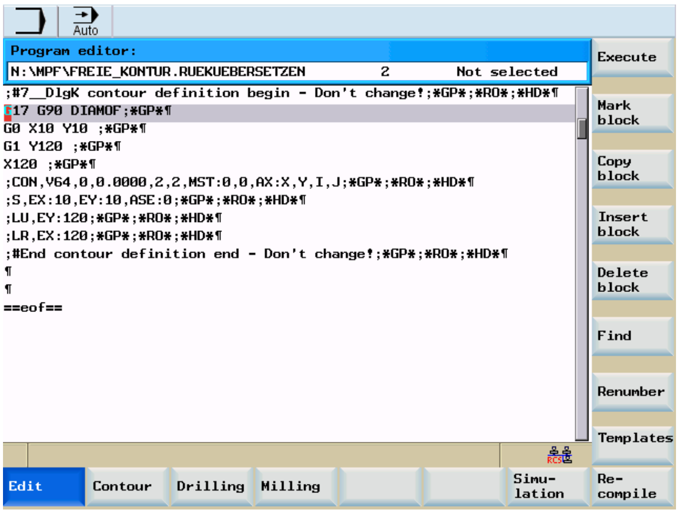
   You will find a guide to defining the start point in the section "Define a start point".

Recompile

Recompile

If you have programmed a contour with the "Contour" function, you can edit this existing contour again from the part program editor using the "Recompile" softkey. You are in the part program editor.

1. Position the editor cursor in a command line of the contour program.



```
N:\MPF\FREIE_KONTUR.RUEKUEBERSETZEN          2          Not selected
;#7__DlgK contour definition begin - Don't change!;*GP*;*RO*;*HD*¶
G17 G90 DIAMOF;*GP*¶
G0 X10 Y10 ;*GP*¶
G1 Y120 ;*GP*¶
X120 ;*GP*¶
;CON,V64,0,0.0000,2,2,MST:0,0,AX:X,Y,I,J;*GP*;*RO*;*HD*¶
;S,EX:10,EY:10,ASE:0;*GP*;*RO*;*HD*¶
;LU,EY:120;*GP*;*RO*;*HD*¶
;LR,EX:120;*GP*;*RO*;*HD*¶
;#End contour definition end - Don't change!;*GP*;*RO*;*HD*¶
¶
¶
==eof==
```

Figure 7-23 Recompile

2. Press the "Recompile" softkey.

   The user interface switches from the start screen of the part program editor into the start screen of the free contour programming.

   The programmed contour is displayed and can be edited.

NOTICE

The contour, that was parameterized using the "Contour" softkey function, is called again using the "Recompile" function. This function decodes the parameterized contour and prepares the screen form with the relevant parameters.

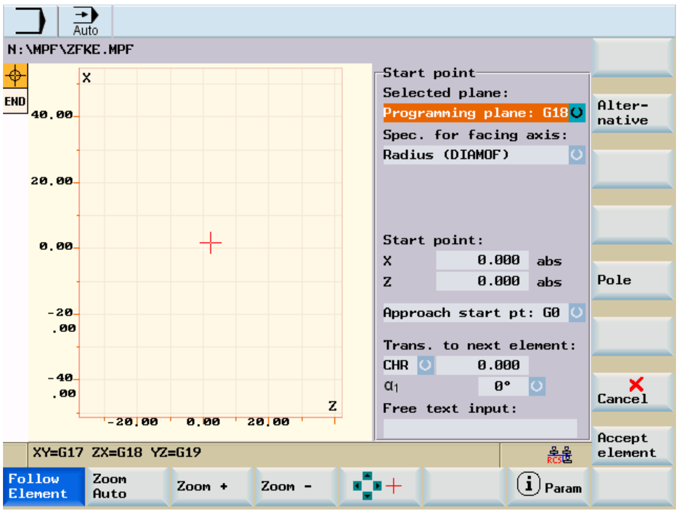
When recompiling, only the contour elements that were generated using the "Contour" function are created again. In addition, only the texts that were added using the "Free text input" input field are recompiled. Any changes you made directly in the program text are lost. However, you can subsequently insert and edit user-defined texts, which will not be lost.

7.6.2 Define a start point

Operating sequences

When entering a contour, begin at a position which you already know and enter it as the starting point. The sequence of operations for defining the start point of a contour is as follows:

- You have opened a part program and selected softkey "Contour" to program a new contour. The input screen for specifying the start point of the contour is displayed (see screenshot below).



N:\MPF\ZFKE.MPF

Start point
Selected plane:
Programming plane: G18
Spec. for facing axis:
Radius (DIAMOF)

Start point:
X 0.000 abs
Z 0.000 abs

Approach start pt: G0

Trans. to next element:
CHR 0.000
α1 0°
Free text input:

XY=G17 ZX=G18 YZ=G19

Figure 7-24 Define a start point

Note

The input field with the input focus is indicated by the dark background color. Once the input is acknowledged with "Accept element" or "Abort", you can navigate around the contour chain (left of the screen) using the ↑, ↓ cursor keys. The current position in the chain is color-highlighted.

---

1. Select programming plane G18 for the turned part using softkey "Alternative" (or the "Select key") in the "Selected plane" input field.

   The default tool axis or programming plane (defined in the machine data) can be changed for machines with more than two geometry axes. The associated starting point axes are automatically adjusted.

   Note

   Together with defining the contour start point, a pole can also be defined for contour programming in polar coordinates. The pole can also be defined or redefined at a later time. The programming of the polar coordinates always refers to the pole that was defined last.

2. In the input field "Spec. for facing axis", use softkey "Alternative" (or the "Select key") to select e.g. "Diameter (DIAMON)".

1 ... 0

3. Enter the values for the starting point.

   The values must be specified as absolute dimensions (reference dimension).

4. Select the approach motion to the start point in input field "Approach start point" with softkey "Alternative" (or the "Select key").

   The approach motion can be changed from G0 (rapid traverse) to G1 (linear interpolation).

   Note

   If you have not yet programmed a feedrate in the part program, you can enter a specific feedrate in the "Free text input" field, e.g. `F100`.

Accept element

5. Press the "Accept element" softkey.

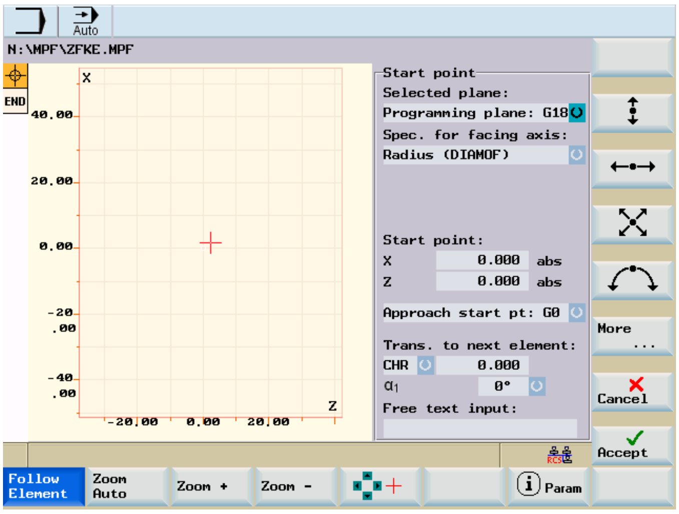
   The start point is saved.

   You can add the next element using softkey commands (see next section "Defining a contour element").

7.6.3 Softkeys and parameters

Functionality

Once you have defined the contour start point, you can begin programming the individual contour elements from the main screen shown below:



N:\MPF\ZFKE.MPF

Start point
Selected plane:
Programming plane: G18
Spec. for facing axis:
Radius (DIAMOF)

Start point:
X 0.000 abs
Z 0.000 abs

Approach start pt: G0

Trans. to next element:
CHR 0.000
α_1 0°
Free text input:

Figure 7-25 Define a contour element

Individual contour elements are programmed with the vertical softkeys. You assign the contour element parameters in the relevant input screen.

Vertical softkeys

The following contour elements are available for programming contours:

↕ Straight line in the vertical direction (X direction)

↔ Straight line in the horizontal direction (Z direction).

⤧ Oblique line in the X/Z direction. Enter the end point of the line using coordinates or an angle.

⌒ Arc with any direction of rotation.

More The "More" softkey in the basic plane of the contour programming accesses the "Pole" subscreen and the "Close contour" softkey.

Pole The pole can only be entered in absolute Cartesian coordinates. The "Pole" softkey is also present in the starting point screen. This enables the pole to be entered at the start of a contour, so that the first contour element can be entered in polar coordinates.

Close contour

The contour is closed by a straight line between the last entered contour point and the starting point.

Cancel

By selecting the "Abort" softkey you can return to the main screen, without transferring the last edited values to the system.

Accept

When you click the "Accept" softkey, you close the contour input screen and return to the ASCII editor.

Horizontal softkeys

You can zoom in or out of the graphic with the first four horizontal softkeys (e.g. "Zoom+").

Follow Element

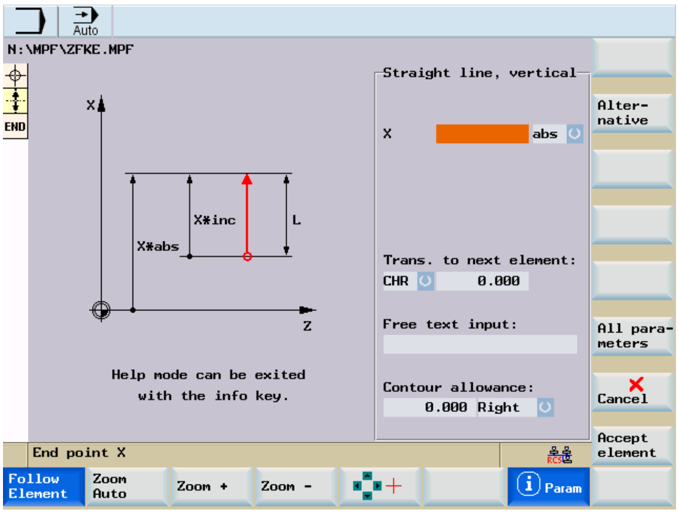
An element was selected using the cursor keys.

"Follower element" enlarges the image section of the selected element.

When you select this softkey, you can move the red cross-hair with the cursor keys and choose a picture detail to display. When you deactivate this softkey, the input focus is positioned in the contour chain again.

Param

If you press this softkey, help graphics are displayed in addition to the relevant parameter (see screenshot below). Press the softkey again to exit help mode.



N:\MPF\ZFKE.MPF

Straight line, vertical

X abs

Trans. to next element:
CHR 0.000

Free text input:

Contour allowance:
0.000 Right

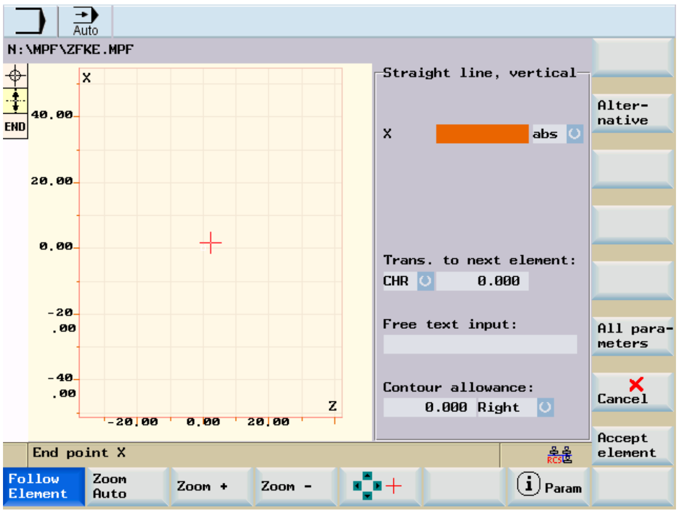
Help mode can be exited with the info key.

End point X

Figure 7-26 Help mode

Parameter

Beginning at the start point, enter the first contour element, e.g. a vertical straight line (see screenshot below).



N:\MPF\ZFKE.MPF

Straight line, vertical

X abs

Trans. to next element:
CHR 0.000

Free text input:

Contour allowance:
0.000 Right

End point X

Figure 7-27 Straight line in vertical direction

All
parameters

Select the "All parameters" softkey to display a selection list of all the parameters for the contour element.

If you leave any parameter input fields blank, the control assumes that you do not know the right values and attempts to calculate these from the settings of the other parameters.

The contour is always machined in the programmed direction.

Transition to next element

A transition element ("Trans. to next element") can be used whenever there is a point of intersection between two neighboring elements; this can be calculated from the input values.

You can choose to insert either a radius **RND,** a chamfer **CHR** or an undercut as the transition element between any two contour elements. The transition is always appended to the end of a contour element. You select transition elements in the parameter input screen for the relevant contour element.

You can access the "Undercut" transition element with softkey "Alternative" (or the "Selection key").

Radius or chamfer at the start or the end of a turning contour:

In simple turning contours a chamfer or radius must often be appended at the start and end of the contour.

A chamfer or radius terminates an axis-parallel contour section on the blank:

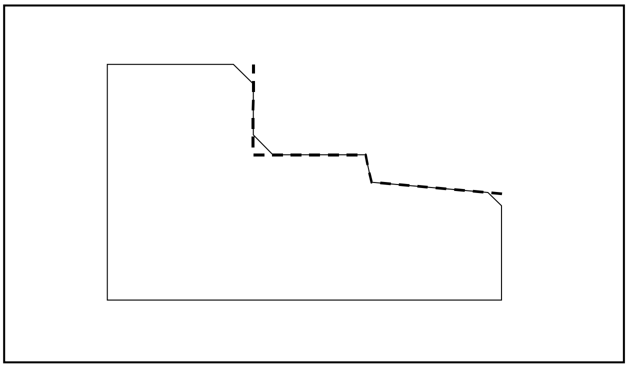


Figure 7-28 Contour with radius or chamfer

You select the direction of transition for the contour start in the starting point screen. You can choose between chamfer and radius. The value is defined in the same manner as for the transition elements.

In addition, four directions can be selected in a single selection field. You select the direction of the transition element for the contour end in the end screen. This selection is always proposed, even if predecessing elements were assigned no transition.

Free text input

In the "Free text input" field you can enter supplementary technological data, such as F1000 feedrate values, H or M functions.

Note

If comments are entered as text, they must always be started with a ";" semicolon.

Example: ; This is a test comment.

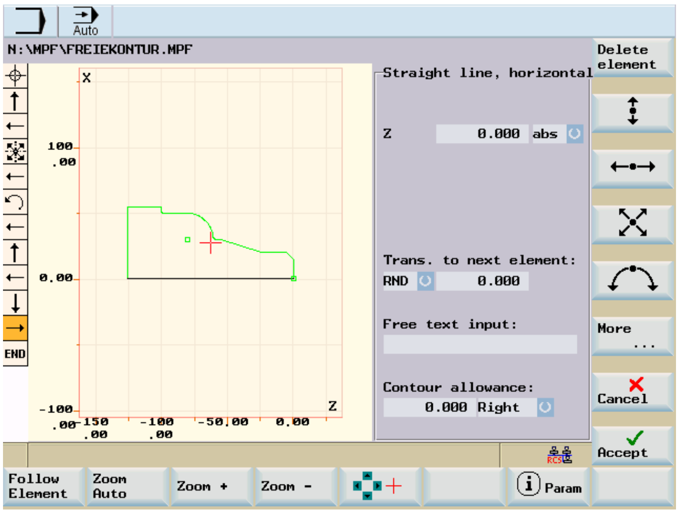
Contour allowance

Under "Contour allowance", you can specify a side-based parallel contour allowance. It is displayed as an allowance in the graphics window.

Contour chain on left in main screen

Once the input is acknowledged with "Accept element" or "Abort", you can navigate around the contour chain (left of the main screen) using the ↑, ↓ cursor keys. The current position in the chain is color-highlighted.

The elements of the contour and pole, if applicable, are displayed in the sequence in which they were programmed.


N:\MPF\FREIEKONTUR.MPF

Straight line, horizontal

Z 0.000 abs

Trans. to next element:
RND 0.000

Free text input:

Contour allowance:
0.000 Right

Figure 7-29 Edit a contour element

INPUT

You can select an existing contour element with the <Input> key and reassign its parameters.

A new contour element is inserted after the cursor when you select one of the contour elements on the vertical softkey bar; the input focus is then switched to the parameter input on the right of the graphic display. You can navigate around the contour chain again after selecting "Accept element" or "Abort".

Programming always continues after the element selected in the contour chain.

Delete element

You can delete the selected element from the chain by selecting softkey "Delete element".

7.6.4 Undercuts for turning technology

Supplementary conditions

The form E and F undercut and form DIN 76 and general thread undercut functions are only activated when turning technology is enabled.

Form E and F undercuts as well as thread undercuts are available only if level G18 is set. Undercuts are permitted only on contour edges of the rotational body, which run in the direction of the longitudinal axis (usually parallel to the Z axis). The longitudinal axis is identified by the machine data.

The machine data MD 20100 $MC_DIAMETER_AX_DEF for turning machines contains the name of the transverse axis (usually X). The other axis in G18 is the longitudinal axis (usually Z). If MD 20100 $MC_DIAMETER_AX_DEF does not contain a name or contains a name that does not conform to G18, there are no undercuts.

Undercuts only appear on corners between horizontal and vertical straight lines, including any straight lines, which are at 0°, 90°, 180° or 270°. A tolerance of ±3° is required here, so that conical threads are also possible (these undercuts do not meet the standard in this case).

Operator focus

When the operator focus is on "Trans. to next element", use the "Select key" or "Alternative" softkey to select Undercut.

When the focus is on the following field, the undercut form can be defined. The "Select key" or the "Alternative" softkey can be used to select the following options:

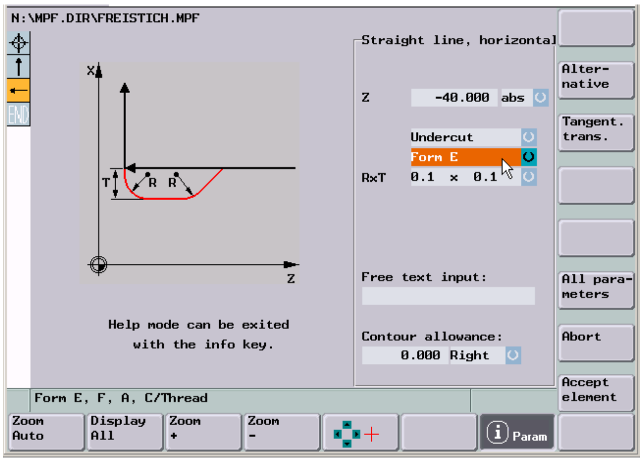
- Form E
- Form F
- DIN 76 thread
- Thread general

Operating sequences

Once you have defined the undercut form, you can choose the desired coordinates in the "RxD" (radius * depth) field with the "Select key" or softkey "Alternative".

If the diameter is already known when selecting the undercut, the list box displays a suggested value.

Za is the machining allowance (grinding allowance) permitted according to DIN 509.



N:\MPF.DIR\FREISTICH.MPF

Straight line, horizontal

Z -40.000 abs

Undercut

Form E

RxT 0.1 x 0.1

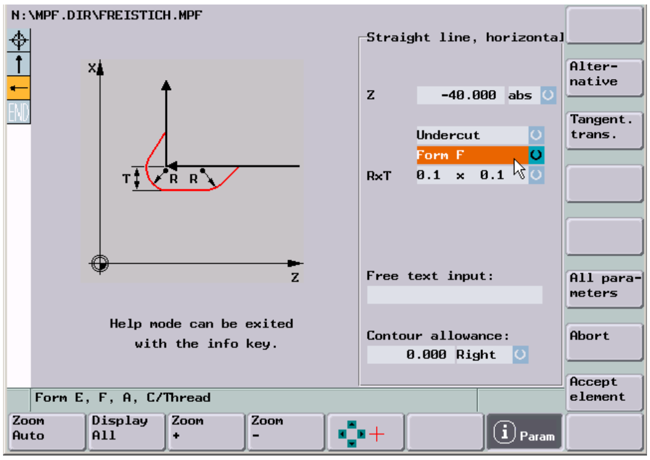
Free text input:

Contour allowance:
0.000 Right

Help mode can be exited
with the info key.

Form E, F, A, C/Thread

Figure 7-30 Undercut E



N:\MPF.DIR\FREISTICH.MPF

Straight line, horizontal

Z -40.000 abs

Undercut

Form F

RxT 0.1 x 0.1

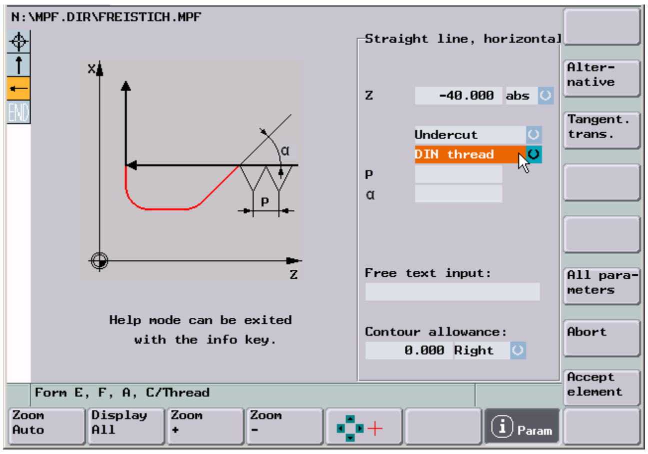
Free text input:

Contour allowance:
0.000 Right

Help mode can be exited
with the info key.

Form E, F, A, C/Thread

Figure 7-31 Undercut F



N:\MPF.DIR\FREISTICH.MPF

Straight line, horizontal

Z -40.000 abs

Undercut
DIN thread

P

α

Free text input:

Contour allowance:
0.000 Right

Help mode can be exited with the info key.

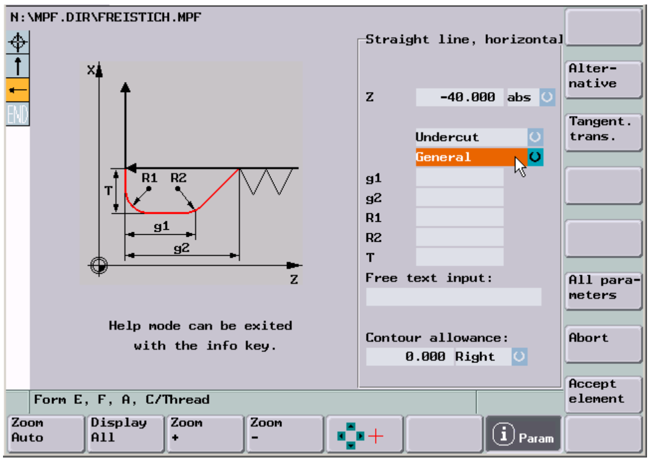
Form E, F, A, C/Thread

Figure 7-32 DIN thread

In the case of standard thread undercuts, the characteristic size of the thread pitch is P. The depth, length and transition radius of the undercut are calculated according to the DIN standard. The (metric) thread pitches specified in DIN 76 can be used. The entry angle can be freely selected in the 30°-90° range. If the diameter is known when selecting the undercut, an appropriate thread pitch is suggested. Forms DIN 76 A (external control) and DIN 76 C (internal control) are available. The program detects the two forms automatically using their geometry and topology.

Thread general

Based on the thread undercut according to DIN (see above) you can use the “General thread” undercut type to create specific undercuts, e.g. for inch threads. The following inputs can be made:



N:\MPF.DIR\FREISTICH.MPF

Straight line, horizontal

Z -40.000 abs

Undercut
General

g1

g2

R1

R2

T

Free text input:

Contour allowance:
0.000 Right

Help mode can be exited with the info key.

Form E, F, A, C/Thread

Figure 7-33 Thread

7.6.5 Parameterize contour element

Functionality

The following softkeys are provided for programming the contour on the basis of preassigned parameters:

Tangent to preceding element

The "Tangent preced. elem." softkey presets the angle α2 to a value of 0. The contour element has a tangential transition to the preceding element, i.e. the angle to the preceding element (α2) is set to 0 degrees.

Display additional parameters

All parameters

If your drawing contains further data (dimensions) for a contour element, select the "All parameters" softkey to extend the range of input options for the element.

Alternative

The "Alternative" softkey is displayed only in cases where the cursor is positioned on an input field with several switchover settings.

Select dialog

Select dialog

Some parameter configurations can produce several different contour characteristics. In such cases, you will be asked to select a dialog. By clicking the "Select dialog" softkey, you can display the available selection options in the graphic display area.

Select dialog | Accept dialog

Select the "Select dialog" softkey to make the correct selection (green line). Confirm your choice with the "Accept dialog" softkey.

Change a selected dialog

Change selection

If you want to change an existing dialog selection, you must select the contour element in which the dialog was originally chosen. Both alternatives are displayed again when you select the "Change selection" softkey.

Select dialog | Accept dialog

You can select another dialog.

Clear a parameter input field

Delete value

You can delete the value in the selected parameter input field with the DEL key or "Delete value" softkey.

Save a contour element

Accept element

If you have entered the available data for a contour element or selected the desired contour by means of the "Select dialog" softkey, select the "Accept element" softkey to store the contour element and return to the main screen. You can then program the next contour element.

Append contour element

Use the cursor keys to select the element in front of the end marker.

Use the softkeys to select the contour element of your choice and enter the values you know in the input screen for that element.

Accept element

Confirm your inputs with the "Accept element" softkey.

Select contour element

INPUT

Position the cursor on the desired contour element in the contour chain, and select it using the <Input> key.

The parameters for the selected element will then be displayed. The name of the element appears at the top of the parameterization window.

If the contour element can be represented geometrically, it is highlighted accordingly in the graphic display area, i.e. the color of the contour element changes from white to black.

Modifying contour element

INPUT

You can use the cursor keys to select a programmed contour element in the contour chain. The <Input> key displays the parameter input fields. The parameters can now be edited.

Insert a contour element

Use the cursor keys in the contour chain to select the contour element in front of the position for the new element.

Then select the contour element to be inserted from the softkey bar.

Accept element

After you have configured the parameters for the new contour element, confirm the insert operation by selecting the "Accept element" softkey.

Subsequent contour elements are updated automatically according to the new contour status.

Delete contour element

Delete element

Use the cursor keys to select the element you wish to delete. The selected contour symbol and associated contour element in the programming graphic are highlighted in red. Then press the "Delete element" softkey and confirm the query.

Undo an input

Cancel

By selecting the "Abort" softkey you can return to the main screen **without** transferring the last edited values to the system.

Contour symbol colors

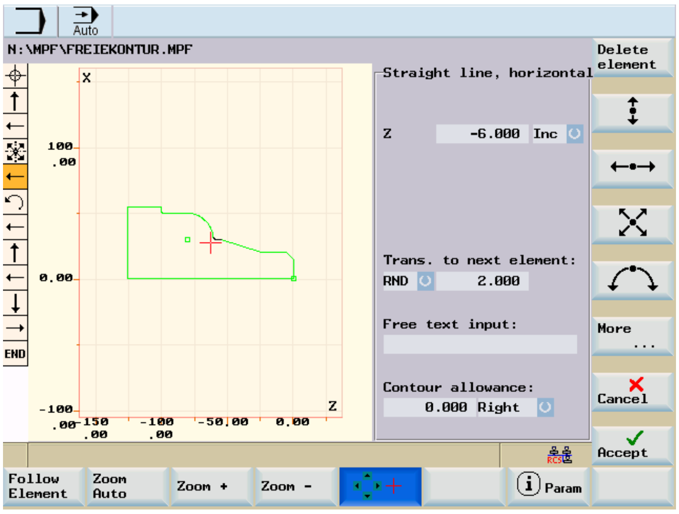
The meaning of the symbol colors in the contour chain on the left of the main screen is as follows:

Icon	Significance
Selected	Symbol color black on a red background -> Element is defined geometrically Symbol color black on a light yellow background -> Element is not defined geometrically
Not selected	Symbol color black on a gray background -> Element is defined geometrically Symbol color white on a gray background -> Element is not defined geometrically

7.6.6 Graphic representation of the contour

Functionality

The graphics window displays the progress of the contour chain as you configure the parameters for the contour elements. The element you have selected is displayed in black in the graphics window. Navigation within the contour is described in section "Program contour".



N:\MPF\FREIEKONTUR.MPF

Straight line, horizontal

Z -6.000 Inc

Trans. to next element:
RND 2.000

Free text input:

Contour allowance:
0.000 Right

Figure 7-34 Contour with arrow

The contour is displayed to the extent it can be interpreted by the control on the basis of parameter inputs. If the contour is still not displayed in the programming graphic, further values must be entered. Check the contour elements you have already programmed, if required. You may have forgotten to enter all of the known data.

The coordinate system scaling is automatically adapted to changes in the complete contour.

The position of the coordinate system is displayed in the graphics window.

Follow Element

An element was selected using the cursor keys.

"Follower element" enlarges the image section of the selected element.

Contour allowance

The stock allowance entered here runs completely parallel to the selected side of the contour.

7.6.7 Specify contour elements in polar coordinates, close the contour

Functionality

The description given above of defining the coordinates of contour elements applies to the specification of positional data in the Cartesian coordinate system. Alternatively, you have the option to define positions using polar coordinates.

When programming contours, you can define a pole at any time prior to using polar coordinates for the first time. Programmed polar coordinates subsequently refer to this pole. The pole is modal and can be re-defined at any time. It is always entered in absolute Cartesian coordinates. The contour calculator converts values entered as polar coordinates into Cartesian coordinates. Positions can be programmed in polar coordinates only **after** a pole has been specified. The pole input does not generate a code for the NC program.

Pole

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

The pole is a contour element that can be edited, which itself does not contribute to the contour. It can be entered when the starting point of the contour is defined or anywhere within the contour. The pole cannot be created before the starting point of the contour.

Enter polar coordinates

More

The "More" softkey in the basic plane of the contour programming accesses the "Pole" subscreen and the "Close contour" softkey.

Pole

The pole can only be entered in absolute Cartesian coordinates. The "Pole" softkey is also present in the starting point screen. This enables the pole to be entered at the start of a contour, so that the first contour element can be entered in polar coordinates.

Close
contour

The contour is closed by a straight line between the last entered contour point and the starting point.

Further notes

If the straight line that was generated with close contour is linked to the start element of the contour with a radius or chamfer, the radius or chamfer must be specified explicitly as follows:

- Close contour, input key, enter radius/chamfer, accept element. The result then corresponds exactly to what would occur if the closing element were to be entered with the radius or chamfer.

  Close contour can only be used for entering contour elements in **polar coordinates** if the starting point of the contour was set to polar and the **same pole** is still valid when the contour is closed.

Input switchover: Cartesian/polar

The following contour elements can be entered optionally in polar coordinates only after a pole has been defined, whether this was done at the outset or later in the process:

- Circular arcs,
- Straight lines (horizontal, vertical, any direction)

To switchover between Cartesian and polar, additional toggle fields are displayed for "Straight line any" and "Circular arc" in both the basic contour input view as well as the view containing "All parameters".

A toggle field is not displayed if no pole exists. Input fields and display fields are then only available for Cartesian values.

Absolute/incremental input

Absolute and incremental polar coordinates can be input for "polar/Cartesian". The input fields and display fields are labeled **ink** and **abs**.

Absolute polar coordinates are defined by an absolute distance to the pole that is always positive and an angle in the range of 0° ... +/- 360°. When absolute dimensions are specified, the angular reference is based on a horizontal axis of the working plane, e.g. X axis with G17. The positive direction of rotation runs counter-clockwise.

If there are several input poles, the definitive pole is always the **last pole** before the input or edited element.

Incremental polar coordinates relate to both the definitive pole and the end point of the preceding element.

For an incremental input, **the absolute distance** to the pole is calculated using the absolute distance from the end point of the preceding element to the pole plus the length increment that was entered.

The increment can be positive or negative.

The absolute angle is calculated accordingly using the absolute polar angle of the preceding element plus the angular increment. Here it is not necessary for the preceding element to have been entered as polar.

In contour programming, the contour calculator converts the Cartesian coordinates of the preceding end point using the definitive pole into polar coordinates. This also applies if the preceding element has been given in polar coordinates, since this could relate to another pole if a pole has been inserted in the meantime.

Pole change example

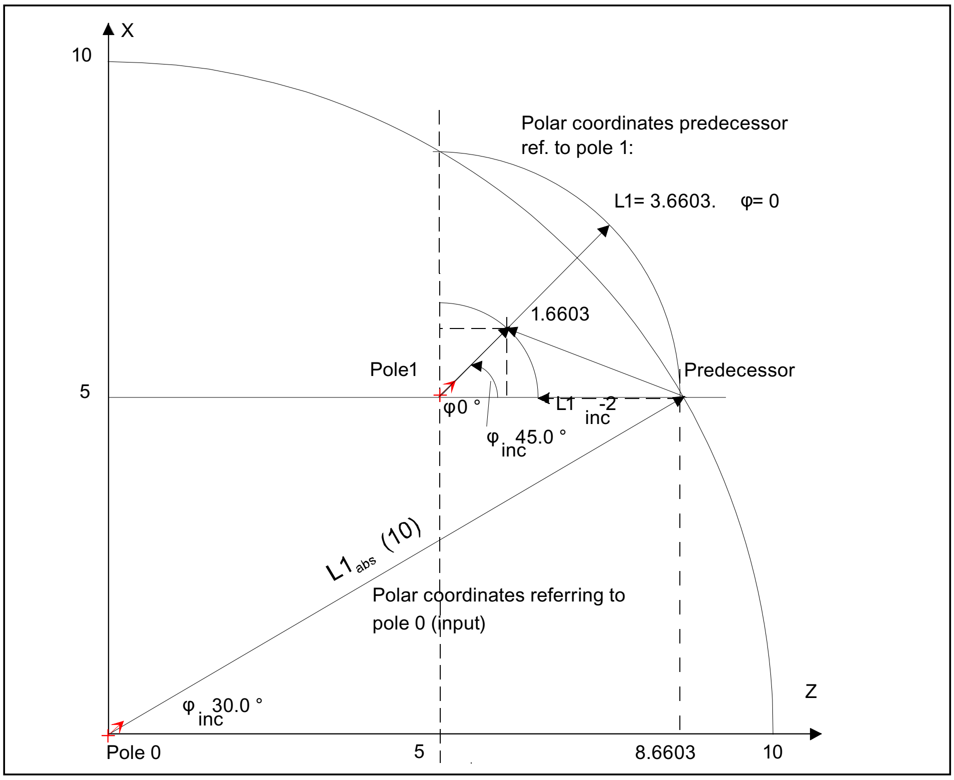
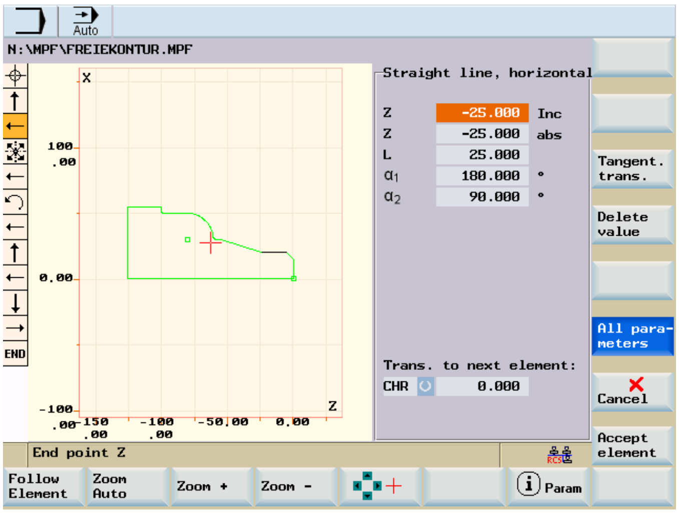


Figure 7-35 Pole change

Pole:	Zpole = 0.0,	Xpole = 0.0,	(Pole 0)
End point:			
L1abs = 10.0	ϕabs = 30.0°	Calculated Cart. Coordinates	
		Zabs = 8.6603	Xabs = 5.0
New pole:			
Zpole1 = 5.0	Xpole1 = 5.0		(Pole 1)
		Calculated polar coord. Predecessor	
		L1abs = 3.6603	ϕabs = 0.0°
Next point:			
L1inc = -2.0	ϕinc = 45.0°		
		Absolute polar coordinates for current element	
		L1abs = 1.6603	ϕabs = 45.0°
		Calculate Cartesian coordinates	
		Zabs = 1.1740	Xabs = 1.1740

7.6.8 Parameter description of straight line/circle contour elements

Parameters for contour element "Straight line"



N:\MPF\FREIEKONTUR.MPF

Straight line, horizontal

Z	-25.000	Inc
Z	-25.000	abs
L	25.000	
α1	180.000	°
α2	90.000	°

Trans. to next element:

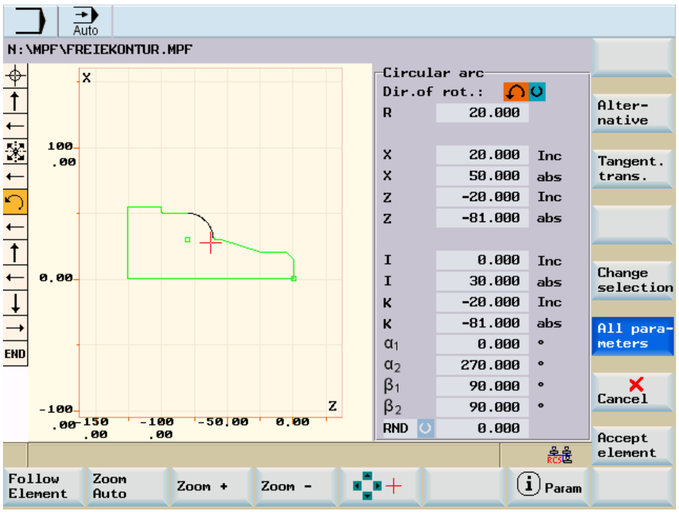
CHR 0.000

End point Z

Figure 7-36 Straight horizontal line

Parameter	Contour element "Straight line"
X inc X abs	Incremental end position in X direction Absolute end position in X direction
Z inc Z abs	Incremental end position in Z direction Absolute end position in Z direction
L	Length of straight line
α1	Pitch angle with reference to X axis
α2	Angle to preceding element; tangential transition: α2=0
Transition to next element	Transition element to next contour is a chamfer (CHR) Transition element to next contour is a radius (RND) CHR=0 or RND=0 means no transition element.

Parameters for contour element "Circular arc"



N:\MPF\FREIEKONTUR.MPF

```
Circular arc
Dir.of rot.:
R      20.000
X      20.000  Inc
X      50.000  abs
Z     -20.000  Inc
Z     -81.000  abs
I       0.000  Inc
I      30.000  abs
K     -20.000  Inc
K     -81.000  abs
α1      0.000  °
α2    270.000  °
β1     90.000  °
β2     90.000  °
RND     0.000
```

Figure 7-37 Circular arc

Parameter	Contour element "Circle"
Direction of rotation	In clockwise or counter-clockwise direction
R	Radius of circle
X inc X abs	Incremental end position in X direction Absolute end position in X direction
Z inc Z abs	Incremental end position in Z direction Absolute end position in Z direction
I	Position of circle center point in X direction (abs. or incr.)
K	Position of circle center point in Z direction (abs. or incr.)
α1	Starting angle with reference to X axis
α2	Angle to preceding element; tangential transition: α2=0
β1	End angle with reference to X axis
β2	Angle of aperture of circle
Transition to next element	Transition element to next contour is a chamfer (CHR) Transition element to next contour is a radius (RND) CHR=0 or RND=0 means no transition element.

Machine manufacturer

The names of the identifiers (X or Z ...) are defined in the machine data where they can also be changed.

7.6.9 Cycle support

Functionality

The following technologies are provided with additional support in the form of pre-defined cycles, which must then be parameterized.

- Drilling
- Turning

See also

Cycles (Page 343)

7.6.10 Programming example for turning application

Example

The following diagram shows a programming example for the "Free contour programming" function.

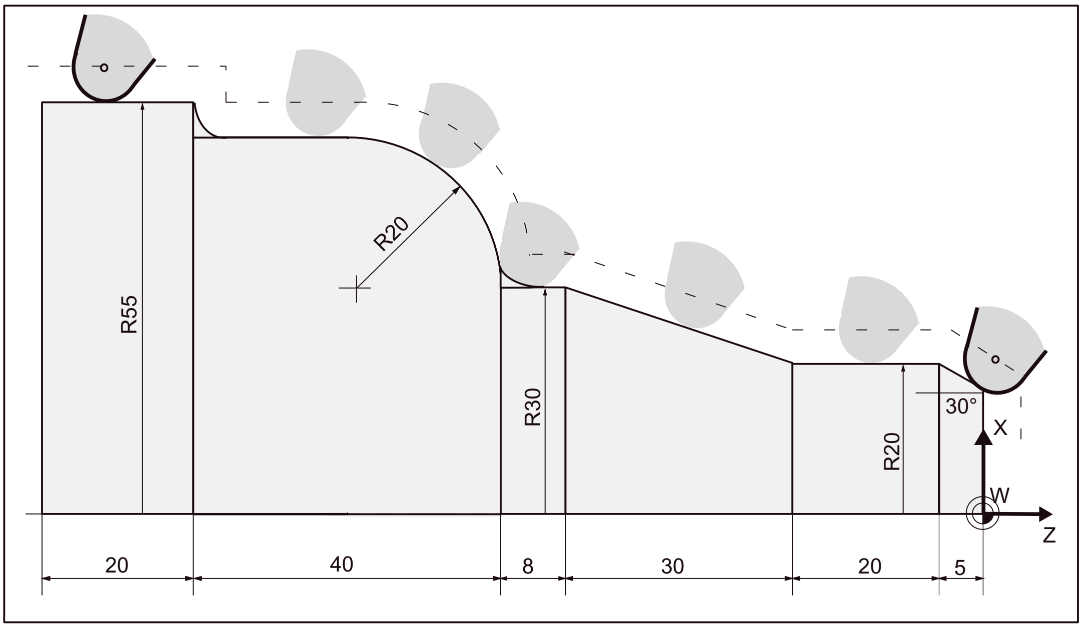


Figure 7-38 Programming example for turning application

Operating sequences

You have opened a part program in the "Program Manager" operating area

The sequence of individual actions required to program the contour are listed in the table below.

Note

When a contour is programmed in the input screens, the input field with the input focus is highlighted by a dark background color. Once the input is acknowledged with "Accept element" or "Abort", you can navigate around the contour chain (left of the graphic) using the ↑, ↓ cursor keys. The current position in the chain is color-highlighted.

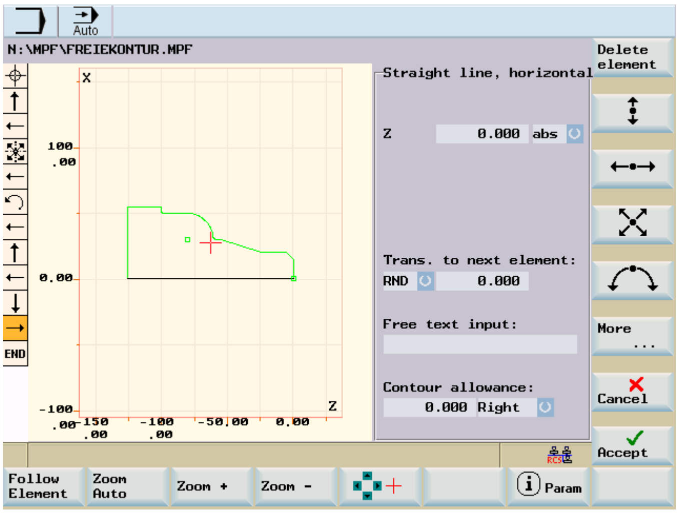
You can call the particular input screen form using the <Input> key and enter new parameters.

Table 7- 1 Programming example for turning application

Operating step	Softkey	Parameter
1	"Contour" "Accept element"	Enter start point: Programming plane: G18 ;Dimension for transverse axis: Radius DIAMOF Z: 0 X: 0
2	↕ "Accept element"	Enter parameters for "Straight vertical line" element: X: 20 inc CHR: Chamfer length = 5*1.1223 = 5.6115
3	←•→ "Accept element"	Enter parameters for "Straight horizontal line" element: Z: -25 inc
4	⤧ "Accept element"	Enter parameters for "Straight line in any direction" element: X: 10 inc Z: -30 inc
5	←•→ "Accept element"	Enter parameters for "Straight horizontal line" element: Z: -8 inc. Transition to following element: RND: 2
6	↶•↷ "Select dialog" "Accept dialog" "Accept element"	Enter parameters for "Circular arc" element: Direction of rotation: Counter-clockwise R: 20 X: 20 inc Z: -20 inc

Operating step	Softkey	Parameter
7	↔ "Accept element"	Enter parameters for "Straight horizontal line" element: Z: -20 inc Transition to following element: RND: 2
8	↕ "Accept element"	Enter parameters for "Straight vertical line" element: X: 5 inc.
9	↔ "Accept element" "Continue..." "Close contour" "<<Back" "Accept"	Enter parameters for "Straight horizontal line" element: Z: -25 inc

The following screenshot shows the programming contour:



N:\MPF\FREIEKONTUR.MPF

Straight line, horizontal

Z 0.000 abs

Trans. to next element:
RND 0.000

Free text input:

Contour allowance:
0.000 Right

Figure 7-39 Programmed contour

System 8

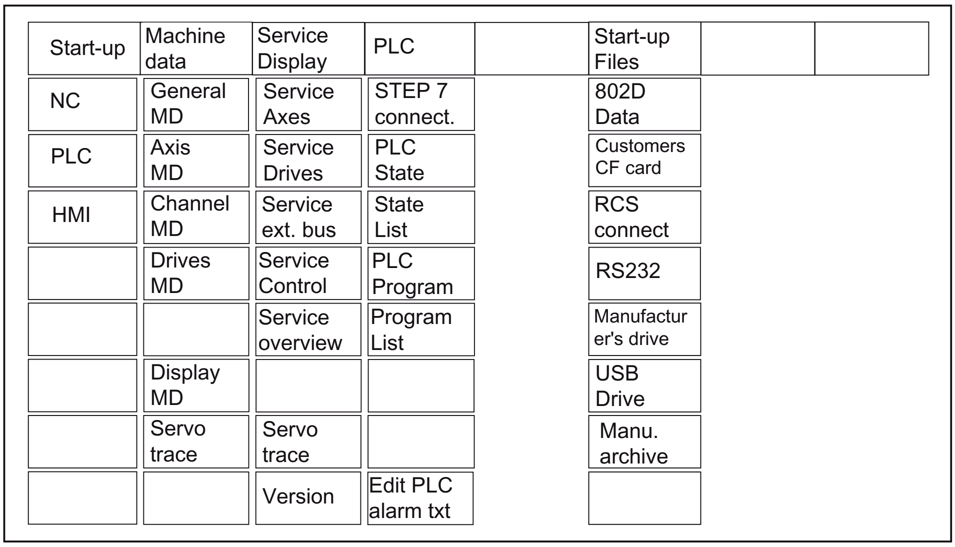
8.1 "System" operating area

Functionality

The SYSTEM operating area includes functions required for parameterizing and analyzing the NCK, the PLC and the drive.

Depending on the functions selected, the horizontal and the vertical softkey bars change. The menu tree shown below **only** includes the horizontal softkeys.

Menu tree



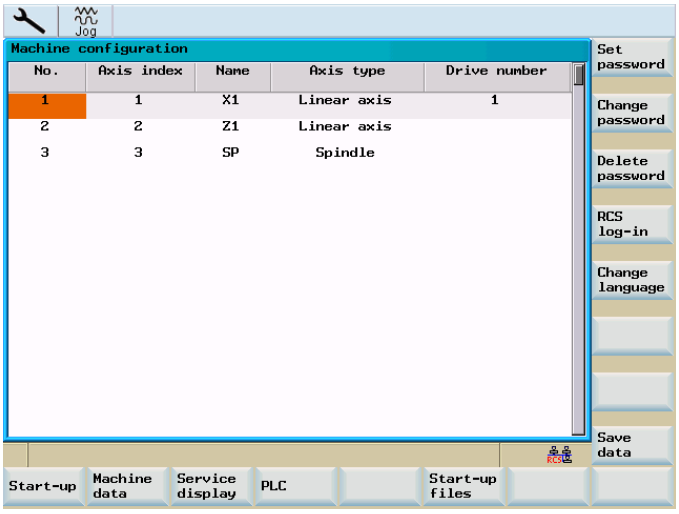
Start-up	Machine data	Service Display	PLC		Start-up Files		
NC	General MD	Service Axes	STEP 7 connect.		802D Data		
PLC	Axis MD	Service Drives	PLC State		Customers CF card		
HMI	Channel MD	Service ext. bus	State List		RCS connect		
	Drives MD	Service Control	PLC Program		RS232		
		Service overview	Program List		Manufacturer's drive		
	Display MD				USB Drive		
	Servo trace	Servo trace			Manu. archive		
		Version	Edit PLC alarm txt				

Figure 8-1 System menu tree

Operating sequence

SHIFT + SYSTEM ALARM

The full CNC keyboard is used to change to the <SHIFT> and <SYSTEM> operating areas and the start screen is displayed.



Machine configuration

No.	Axis index	Name	Axis type	Drive number
1	1	X1	Linear axis	1
2	2	Z1	Linear axis	
3	3	SP	Spindle	

Figure 8-2 "System" operating area start screen

Softkeys

The start screen vertical softkeys are described below.

Set password

"Set password"

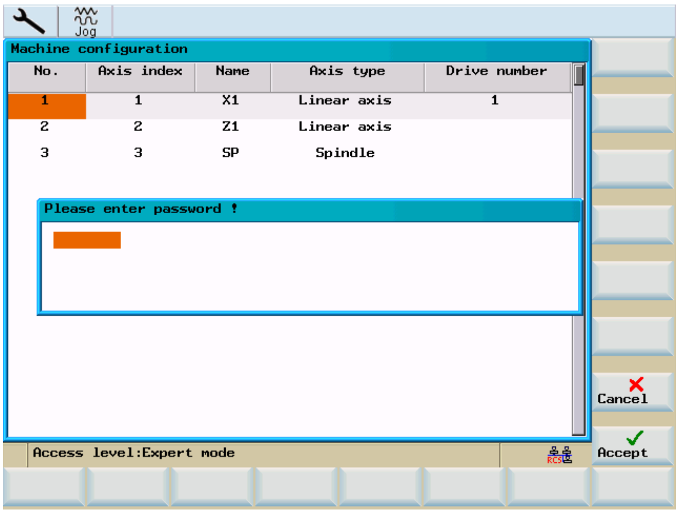
Three password levels are distinguished in the control system, which provide different access rights:

- System password
- Manufacturer password
- User password

It is possible to change certain data corresponding to the access levels. If you do not know the password, access will be denied.

Note

Also see SINUMERIK 802D sl "Lists".



Machine configuration

No.	Axis index	Name	Axis type	Drive number
1	1	X1	Linear axis	1
2	2	Z1	Linear axis	
3	3	SP	Spindle	

Please enter password !

Access level:Expert mode

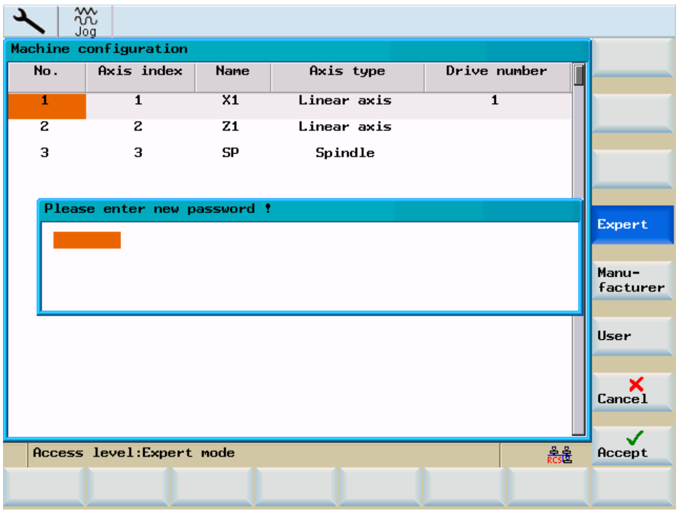
Figure 8-3 Entering the password

After selecting the "Accept" softkey, the password is set.

Use "Abort" to return without any action to the "System" start screen.

Change password

"Change Password"



Machine configuration

No.	Axis index	Name	Axis type	Drive number
1	1	X1	Linear axis	1
2	2	Z1	Linear axis	
3	3	SP	Spindle	

Please enter new password !

Access level:Expert mode

Figure 8-4 Change password

Depending on the access right, various possibilities are offered in the softkey bar to change the password.

Select the password level using the appropriate softkeys. Enter the new password and press "Accept" to complete your input. You will be prompted to enter the new password once more for confirmation.

Press "Accept" to complete the password change.

Use "Abort" to return without any action to the start screen.

Delete password

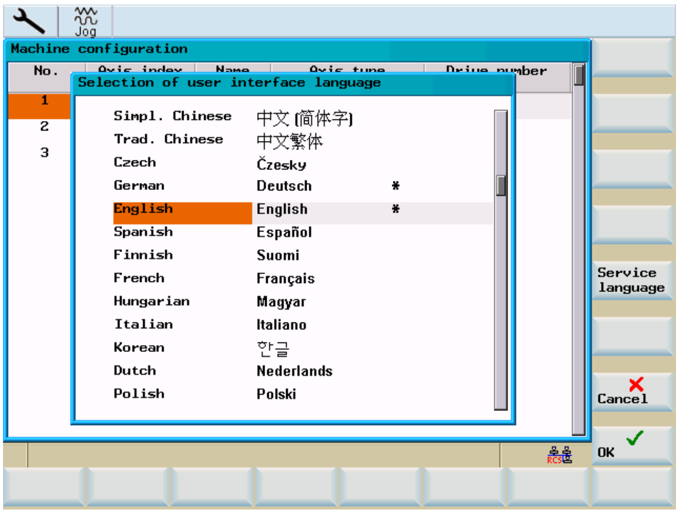
Resetting the credential

RCS log-in

User network log-in

Change language

Use "Change language" to select the user interface language.



Selection of user interface language

Simpl. Chinese	中文 (简体字)	
Trad. Chinese	中文繁体	
Czech	Česky	
German	Deutsch	*
English	English	*
Spanish	Español	
Finnish	Suomi	
French	Français	
Hungarian	Magyar	
Italian	Italiano	
Korean	한글	
Dutch	Nederlands	
Polish	Polski	

Figure 8-5 User interface language

Use the cursor keys to select the language and confirm it by pressing "OK".

Note

The HMI is automatically restarted when a new language is selected.

Service language

Use "Service language" to always select "English" as the user interface language.

Press the "Service language" softkey again to restore the previously active language (e. g. "Simpl. Chinese").

Note

An asterisk "*" marks the languages you have used.

Save data

"Save data"

This function will save the contents of the volatile memory into a nonvolatile memory area.

Requirement: There is no program currently executed.

Do not carry out any operator actions while the data backup is running!

The NC and PLC data are backed up. The drive data are not backed up.

Note

Saved data can be called via the following operator action:

- Press the <SELECT> key while the control system is booting.
- In the setup menu, select "Reload saved user data".
- Press the <Input> key

Note

Data that have been backed up can be called again from the operating area <SYSTEM> > "Start-up" > "Power up with backed up data"!

8.2 SYSTEM - "Start-up" softkeys

Start-up

Commissioning

NC

Use this softkey to select the NC power-up mode.

Select the desired mode using the cursor.

- Normal power-up
  The system is restarted
- Power-up with default data
  The display machine data are reset to their standard values (restores the initial state when originally supplied)
- Power-up with backed up data
  The system restarts with the data that were last backed up (see Backup data)

PLC

The PLC can be started in the following modes:

- Restart
- Memory reset

Furthermore, it is possible to link the start with a subsequent **debugging mode**.

HMI

Selects the power-up mode of the HMI.

Select the desired mode using the cursor.

- Normal power-up
  The system is restarted
- Power-up with default data
  The system restarts with default values (restores the initial state when originally supplied)

OK ✓

Use "OK" to RESET the control system and to carry out a restart in the mode selected.

Use the <RECALL> key to return to the system start screen without performing any action.

8.3 SYSTEM - "Machine data" softkeys

References

You will find a description of the machine data in the following manufacturers´ documents:

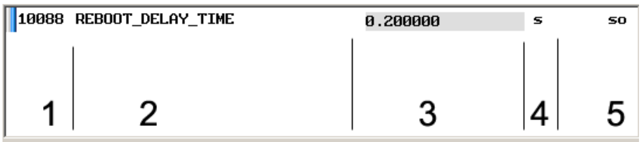
SINUMERIK 802D sl List Manual

SINUMERIK 802D sl Function Manual for turning, milling, nibbling

Machine data

Machine data

Any changes in the machine data have a substantial influence on the machine.



```
10088 REBOOT_DELAY_TIME          0.200000     s     so
```

Figure 8-6 Structure of a machine data line

Table 8- 1 Legend

No.	Significance		
1	MD number		
2	Name		
3	Value		
4	Unit		
5	Effective	so	immediately effective
		cf	with confirmation
		re	Reset
		po	Power on

⚠ CAUTION

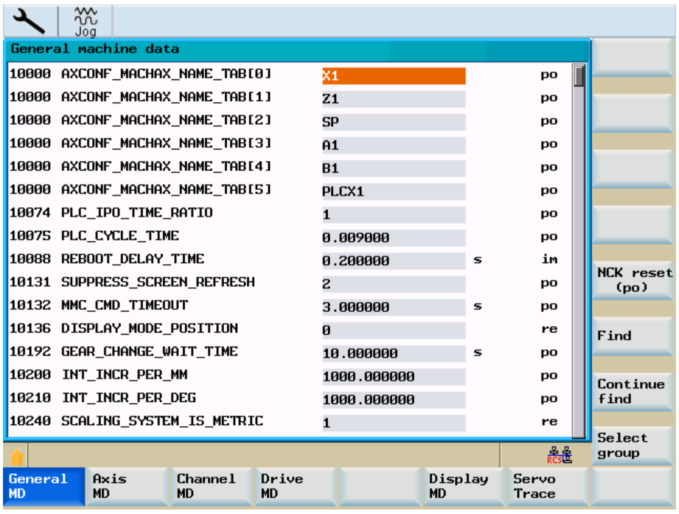
Incorrect parameterization may result in destruction of the machine!

The machine data are divided into the groups described in the following.

General machine data

General MD

Open the "General machine data" window. Use the Page Up / Page Down keys to browse forward / backward.



General machine data

10000	AXCONF_MACHAX_NAME_TAB[0]	X1		po
10000	AXCONF_MACHAX_NAME_TAB[1]	Z1		po
10000	AXCONF_MACHAX_NAME_TAB[2]	SP		po
10000	AXCONF_MACHAX_NAME_TAB[3]	A1		po
10000	AXCONF_MACHAX_NAME_TAB[4]	B1		po
10000	AXCONF_MACHAX_NAME_TAB[5]	PLCX1		po
10074	PLC_IPO_TIME_RATIO	1		po
10075	PLC_CYCLE_TIME	0.009000		po
10088	REBOOT_DELAY_TIME	0.200000	s	im
10131	SUPPRESS_SCREEN_REFRESH	2		po
10132	MMC_CMD_TIMEOUT	3.000000	s	po
10136	DISPLAY_MODE_POSITION	0		re
10192	GEAR_CHANGE_WAIT_TIME	10.000000	s	po
10200	INT_INCR_PER_MM	1000.000000		po
10210	INT_INCR_PER_DEG	1000.000000		po
10240	SCALING_SYSTEM_IS_METRIC	1		re

Figure 8-7 General machine data

NCK reset (po)

Executes a warm restart at the control.

Find

"Find"

Type the number or the name (or a part of the name) of the machine data you are looking for and press "OK".

The cursor will jump to the data searched.

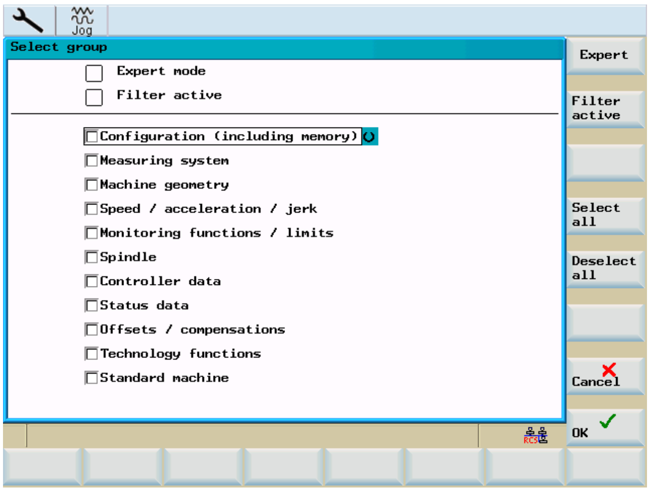
Continue find

Use this softkey to continue searching for the next match.

Select group

This function provides various display filters for the active machine data group. Further softkeys are provided:

- "Expert": Use this softkey to select all data groups of the expert mode for display.
- "Filter active": Use this softkey to activate all data groups selected. After you have quit the window, you will only see the selected data on the machine data display.
- "Select all": Use this softkey to select all data groups of the Expert mode for display.
- "Deselect all": Selecting this softkey deselects all data groups.



Select group

□ Expert mode
□ Filter active

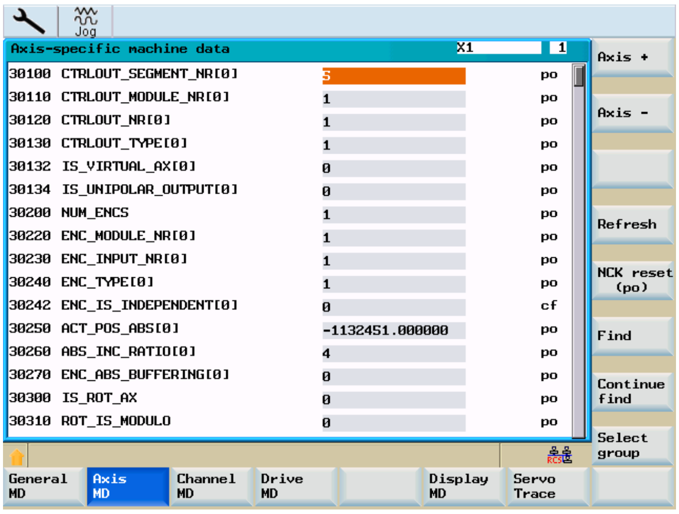
□ Configuration (including memory)
□ Measuring system
□ Machine geometry
□ Speed / acceleration / jerk
□ Monitoring functions / limits
□ Spindle
□ Controller data
□ Status data
□ Offsets / compensations
□ Technology functions
□ Standard machine

Figure 8-8 Display filter

Axis-specific machine data

axis MD

Open the "Axis-specific machine data" window. The softkey bar will be supplemented by the softkeys "Axis +" and "Axis ".



Axis-specific machine data X1 1

30100	CTRLOUT_SEGMENT_NR[0]	5	po
30110	CTRLOUT_MODULE_NR[0]	1	po
30120	CTRLOUT_NR[0]	1	po
30130	CTRLOUT_TYPE[0]	1	po
30132	IS_VIRTUAL_AX[0]	0	po
30134	IS_UNIPOLAR_OUTPUT[0]	0	po
30200	NUM_ENCS	1	po
30220	ENC_MODULE_NR[0]	1	po
30230	ENC_INPUT_NR[0]	1	po
30240	ENC_TYPE[0]	1	po
30242	ENC_IS_INDEPENDENT[0]	0	cf
30250	ACT_POS_ABS[0]	-1132451.000000	po
30260	ABS_INC_RATIO[0]	4	po
30270	ENC_ABS_BUFFERING[0]	0	po
30300	IS_ROT_AX	0	po
30310	ROT_IS_MODULO	0	po

Figure 8-9 Axis-specific machine data

The data of axis 1 are displayed.

Axis +

Use "Axis +" or "Axis " to switch to the machine area of the next or previous axis.

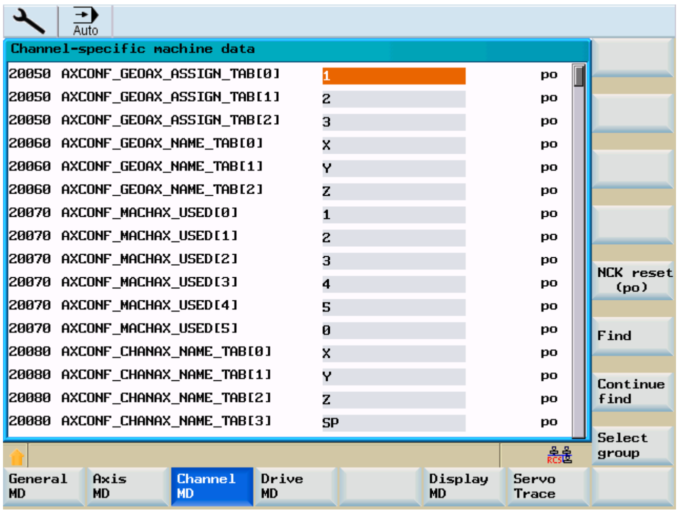
Update

The contents of the machine data are updated.

Channel-specific machine data

chan MD

Open the "Channel-specific machine data" window. Use the PageUp / PageDown keys to browse forward / backward.



Channel-specific machine data

20050	AXCONF_GEOAX_ASSIGN_TAB[0]	1	po
20050	AXCONF_GEOAX_ASSIGN_TAB[1]	2	po
20050	AXCONF_GEOAX_ASSIGN_TAB[2]	3	po
20060	AXCONF_GEOAX_NAME_TAB[0]	X	po
20060	AXCONF_GEOAX_NAME_TAB[1]	Y	po
20060	AXCONF_GEOAX_NAME_TAB[2]	Z	po
20070	AXCONF_MACHAX_USED[0]	1	po
20070	AXCONF_MACHAX_USED[1]	2	po
20070	AXCONF_MACHAX_USED[2]	3	po
20070	AXCONF_MACHAX_USED[3]	4	po
20070	AXCONF_MACHAX_USED[4]	5	po
20070	AXCONF_MACHAX_USED[5]	0	po
20080	AXCONF_CHANAX_NAME_TAB[0]	X	po
20080	AXCONF_CHANAX_NAME_TAB[1]	Y	po
20080	AXCONF_CHANAX_NAME_TAB[2]	Z	po
20080	AXCONF_CHANAX_NAME_TAB[3]	SP	po

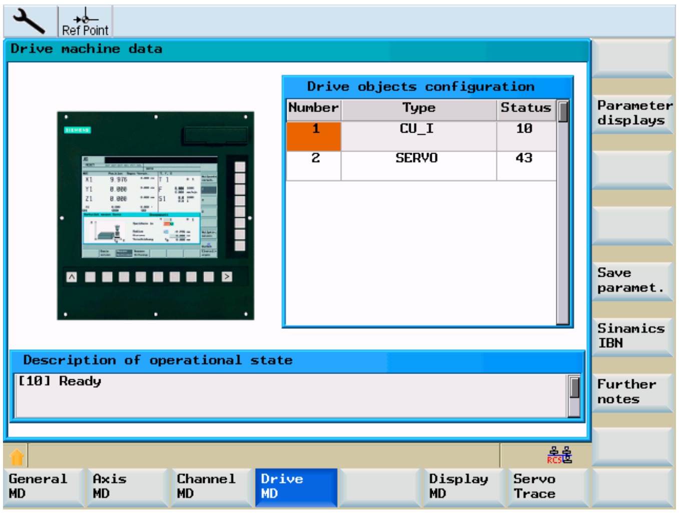
Figure 8-10 Channel-specific machine data

SINAMICS drive machine data

Drive MD

Open the "Drive machine data" dialog box.

The first dialog box displays the current configuration, as well as the states of the control, power supply and drive units.



Drive machine data

Drive objects configuration

Number	Type	Status
1	CU_I	10
2	SERVO	43

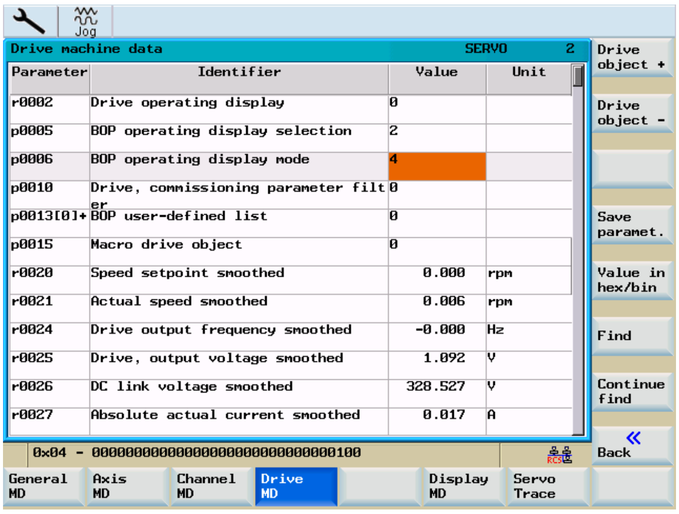
Description of operational state

[10] Ready

Figure 8-11 Drive machine data

Display parameter

To display the parameters, position the cursor on the appropriate unit and press the "Parameter display" softkey. For a description of the parameters, please refer to the documentation of SINAMICS drives.



Drive machine data SERVO 2

Parameter	Identifier	Value	Unit
r0002	Drive operating display	0	
p0005	BOP operating display selection	2	
p0006	BOP operating display mode	4	
p0010	Drive, commissioning parameter filter	0	
p0013[0]+	BOP user-defined list	0	
p0015	Macro drive object	0	
r0020	Speed setpoint smoothed	0.000	rpm
r0021	Actual speed smoothed	0.006	rpm
r0024	Drive output frequency smoothed	-0.000	Hz
r0025	Drive, output voltage smoothed	1.092	V
r0026	DC link voltage smoothed	328.527	V
r0027	Absolute actual current smoothed	0.017	A

0x04 - 00000000000000000000000000000100

Figure 8-12 Parameter list

Drive object +

Drive object -

Switch to the respective drive objects.

Value in hex/bin

In the note line, the selected value is displayed in hexadecimal and binary values.

Find

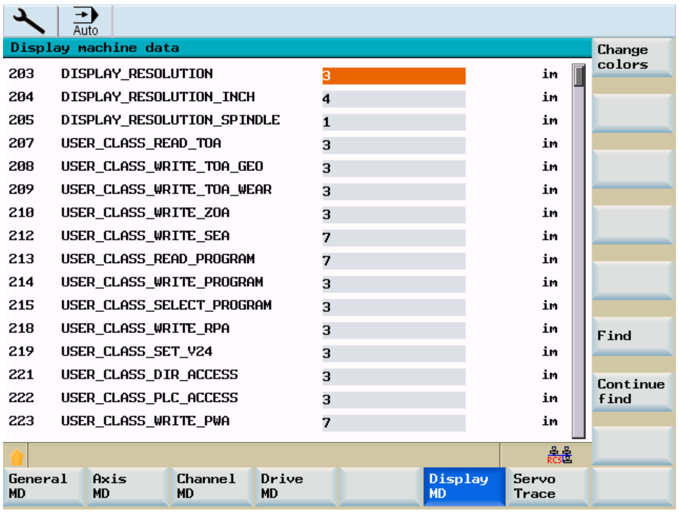
Use these functions to search in the parameter list for the term you are looking for.

Continue find

Display of machine data

Display MD

Open the "Display machine data" window. Use the PageUp / PageDown keys to browse forward / backward.



Display machine data

203	DISPLAY_RESOLUTION	3	im
204	DISPLAY_RESOLUTION_INCH	4	im
205	DISPLAY_RESOLUTION_SPINDLE	1	im
207	USER_CLASS_READ_TOA	3	im
208	USER_CLASS_WRITE_TOA_GEO	3	im
209	USER_CLASS_WRITE_TOA_WEAR	3	im
210	USER_CLASS_WRITE_ZOA	3	im
212	USER_CLASS_WRITE_SEA	7	im
213	USER_CLASS_READ_PROGRAM	7	im
214	USER_CLASS_WRITE_PROGRAM	3	im
215	USER_CLASS_SELECT_PROGRAM	3	im
218	USER_CLASS_WRITE_RPA	3	im
219	USER_CLASS_SET_V24	3	im
221	USER_CLASS_DIR_ACCESS	3	im
222	USER_CLASS_PLC_ACCESS	3	im
223	USER_CLASS_WRITE_PWA	7	im

Figure 8-13 Display of machine data

Color changing

Use the "Softkey color" and "Window color" softkeys to specify user-defined color settings. The displayed color consists of the components red, green and blue.
The "Change color" window displays the values currently set in the input fields. The desired color can be produced by changing these values. In addition, the brightness can be changed.

The next mixing ratio is displayed temporarily upon completion of an input. Use the cursor keys to switch between the input fields.

With "OK", the settings are accepted and the dialog box is closed. Selecting the "Abort" softkey will close the dialog box without accepting your changes.

Color Softkey

Use this function to change the colors of the tip and softkey area.

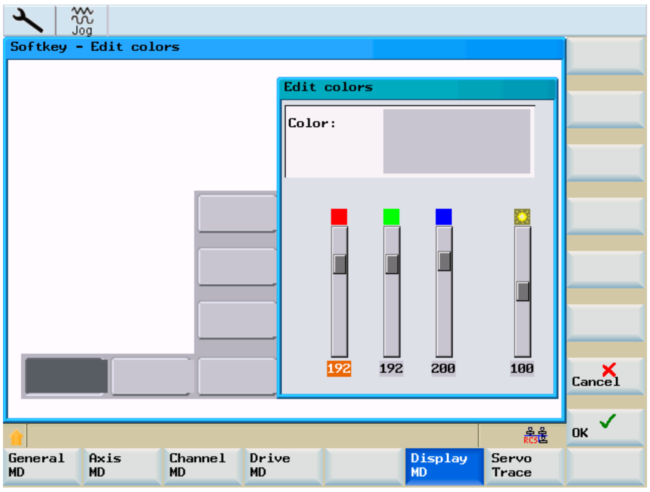


Figure 8-14 Edit softkey color.

Color Window

Use this softkey to change the color of the border of dialog boxes.
The "Active window" softkey function will assign your settings to the focus window, and the "Inactive window" function to the non-active window.

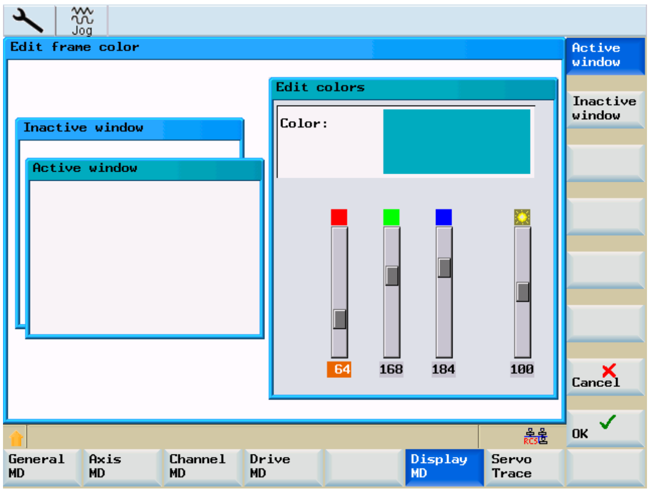


Figure 8-15 Edit frame color.

8.4 SYSTEM - "Service display"

Service display

The "Service display" window appears on the screen.

The start screen for the "Service control" function is shown in the following diagram.

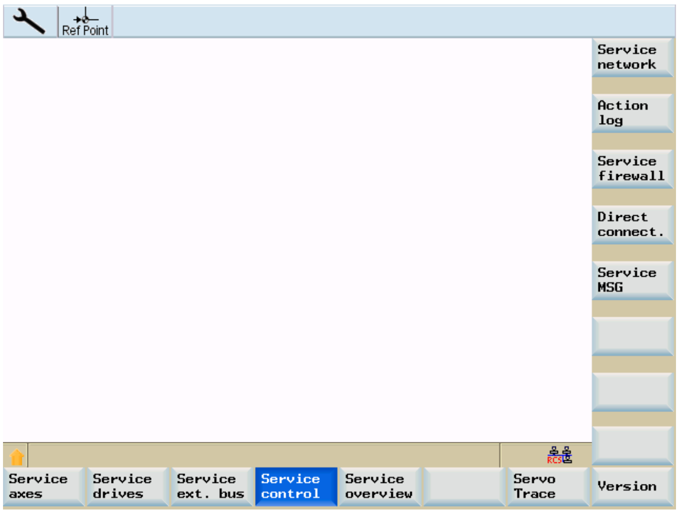


Figure 8-16 The "Service control" start screen

Service axes

This window displays information on the axis drive.

The "Axis +" or "Axis -" softkeys are additionally displayed. These can be used to display the values for the next or previous axis.

Service drives

This window displays information in respect of the digital drive.

Service ext. Bus

The window displays information on external bus settings.

Service control

Use this softkey function to activate the window for the following functions:

- "Service network" (see chapter "Network operation")
- "Action log" (see chapter "Action log")
- "Service Firewall" (see chapter "Network operation")
- "Direct connect." (see chapter "Network operation")
- "Service MSG" (see chapter "Service MSG")

Service Overview

This window contains information about

- Assignment, Machine axis <=> Channel axis <=> Drive number
- The enable status of the NC and drive
- Drive state regarding ready, faults and alarms

Servo trace

An oscilloscope function is available in this window to optimize the drives (see chapter "Servo trace").

Version

This window displays the version numbers and the date of creation of the individual CNC components.

The following functions can be selected from this window (also see chapter "Versions"):

- "HMI details"
- "License key"
- "Options"
- "Save as"

  The displayed versions can be saved in a text file

8.4.1 Action log

Action log

The function "Action log" is provided for service events. The contents of the action log file can only be accessed through a system password on the HMI.

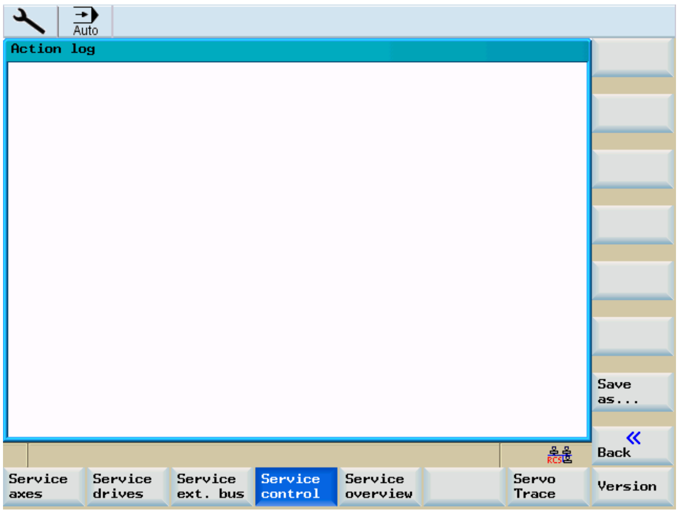


Figure 8-17 Action log

Save under

Irrespective of the system password, it is possible to output the file using softkey "Save under..." on a CF card or on the USB FlashDrive.

In case of difficulties, please contact the hotline (see the "Technical Support" Section in the preface for contacting the hotline).

8.4.2 Servo trace

Servo trace

An oscilloscope function is provided for the purpose of optimizing the drives. This enables graphical representation:

- of the velocity setpoint
- of the contour violation
- of the following error
- of the actual position value
- of the position setpoint
- of exact stop coarse / fine

The start of tracing can be linked to various criteria allowing a synchronous tracing of internal control states. This setting must be made using the "Select signal" function.

To analyze the result, the following functions are provided:

- Changing and scaling of abscissa and ordinate;
- Measuring of a value using the horizontal or vertical marker;
- Measuring of abscissa and ordinate values as a difference between two marker positions;
- Storing of the result as a file in the part program directory. Thereafter, it is possible to export the file using either RCS802 or the CF card and to process the data in MS Excel.

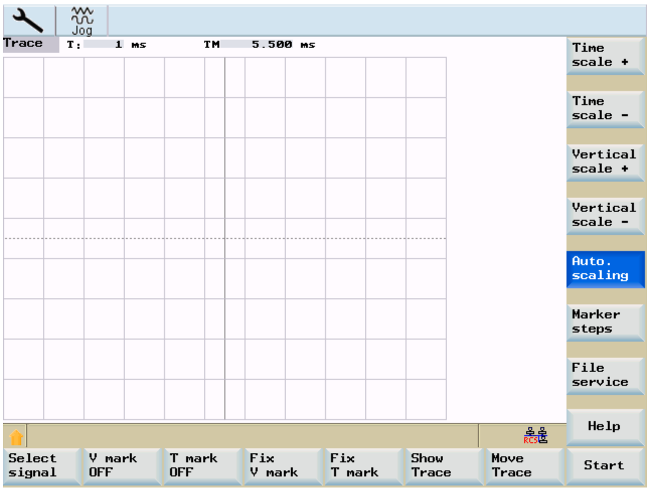
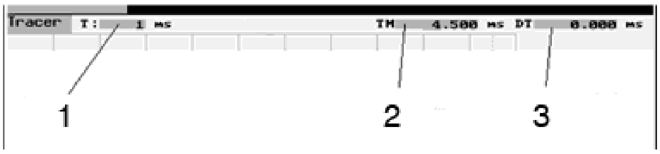


Figure 8-18 Servo trace start screen

The header of the diagram contains the current scaling of the abscissa and the difference value of the markers.

The diagram shown above can be moved within the visible screen area using the cursor keys.



Tracer T: 1 ms TM 4.500 ms DT 0.000 ms

1 Time Base

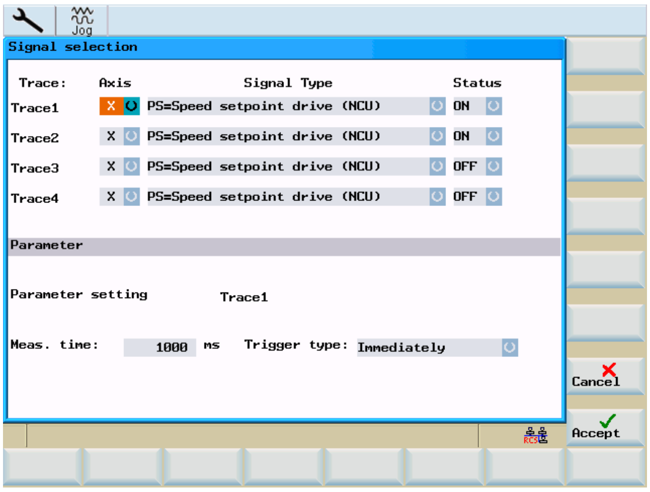
2 Marker position time

3 Difference in time between marker 1 and current marker position.

Figure 8-19 Meaning of the fields

Select signal

Use this menu to parameterize the measuring channel.



Signal selection

Trace:	Axis	Signal Type	Status
Trace1	X	PS=Speed setpoint drive (NCU)	ON
Trace2	X	PS=Speed setpoint drive (NCU)	ON
Trace3	X	PS=Speed setpoint drive (NCU)	OFF
Trace4	X	PS=Speed setpoint drive (NCU)	OFF

Parameter

Parameter setting Trace1

Meas. time: 1000 ms Trigger type: Immediately

Figure 8-20 Select signal

- **Selecting the axis:** To select the axis, use the "Axis" toggle field.
- **"Signal type":**

  Following error
  Controller difference
  Contour deviation
  Position actual value
  Speed actual value
  Speed setpoint
  Compensation value
  Parameter block
  Position setpoint controller input
  Speed setpoint controller input
  Acceleration setpoint controller input
  Speed feedforward control value
  Exact stop fine signal
  Exact stop coarse signal
- **"Status":**

  On: Tracing is performed in this channel
  Off: Channel inactive

The parameters for the measuring time and for the trigger type for channel 1 can be set in the lower screen half. The remaining channels will accept this setting.

- **Determining the measuring period:** The measuring period in ms is entered directly into the "Measuring period" input field (6,133 ms max.).
- **Selecting the trigger condition:** Position the cursor on the "Trigger condition" field and select the relevant condition using the toggle key.
  – No trigger, i.e. the measurement starts directly after selecting the "Start" softkey;
  – Positive edge;
  – Negative edge
  – Exact stop fine reached;
  – Exact stop coarse reached

V mark off

T mark off

The "V mark ON" / "V mark OFF"softkeys are used to hide/show the vertical gridlines. Using the "Select signal" function you can determine the signal to be displayed in the vertical axis.

The "T mark ON" / "T mark OFF"softkeys are used to hide/show the horizontal gridlines of the time axis.

Fix V mark

Use the markers to determine the differences in the horizontal or vertical directions. To do this, place the marker on the start point and press "Fix V mark" or "Fix T mark". The difference between the starting point and the current marker position is now displayed in the status bar. The softkey labels will change to "Free V mark" or "Free T mark".

Trace display

This function opens another menu level offering softkeys for hiding / displaying the diagrams. If a softkey is displayed on a black background, the diagrams are displayed for the selected trace channel.

Time scale +

Use this function to zoom in / zoom out the time basis.

Vertical scale +

Use this function to increase / reduce the resolution (amplitude).

Marker steps

Use these softkeys to define the step sizes of the markers.

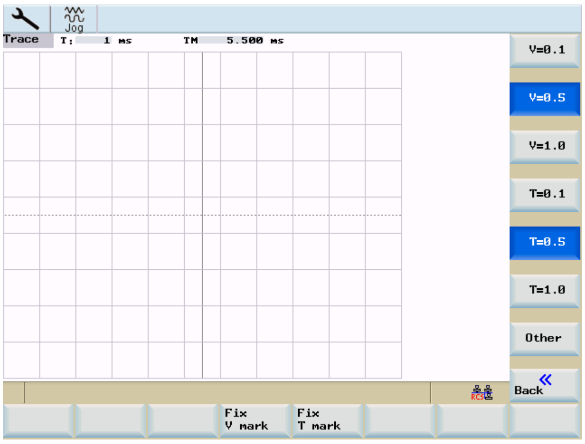
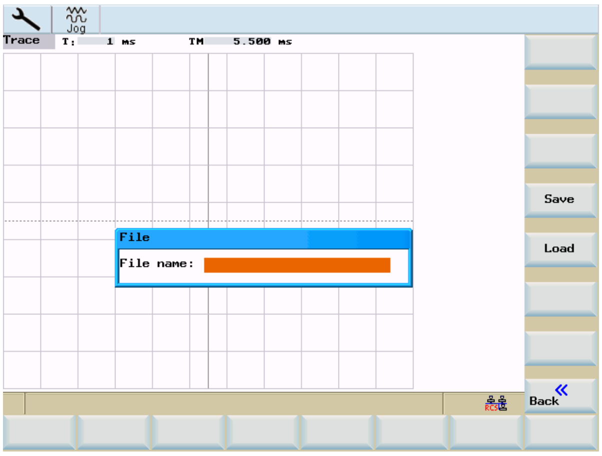


Figure 8-21 Marker steps

The markers are moved using the cursor keys at a step size of one increment. larger step sizes can be set using the input fields. The value specifies how many grid units the marker must be moved per "SHIFT" + cursor movement. When a marker reaches the margin of the diagram, the grid automatically appears in the horizontal or vertical direction.

File service

Use this softkey to save or load trace data.


File

File name:

Figure 8-22 Trace data

Type the desired file name without extension in the "File name" field.

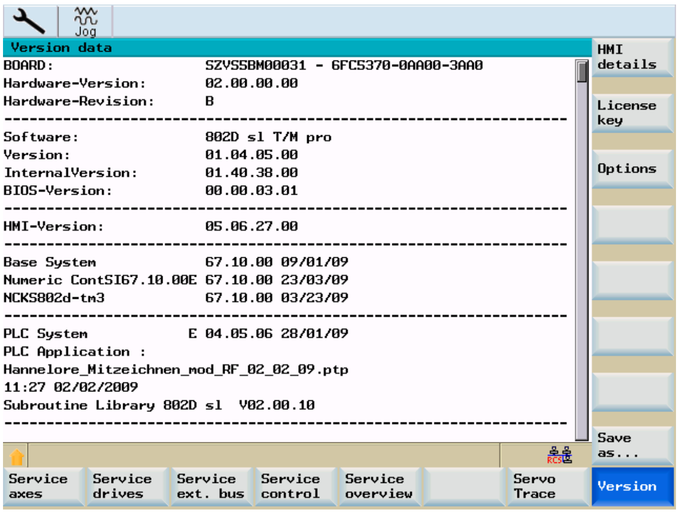
Use the "Save" softkey to save the data with the specified name in the part program directory. Thereafter, the file can be exported, and the data can be processed in MS Excel.

"Load" loads the specified file and graphically displays the data.

8.4.3 Version/HMI details

Version

This window displays the version numbers and the date of creation of the individual CNC components.



```
BOARD:                   SZVS5BM00031 - 6FC5370-0AA00-3AA0
Hardware-Version:        02.00.00.00
Hardware-Revision:       B
--------------------------------------------------------------------
Software:                802D sl T/M pro
Version:                 01.04.05.00
InternalVersion:         01.40.38.00
BIOS-Version:            00.00.03.01
--------------------------------------------------------------------
HMI-Version:             05.06.27.00
--------------------------------------------------------------------
Base System              67.10.00 09/01/09
Numeric ContSI67.10.00E 67.10.00 23/03/09
NCKS802d-tm3             67.10.00 03/23/09
--------------------------------------------------------------------
PLC System             E 04.05.06 28/01/09
PLC Application :
Hannelore_Mitzeichnen_mod_RF_02_02_09.ptp
11:27 02/02/2009
Subroutine Library 802D sl  V02.00.10
--------------------------------------------------------------------
```

Figure 8-23 Version

Note

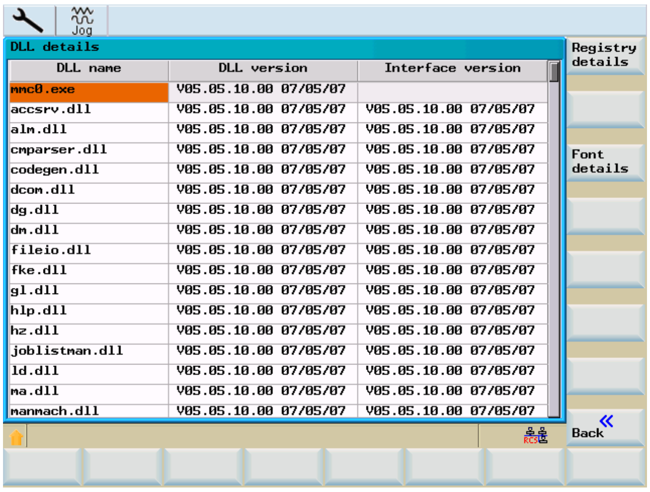
The version releases shown in the version screen shot are for example only.

Save under

Saves the contents of the "Version" window in a text file. The target (e.g. "customer CF card") can be selected.

HMI Details

The "HMI details" menu is intended for servicing and can only be accessed via the user password level. All programs provided by the operator unit are displayed with their version numbers. By reloading software components, the version numbers can be differ from each other.



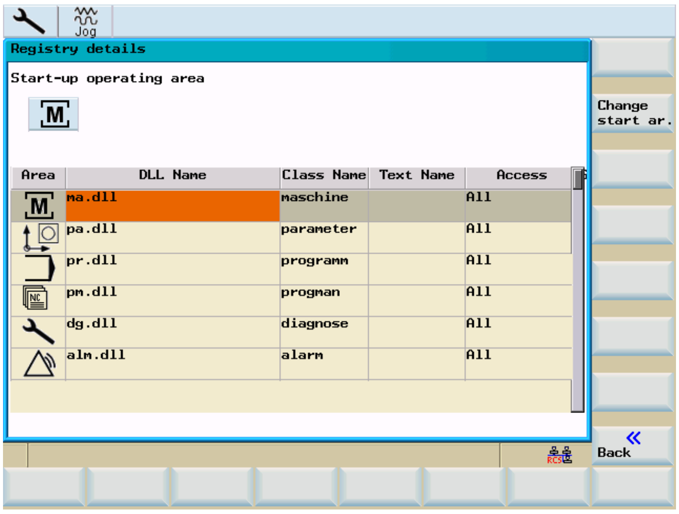
DLL details

DLL name	DLL version	Interface version
mmc0.exe	V05.05.10.00 07/05/07	
accsrv.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
alm.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
cmparser.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
codegen.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
dcom.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
dg.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
dm.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
fileio.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
fke.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
gl.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
hlp.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
hz.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
joblistman.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
ld.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
ma.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
manmach.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07

Figure 8-24 The "HMI version" menu area

Registry Details

This "Registry details" function displays the assignment of the hard keys (operating area keys POSITION (machine), OFFSET PARAM (parameter), PROGRAM (program), PROGRAM MANAGER (progman), ...) for the programs to be started in the form of a list. For the meanings of the individual columns, please refer to the table below.



Registry details

Start-up operating area

Area	DLL Name	Class Name	Text Name	Access
M	ma.dll	maschine		All
	pa.dll	parameter		All
	pr.dll	programm		All
NC	pm.dll	progman		All
	dg.dll	diagnose		All
	alm.dll	alarm		All

Figure 8-25 Registry details

M

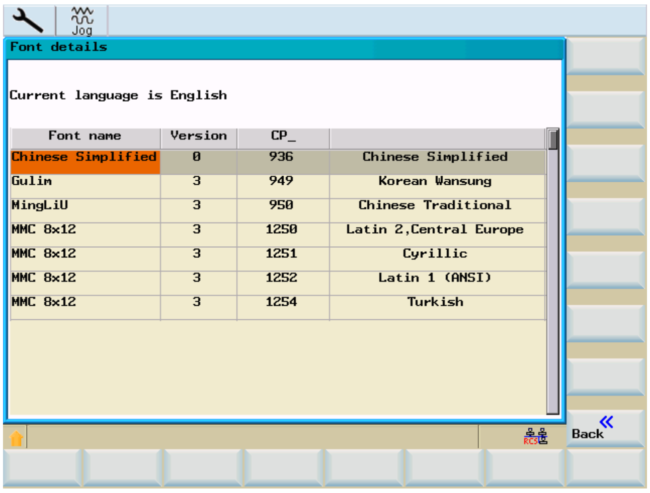
Note

After the system has booted, the control system automatically starts the <POSITION> operating area. If a start behavior is required, the "Change ready to start" function allows defining another starting program.

The starting operating area is then displayed above the table in the "Registry Details" window.

Font Details

The "Font details" function displays the data of the loaded character sets in the form of a list.



Font details

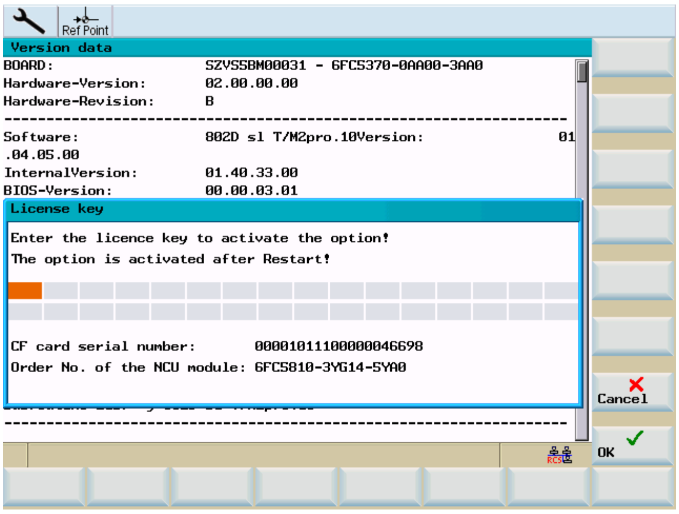
Current language is English

Font name	Version	CP_	
Chinese Simplified	0	936	Chinese Simplified
Gulim	3	949	Korean Wansung
MingLiU	3	950	Chinese Traditional
MMC 8x12	3	1250	Latin 2,Central Europe
MMC 8x12	3	1251	Cyrillic
MMC 8x12	3	1252	Latin 1 (ANSI)
MMC 8x12	3	1254	Turkish

Figure 8-26 Font details

license key

Entering the license key.



```
Version data
BOARD:                  SZVS5BM00031 - 6FC5370-0AA00-3AA0
Hardware-Version:       02.00.00.00
Hardware-Revision:      B
-----------------------------------------------------------------
Software:               802D sl T/M2pro.10Version:              01
.04.05.00
InternalVersion:        01.40.33.00
BIOS-Version:           00.00.03.01
```

```
License key
Enter the licence key to activate the option!
The option is activated after Restart!

CF card serial number:        00001011100000046698
Order No. of the NCU module: 6FC5810-3YG14-5YA0
```

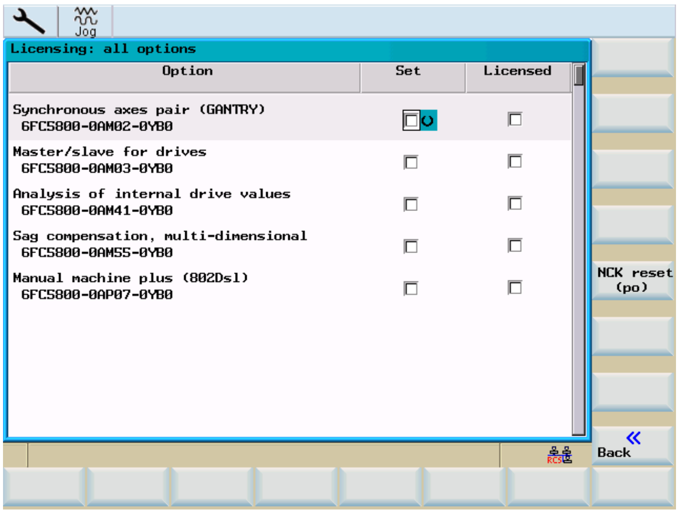
Figure 8-27 License key

References

SINUMERIK 802D sl Operating Instructions for Turning, Milling, Grinding, Nibbling; Licensing in SINUMERIK 802D sl

Options

Setting the licensed options.



Licensing: all options

Option	Set	Licensed
Synchronous axes pair (GANTRY) 6FC5800-0AM02-0YB0	□	□
Master/slave for drives 6FC5800-0AM03-0YB0	□	□
Analysis of internal drive values 6FC5800-0AM41-0YB0	□	□
Sag compensation, multi-dimensional 6FC5800-0AM55-0YB0	□	□
Manual machine plus (802Dsl) 6FC5800-0AP07-0YB0	□	□

Figure 8-28 Options

References

SINUMERIK 802D sl Operating Instructions for Turning, Milling, Grinding, Nibbling; Licensing in SINUMERIK 802D sl

NCK reset (po)

Executes a warm restart at the control.

8.4.4 Service MSG

Service MSG

The "Service MSG" function allows message texts/messages to be output via the following interfaces:

- Output via the RS232 interface (V24) as data stream without protocol
- Output in a file

Message texts/messages include:

- Alarms
- Texts of MSG commands

The message texts/messages are programmed in the part program using a specified syntax. The particular syntax is described in the following table:

Table 8- 2 Syntax of the message texts/messages

Output	Syntax ("<interface>: Message text")
via RS232 interface (V24)	MSG ("V24: Message text")
in a file	MSG ("File: Message text")
Alarm line at the HMI	MSG ("Alarm text")

The MSG text output is defined using the MSG command as well as by appropriately parameterizing the output interface. For the alarm output, only the output interface has to be taken into consideration.

If the information line "Processing error MSG command occurred" is output, then the error protocol can be evaluated under the operating area <SYSTEM> > "Service display" > "Service control" > "Service MSG" > "Error protocol".

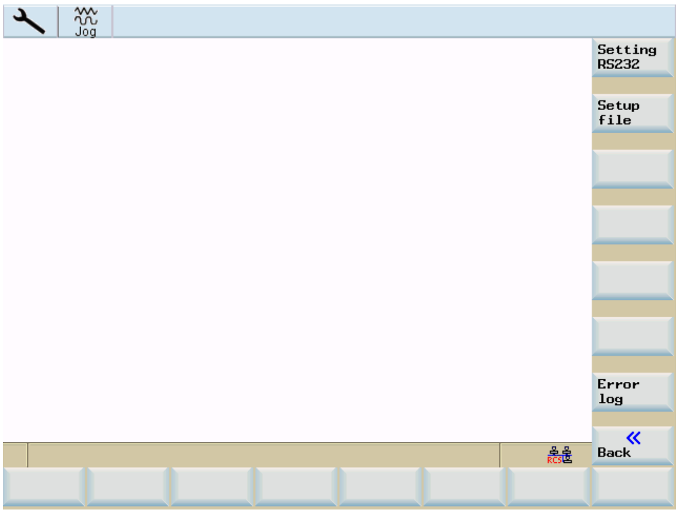
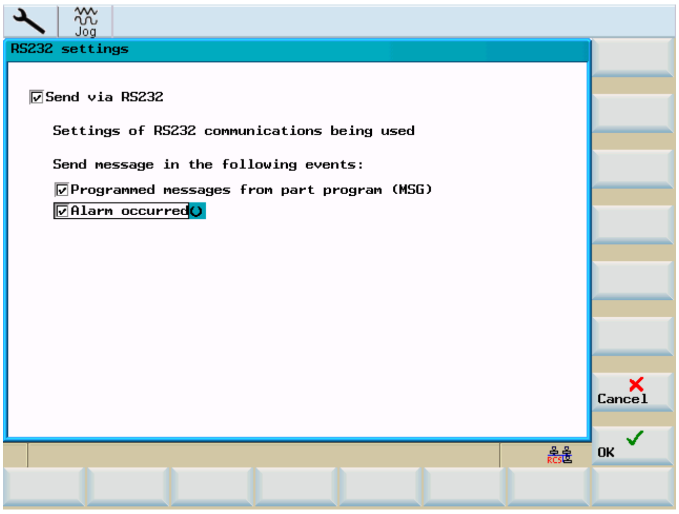


Figure 8-29 Dialog box, Service MSG

Settings for output via the RS232 interface

Setting RS232

Settings of the RS232 output interface.



RS232 settings

☑ Send via RS232

Settings of RS232 communications being used

Send message in the following events:

☑ Programmed messages from part program (MSG)

☑ Alarm occurred

Figure 8-30 Dialog box, RS232 interface settings

"Sending messages via this interface can be activated or deactivated using the "Send via RS232" checkbox. Incoming messages are ignored when the interface is deactivated!

Note

When transferring a file via a serial interface (RS232), please note the end of transmission character for RS232 communication (analog to the communication setting, RS232 on HMI).

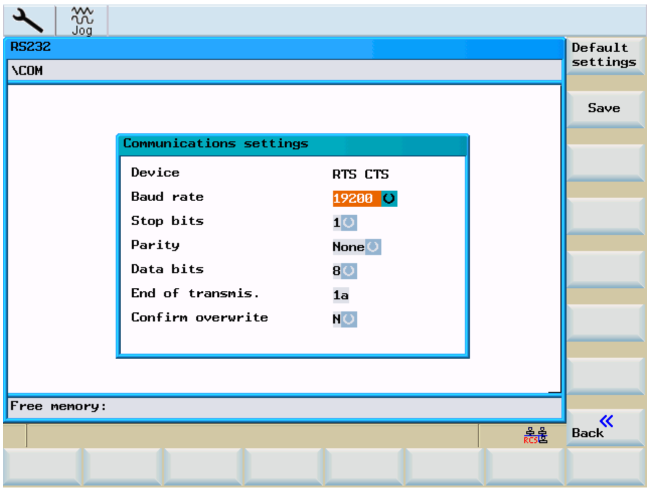
Further, when sending via RS232, it can be defined as to which messages are sent for which events:

- Programmed messages from the part program
- An alarm has occurred

The settings are saved and the dialog box exited by pressing the "OK" softkey.

The dialog box is exited without saving by pressing "Cancel".

To transfer messages via the RS232 interface, the communication settings from the operating area <SYSTEM> > "Start-up files" > "RS232" > "Settings" are used.



Communications settings

Device RTS CTS
Baud rate 19200
Stop bits 1
Parity None
Data bits 8
End of transmis. 1a
Confirm overwrite N

Figure 8-31 Parameters of the RS232 interface

Note

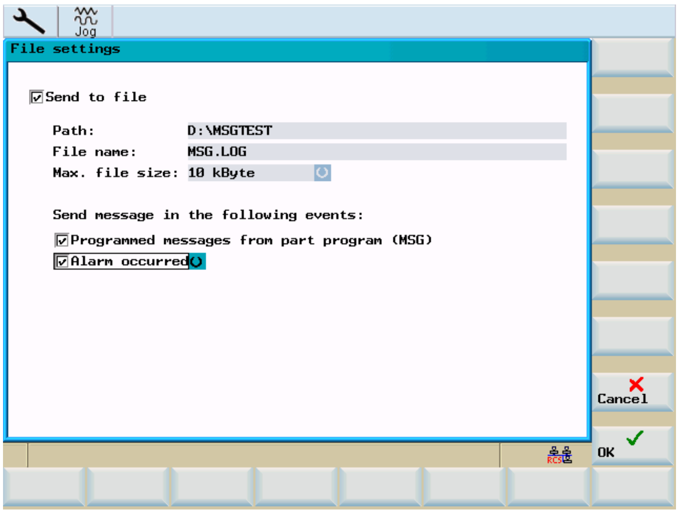
When using the MSG service via RS232, the RS232 interface must not be active for another application.

The means, e.g. the RS232 interface must not be active from the operating area <SYSTEM> "PLC" > "Step7 connect."

Settings to output in a file

Setting File

Settings for the file storage location.



File settings

☑ Send to file

Path: D:\MSGTEST
File name: MSG.LOG
Max. file size: 10 kByte

Send message in the following events:

☑ Programmed messages from part program (MSG)
☑ Alarm occurred

Figure 8-32 Dialog box, file settings

Sending messages to the selected file is activated or deactivated using the "Send to file" checkbox. When the interface is deactivated, messages are not output and the information line "Processing error MSG command occurred".

A path, the file name and the max. size of the file can be selected.

Drive D (customer CF card), F: (USB drive) and the drive connected per RCS connection can be selected in the "Path" input field.

10kByte, 100kByte and 1MByte can be selected as max. file size. When the max. size is reached, the file is written as ring buffer, i.e. at the beginning, as many lines are deleted line-by-line as is required by the new message at the end of the file.

Here, when sending to a file, it can be defined as to which messages are sent for which events:

- Programmed messages from the part program
- An alarm has occurred

The settings are saved and the dialog box exited by pressing the "OK" softkey.

The dialog box is exited without saving by pressing "Cancel".

Error log

Error log

Error log display.

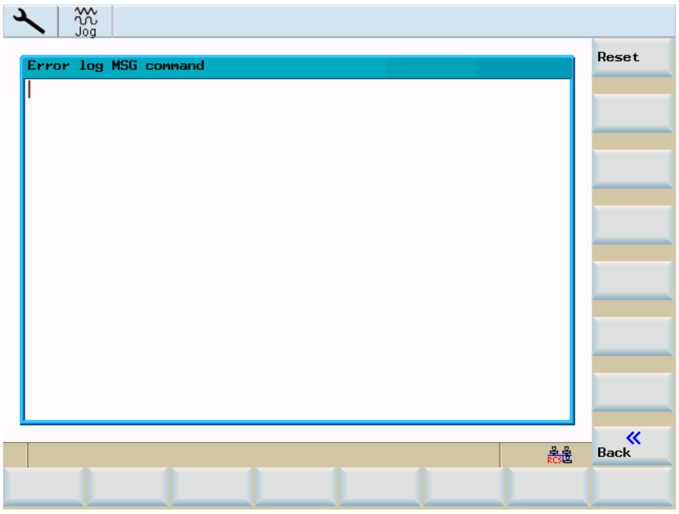


Figure 8-33 Dialog box, error log

All messages with the associated error information, where an error occurred when processing them, are saved in the error log.

The error log can be deleted using the "reset" softkey.

The dialog box is closed by pressing "Back".

Note

The error log can be used for analysis when the information line "Processing error MSG command occurred" is output.

Example of programming using the "MSG" command

For SINUMERIK 802D sl, messages programmed in the NC program are displayed in the alarm display as standard.

Table 8- 3 Activating/deleting messages

```
N10 MSG ("Roughing the contour")             ; The text "Roughing the contour"
                                             is displayed in the alarm display
N20 X… Y… N …
N…
N90 MSG ()                                   ; Delete message from the alarm
                                             display
```

Table 8- 4 Message text contains a variable

```
N10 R12=$AA_IW[X]                           ; Actual position of the X axis in
                                            R12
N20 MSG ("Check position of X axis"<<R12<<) ; Activate message
N20 X… Y… N …
N…
N90 MSG ()                                  ; Delete message from the alarm
                                            display
```

To output messages to other interfaces, an additional command is located in front of the actual message text that defines the output interface of this message.

Table 8- 5 Messages to the RS232 output interface

```
N20 MSG ("V24: Roughing the contour")       ; The text "Roughing the contour" is
                                            sent in the ASCII format via the
                                            RS232 interface
```

Table 8- 6 Messages to the output interface file

```
N20 MSG ("FILE: Roughing the contour")      ; The text "Roughing the contour" is
                                            sent to the selected file
```

Note

If, in the part program, the text for the messages is repeated unchanged, then after each output, a command for an empty text must be entered.

e.g.

```
...
MSG("<interface>: Sample text")
MSG("<interface>:")
...
...
MSG("<interface>: Sample text")
MSG("<interface>:")
...
...
MSG("<interface>: Sample text")
MSG("<interface>:")
```

8.5 SYSTEM - "PLC" softkeys

PLC

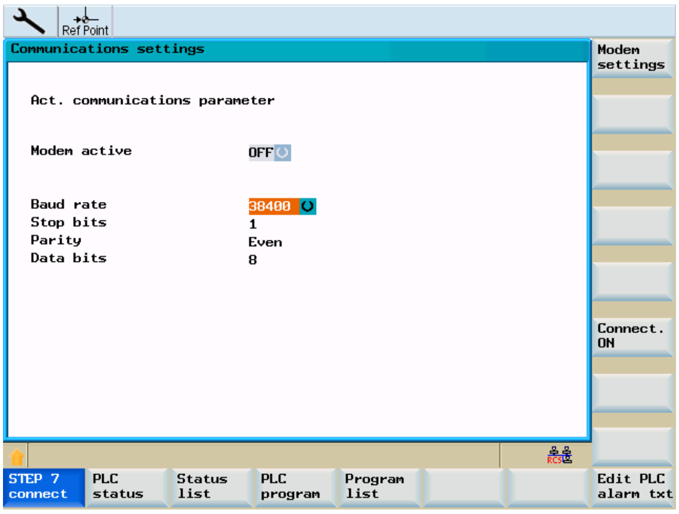
This softkey provides further functions for diagnostics and commissioning of the PLC.

Connect to STEP 7

This softkey opens the configuration dialog for the interface parameters of the STEP 7 connection using the RS232 interface of the control system.

If the RS232 interface is already occupied by the data transfer, you can connect the control system to the PLC802 programming tool on the programming device/PC only if the transmission is completed.

The RS232 interface is initialized with activation of the connection.



Communications settings

Act. communications parameter

Modem active OFF

Baud rate 38400
Stop bits 1
Parity Even
Data bits 8

Figure 8-34 Communication settings

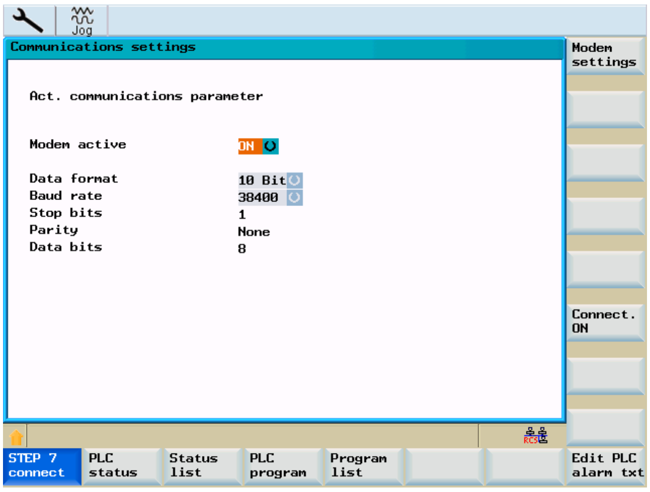
The baud rate is set using the toggle field. The following values are possible: 9600 / 19200 / 38400 / 57600 / 115200.

Note

The appropriate connection symbol is displayed at the bottom right after the connection has been established. The communication setting can then no longer be changed.

Modem

If the data transfer is performed on the RS232 interface via modem, start with the following initialization option:



```
Act. communications parameter

Modem active          ON
Data format           10 Bit
Baud rate             38400
Stop bits             1
Parity                None
Data bits             8
```

Figure 8-35 Initialize the modem

The following initializations are possible via toggle fields:

- Baud rate

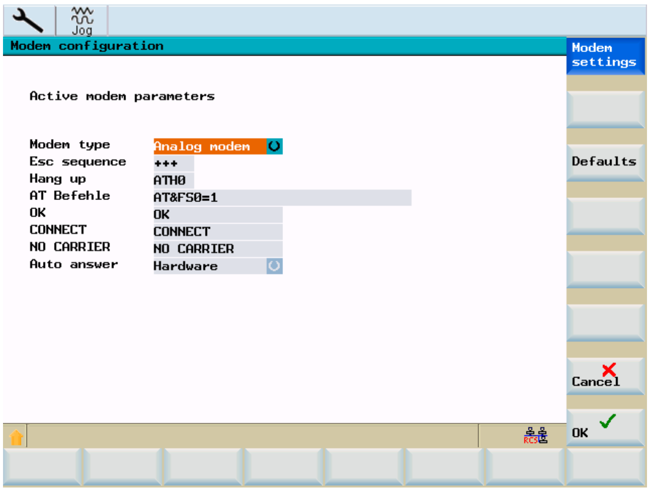
  9600 / 19200 / 38400 / 57600 / 115200.

- Parity:

  "without" for 10 bit

  "odd" for 11 bit

Using the "Modem settings" softkey you can make the following additional settings for a connection that does not yet exist:



```
Active modem parameters

Modem type       Analog modem
Esc sequence     +++
Hang up          ATH0
AT Befehle       AT&FS0=1
OK               OK
CONNECT          CONNECT
NO CARRIER       NO CARRIER
Auto answer      Hardware
```

Figure 8-36 Modem settings

You can select the following modem types via toggle field:

- Analog modem
- ISDN box
- Mobile phone

Note

The types of both communication partners must match with each other.

When you want to enter several AT command sets, you have to start with AT only once and simply have to add all other commands, e.g. AT&FS0=1E1X0&W.

Refer to the manufacturers' manuals to look up the commands and their parameters, as they sometimes differ even between the devices of one manufacturer. The default values of the control system are therefore only a real minimum and should be verified in any case before they are used for the first time.

Connect.
on

Use this softkey to activate the connection between the control system and the programming device/PC. The program waits for the call of Programming Tool PLC802. No modifications to the settings are possible in this state.

The softkey label changes to "Connection inactive".

By pressing "Connection inactive", the transfer from the control system can be terminated at any point. Now it is possible again to make changes in the settings.

The active or inactive state is kept even after Power On (except power-up with the default data). An active connection is displayed by a symbol in the status bar.

Press "RECALL" to exit the menu.

Additional functions

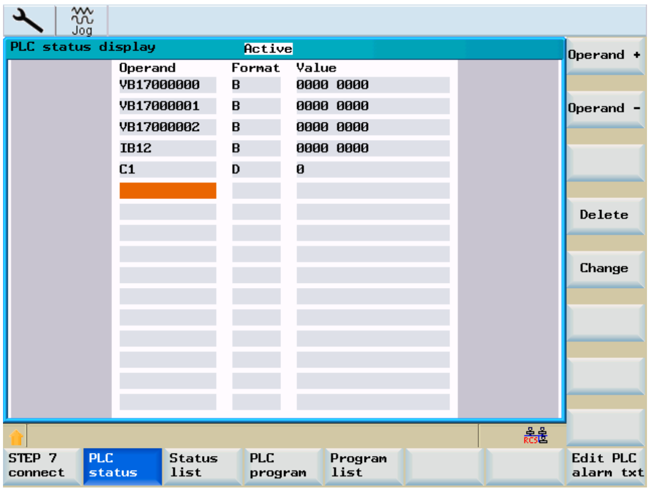
PLC
status

Use this function to display and change the current states of the memory areas listed in the following table.

It is possible to display 16 operands at the same time.

Table 8- 7 Memory areas

Inputs	I	Input byte (IBx), input word (Iwx), input double-word (IDx)
Outputs	Q	Output byte (Qbx), output word (Qwx), output double-word (QDx)
Flags	M	Flag byte (Mx), flag word (Mw), flag double-word (MDx)
Times	T	Time (Tx)
Meters	C	Counter (Cx)
data	V	Data byte (Vbx), data word (Vwx), data double-word (VDx)
Format	B H D	Binary Hexadecimal Decimal
		The binary representation is not possible with double words. Counters and timers are represented decimally.



Operand	Format	Value
VB17000000	B	0000 0000
VB17000001	B	0000 0000
VB17000002	B	0000 0000
IB12	B	0000 0000
C1	D	0

Figure 8-37 PLC status display

Operand +

The operand address displays the value incremented by 1.

Operand -

The operand address displays the value respectively decremented by 1.

Delete

Use this softkey to delete all operands.

Change

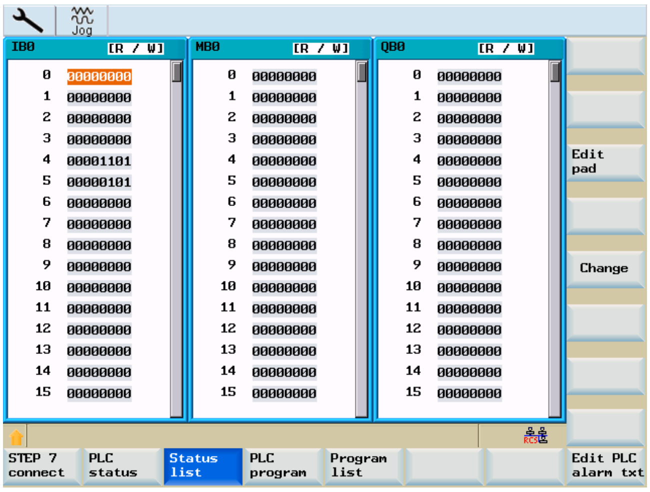
Cyclic updating of the values is interrupted. Then you can change the values of the operands.

Status list

Use the "Status list" function to display and modify PLC signals.

There are 3 lists to choose from:

- Inputs (default setting) left-hand list
- Flags (default setting) center list
- Outputs (default setting) right-hand list
- Variable



IB0	[R / W]	MB0	[R / W]	QB0	[R / W]
0	00000000	0	00000000	0	00000000
1	00000000	1	00000000	1	00000000
2	00000000	2	00000000	2	00000000
3	00000000	3	00000000	3	00000000
4	00001101	4	00000000	4	00000000
5	00000101	5	00000000	5	00000000
6	00000000	6	00000000	6	00000000
7	00000000	7	00000000	7	00000000
8	00000000	8	00000000	8	00000000
9	00000000	9	00000000	9	00000000
10	00000000	10	00000000	10	00000000
11	00000000	11	00000000	11	00000000
12	00000000	12	00000000	12	00000000
13	00000000	13	00000000	13	00000000
14	00000000	14	00000000	14	00000000
15	00000000	15	00000000	15	00000000

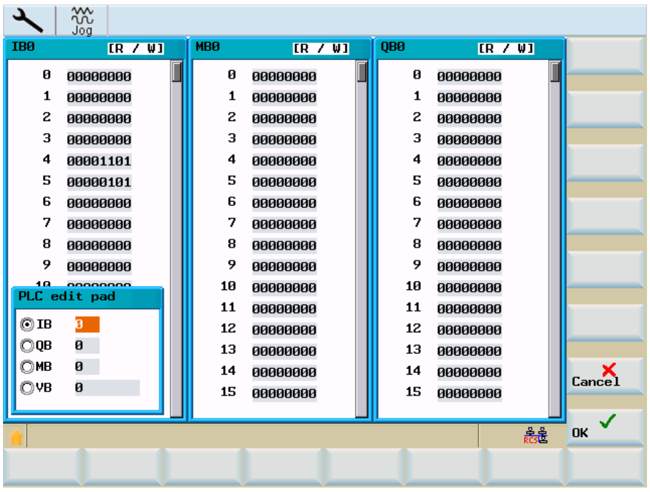
Figure 8-38 PLC status list

Change

Use this softkey to change the value of the highlighted variable. Press the "Accept" softkey to confirm your changes.

Edit
pad

Use this softkey to assign the active column a new area. To this end, the interactive screenform offers four areas to choose from. For each column, a start address can be assigned which must be entered in the relevant input field. When you quit the interactive screenform, the control system will save your settings.



	IB0 [R / W]		MB0 [R / W]		QB0 [R / W]
0	00000000	0	00000000	0	00000000
1	00000000	1	00000000	1	00000000
2	00000000	2	00000000	2	00000000
3	00000000	3	00000000	3	00000000
4	00001101	4	00000000	4	00000000
5	00000101	5	00000000	5	00000000
6	00000000	6	00000000	6	00000000
7	00000000	7	00000000	7	00000000
8	00000000	8	00000000	8	00000000
9	00000000	9	00000000	9	00000000
10	00000000	10	00000000	10	00000000
		11	00000000	11	00000000
		12	00000000	12	00000000
		13	00000000	13	00000000
		14	00000000	14	00000000
		15	00000000	15	00000000

PLC edit pad

● IB 0
○ QB 0
○ MB 0
○ VB 0

Figure 8-39 The "Data type" selection screen

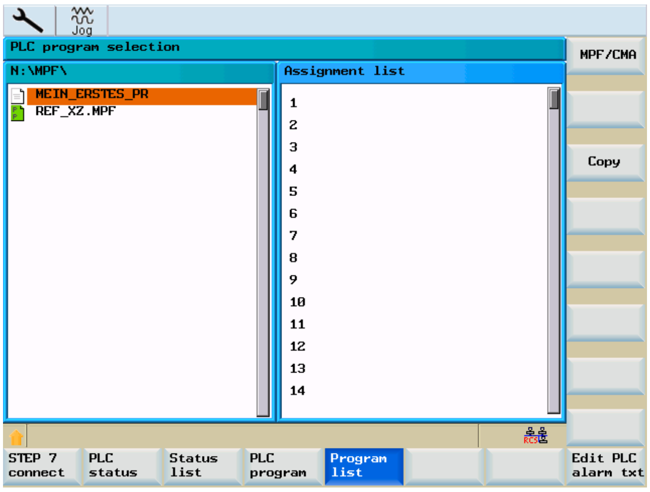
Use the cursor keys and the "Page Up" / "Page Down" keys to navigate in and between the columns.

PLC
program

PLC diagnosis using a ladder diagram (see chapter "PLC diagnosis using a ladder diagram").

Program
list

Using the PLC, you may select part programs and run them via the PLC. To this end, the PLC user program writes a program number to the PLC interface, which is then converted to a program name using a reference list. It is possible to manage max. 255 programs.



PLC program selection

N:\MPF\

MEIN_ERSTES_PR
REF_XZ.MPF

Assignment list

1
2
3
4
5
6
7
8
9
10
11
12
13
14

Figure 8-40 PLC program list

This dialog displays all files of the MPF directory and their assignment in the reference list (PLCPROG.LST) in the form of a list. You can use the TAB key to switch between the two columns. The Copy, Insert and Delete softkey functions are displayed with reference to a specific context. If the cursor is placed on the left-hand side, only the Copy function is available. On the right-hand side, the Insert and Delete functions are provided to modify the reference list.

List of references for interface signals

SINUMERIK 802D sl Function Manual; Various Interface Signals (A2)

SINUMERIK 802D sl List Manual

Copy

Writes the selected file name to the clipboard.

Paste

Pastes the file name at the current cursor position.

Delete

Deletes the selected file name from the assignment list.

Structure of the reference list (file PLCPROG.LST)

It is divided into 3 areas:

Number	range	Protection level
1 to 100	User area	User
101 to 200	Machine manufacturer	Machine manufacturer
201 to 255	Siemens	Siemens

The notation is carried out for each program by lines. Two columns are intended per line, which must be separated from each other by TAB, space or the "|" character. In the first column, the PLC reference number must be specified, and in the second column, the file name.

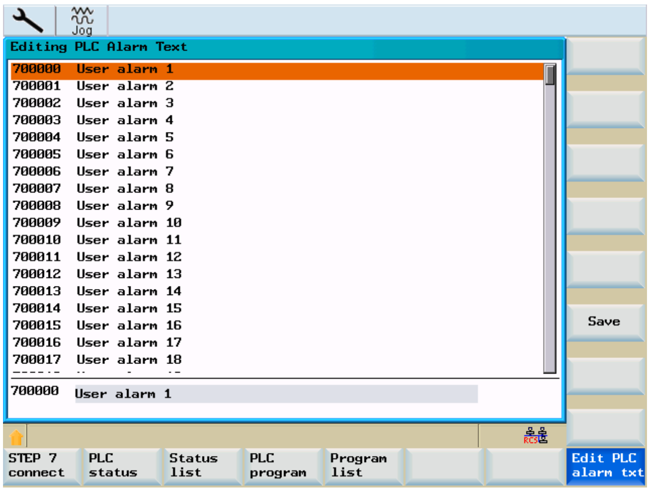
Example:

1 | shaft.mpf

2 | taper.mpf

Edit PLC alarm txt

This function can be used to insert or modify PLC user alarm texts. Select the desired alarm number using the cursor. At the same time, the text currently valid is displayed in the input line.



Editing PLC Alarm Text

```
700000  User alarm 1
700001  User alarm 2
700002  User alarm 3
700003  User alarm 4
700004  User alarm 5
700005  User alarm 6
700006  User alarm 7
700007  User alarm 8
700008  User alarm 9
700009  User alarm 10
700010  User alarm 11
700011  User alarm 12
700012  User alarm 13
700013  User alarm 14
700014  User alarm 15
700015  User alarm 16
700016  User alarm 17
700017  User alarm 18

700000  User alarm 1
```

Figure 8-41 Editing the PLC alarm text

Enter the new text in the input line. Press the "Input" key to complete your input and select "Save" to save it.

For the notation of the texts, please refer to the operating instructions.

8.6 SYSTEM - "Start-up files" softkeys

Start-up files

The menu allows general files, commissioning archives and PLC projects to be created, read-out or read-in, copied, deleted etc.

This window displays the contents of the selected drive in a tree structure. The horizontal softkeys display the available drives for selection in the form of a list. The vertical softkeys provide the control functions possible for the drive in question.

There are the following permanently set drive assignments:

- 802D data: Commissioning data
- Customer CF card: Customer data on the CF card
- RCS connection: Data of a drive released on PC/PG via the the RCS tool (only for SINUMERIK 802D sl pro)
- RS232: Serial Interface
- Manufacturer drive: Data that the manufacturer specifically stored
- USB drive: Customer data on a USB FlashDrive
- Manufacturer archive: Commissioning data archived on the system CompactFlash Card

All data is handled using the "Copy & Paste" principle.

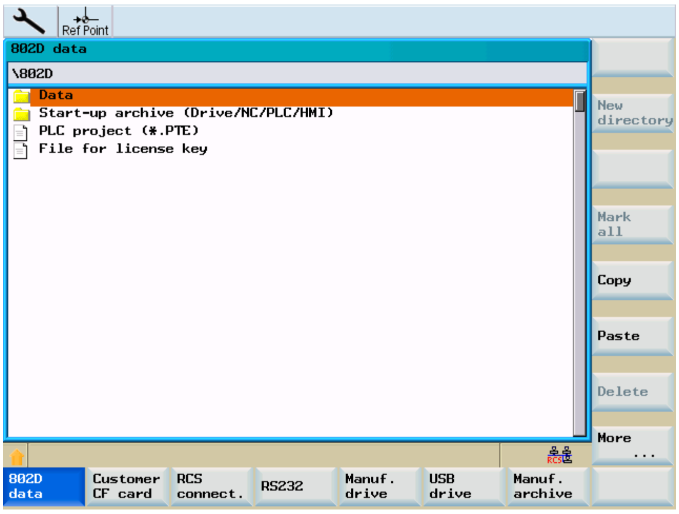


Figure 8-42 Start-up files

802D data

The individual data groups in the "802D data" area have the following significance:

Note

The sag compensation is ONLY listed if the associated function was activated.

- Data (in text format)

  These data are special initialization data and are transferred in an ASCII file.
  - Machine data
  - Setting data
  - Tool data
  - R parameters
  - Work offset
  - Leadscrew error compensation
  - Sag compensation
  - Global user data
- Commissioning archive (drive/NC/PLC/HMI)

  These data constitute a commissioning file for HMI data and are transferred in the binary format using the HMI archive format.
  - Drive machine data
  - NC data
  - NC directories
  - Display machine data
  - Leadscrew error compensation
  - Sag compensation
  - PLC project
  - HMI data and applications
- PLC project (*.PTE)

  A direct exchange between the control system and programming tool is possible without conversion with the support of PLC project handling in the programming tool export format.
- File for license key

Customer CF card

Reading-in and reading-out data on a CompactFlash Card (CF card).

RCS connect.

Reading-in and reading-out data to a PG/PC via a network. The RCS tool must be installed on the PG/PC (only for SINUMERIK 802D sl pro).

Note

The RCS tool provides a detailed online help function. Refer to this help menu for further details e.g. establishing a connection, project management etc.

RS232

Reading-in and reading-out data via the RS232 interface.

More ...

Error log

Note

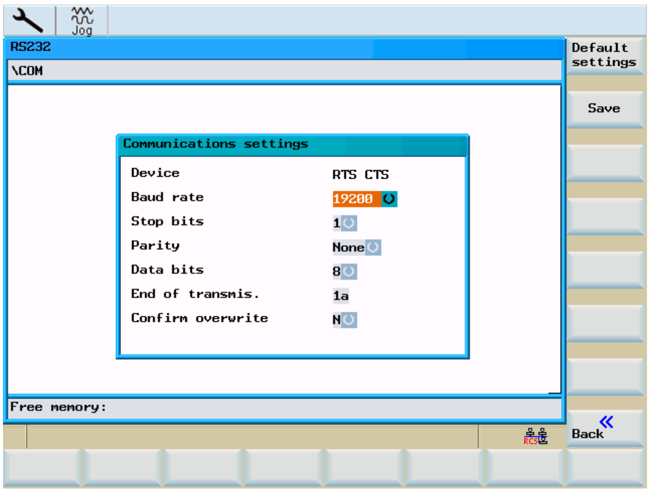
Using the softkey function "Continue...", you may also inspect the transmission log. The "Error log" function is available for that.

Set-ting.

Use this function to display and change the RS232 interface parameters. Any changes in the settings come into effect immediately.

Selecting the "Save" softkey will save the selected settings even beyond switching off.

The "Default settings" softkey will reset all settings to their default settings.



RS232
\COM

Communications settings

Device	RTS CTS
Baud rate	19200
Stop bits	1
Parity	None
Data bits	8
End of transmis.	1a
Confirm overwrite	N

Free memory:

Figure 8-43 Parameters of the RS232 interface

Interface parameters

Table 8- 8 Interface parameters

Parameter	Description
Device type	**RTS CTS** The signal RTS (Request to Send) controls the send mode of the data transfer device. The CTS signal indicates the readiness to transmit data as the acknowledgment signal for RTS.
Baud rate	... used to set the interface transmission rate. 300 baud 600 baud 1200 baud 2400 baud 4800 baud 9600 baud 19200 baud 38400 baud 57600 baud 115200 baud
Stop bits	Number of stop bits with asynchronous transmission Input: 1 stop bit (default setting) 2 stop bits
Parity	Parity bits are used for error detection. These are added to the coded character to convert the number of digits set to "1" into an odd or even number. Input: No parity (default setting) Even parity Odd parity
Data bits	Number of data bits with asynchronous transmission Input: 7 data bits 8 data bits (default)
Overwriting with confirmation	Y: When reading in, it is checked whether the file already exists in the NC. N: The files are overwritten without confirmation warning.

Manufac-turer drive

Reading-in and reading-out data of the manufacturer's directory "F".

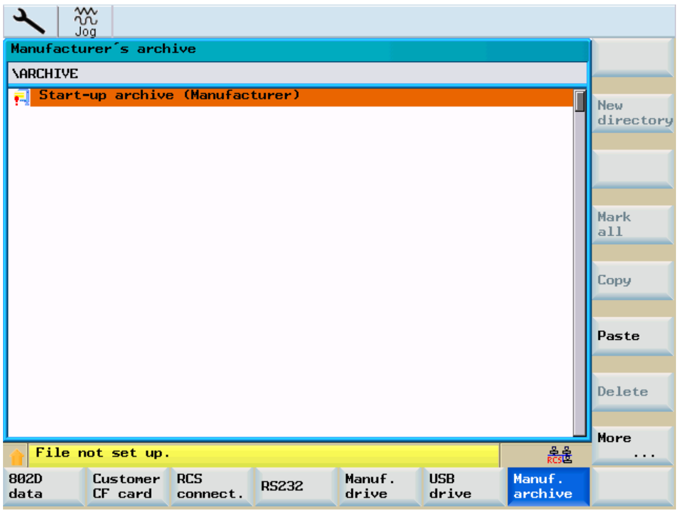
USB drive

Reading-in and reading-out data of a USB FlashDrive.

manu.
archive

Use this function to create/restore a commissioning archive on/from the system CompactFlash Card.

No archive file has been created in the following display. The symbol for the zip archive sends a signal with an exclamation mark.



File not set up.

Figure 8-44 Manufacturers` archive, archive file not yet created

Vertical softkeys

The following vertical softkeys are available upon activating the file functions:

- "Rename": Use this function to rename a file selected beforehand using the cursor.
- "New directory": Creates a new directory
- "Copy": Use this softkey to copy one or more files to the clipboard.
- "Paste": Use this softkey to paste files or directories from the clipboard to the current directory.
- "Delete": Deletes the selected file name from the assignment list.
- "Select all": Use this softkey to select all files for subsequent operations.
- "Properties": Display memory capacity.
- "Job list": Displays a list with active file jobs and provides the option to terminate or display a file job.

More
...

Use this function to switch to the respective vertical softkeys.

Note

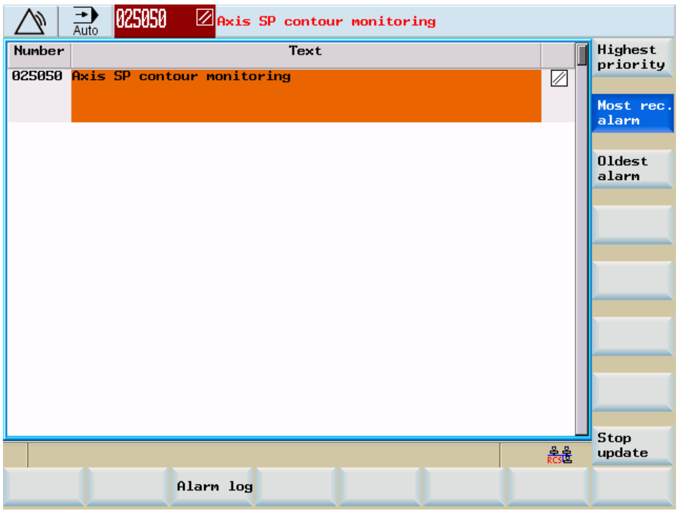
If individual functions are grayed out, then these functions are not available for the displayed drive/directory.

8.7 Alarm display

Operating sequence

SYSTEM ALARM

The alarm window is opened. You can sort the NC alarms using softkeys; PLC alarms will not be sorted.



025050 Axis SP contour monitoring

Number	Text
025050	Axis SP contour monitoring

Figure 8-45 Alarm display window

Softkeys

Highest priority

Use this softkey to display all alarms sorted by their priorities. The highest priority alarm is at the beginning of the list.

Most rec. alarm

Use this softkey to display the alarms sorted by the time of their occurrence. The most recent alarm stands at the beginning of the list.

Oldest alarm

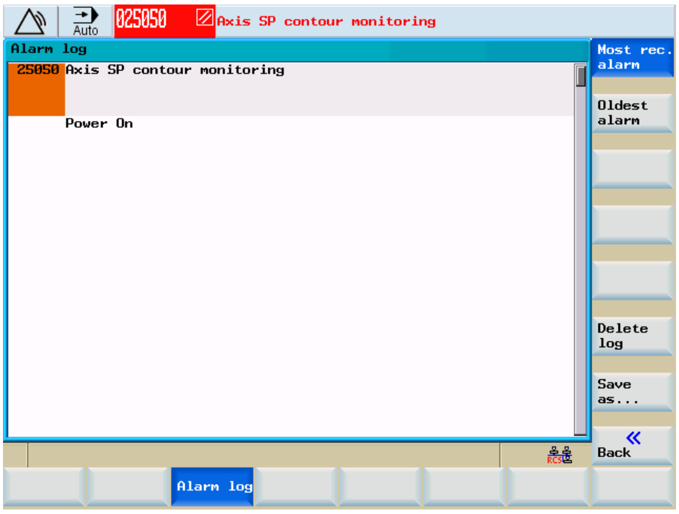
Use this softkey to display the alarms sorted by the time of their occurrence. The oldest alarm stands at the beginning of the list.

Stop updating

Updating of pending alarms is stopped / started.

Alarm log

All alarms are logged.



025050 Axis SP contour monitoring

Alarm log

25050 Axis SP contour monitoring

Power On

Figure 8-46 Alarm log

The log is deleted using softkey "Delete log".

Save under

The file is output using softkey "Save under..." on a CF card or on the USB FlashDrive.

Programming 9

9.1 Fundamental Principles of NC Programming

9.1.1 Program names

Each program has its own program name. The name can be freely chosen during program creation, taking the following conventions into account:

- The first two characters must be letters;
- Use only letters, digits or underscore.
- Do not use delimiters (see Section "Character set").
- The decimal point must only be used for separation of the file extension.
- Do not use more than 27 characters.

Example

WORKPIECE527

9.1.2 Program structure

Structure and contents

The NC program consists of a sequence of **blocks** (see Table below).

Each block represents a 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**: e.g. **M2**.

Table 9- 1 NC program structure

Set	Word	Word	Word	...	; Comment
Set	N10	G0	X20	...	; 1. Set
Set	N20	G2	Z37	...	; 2. Set
Set	N30	G91	...	...	; ...
Set	N40	...	...	...	
Set	N50	M2			; End of program

9.1.3 Word structure and address

Functionality/structure

A word is a block element and mainly constitutes a control command. The word consists of

- **address character**: generally a letter
- **numerical value**: a sequence of digits which with certain addresses can be added by a sign put in front of the address, and a decimal point.

  A positive sign (+) can be omitted.

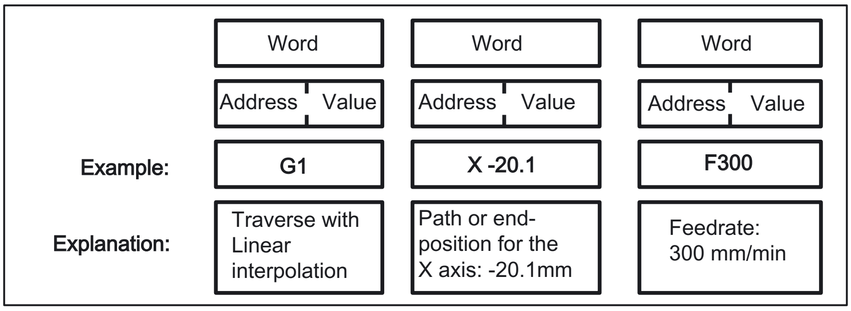


Figure 9-1 Word structure (example)

Several address characters

A word can also contain several address letters. In this case, however, the numerical value must be assigned via the intermediate character "=".
Example: **CR=5.23**

Additionally, it is also possible to call G functions using a symbolic name (see also section "Overview of instructions").
Example: SCALE ; Enable scaling factor

Extended address

With the addresses

R	Arithmetic parameters
H	H function
I, J, K	Interpolation parameters/intermediate point
M	Special function M, only affecting the spindle
S	Spindle speed (Spindle 1 or 2)

the address is extended by 1 to 4 digits to obtain a higher number of addresses. In this case, the value must be assigned using an equality sign "=" (see also section "List of instructions").

Table 9- 2 Examples:

R10=6.234	H5=12.1	I1=32.67	M2=5	S2=400

9.1.4 Block format

Functionality

A block should contain all data required to execute a machining step.

Generally, a block consists of several **words** and is always completed with the **end-of-block character** " LF " (Linefeed). This character is automatically generated when pressing the linefeed or <Input> key when writing.

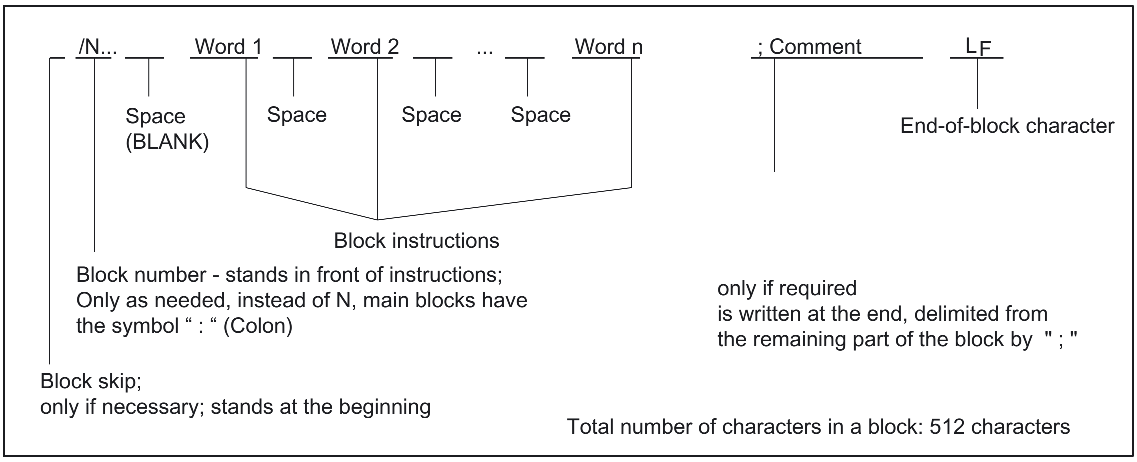

/N...
Word 1
Word 2
...
Word n
; Comment
LF
Space
(BLANK)
Space
Space
Space
End-of-block character
Block instructions
Block number - stands in front of instructions;
Only as needed, instead of N, main blocks have
the symbol " : " (Colon)
only if required
is written at the end, delimited from
the remaining part of the block by " ; "
Block skip;
only if necessary; stands at the beginning
Total number of characters in a block: 512 characters

Figure 9-2 Block structure diagram

Word order

If there are several instructions in a block, the following order is recommended:
N... G... X... Z... F... S... T... D... M... H...

Note regarding block numbers

First select the block numbers in steps of 5 or 10. Thus, you can later insert blocks and nevertheless observe the ascending order of block numbers.

Block skip

Blocks of a program, which are to be executed not with each program run, can be **marked** by a slash / in front of the block number.
The block skip itself is activated via **Operation** (program control: "SKP") or by the programmable controller (signal). A section can be skipped by several blocks in succession using " / ".
If a block must be skipped during program execution, all program blocks marked with " / " are not executed. All instructions contained in the blocks concerned will not be considered. The program is continued with the next block without marking.

Comment, remark

The instructions in the blocks of a program can be explained using comments (remarks). A comment always starts with a semicolon " ; " and ends with end-of-block.
Comments are displayed together with the contents of the remaining block in the current block display.

Messages

Messages are programmed in a separate block. A message is displayed in a special field and remains active until a block with a new message is executed or until the end of the program is reached. Max. 65 characters can be displayed in message texts.
A message without message text cancels a previous message.
MSG("THIS IS THE MESSAGE TEXT")

See also chapter "Service MSG".

Programming example

```
N10                                       ; G&S company, order no. 12A71
N20                                       ; Pump part 17, drawing no.: 123 677
N30                                       ; Program created by H. Adam, Dept. TV 4
N40 MSG("DRAWING NO.: 123677")
:50 G54 F4.7 S220 D2 M3                   ;Main block
N60 G0 G90 X100 Z200
N70 G1 Z185.6
N80 X112
/N90 X118 Z180                            ; Block can be suppressed
N100 X118 Z120
N110 G0 G90 X200
N120 M2                                   ; End of program
```

9.1.5 Character set

The following characters are used for programming; they are interpreted in accordance with the relevant definitions.

Letters, digits

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
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
No distinction is made between lowercase and uppercase letters.

Printable special characters

(	Open parenthesis	„	Inverted commas
)	Close parenthesis	_	Underscore (belongs to letters)
[	Open square bracket	.	Decimal point
]	Close square bracket	,	Comma, separator
<	less than	;	Comment start
>	greater than	%	Reserved; do not use
:	Main block, end of label	&	Reserved; do not use
=	Assignment, part of equation	'	Reserved; do not use
/	Division, block suppression	$	System variable identifiers
*	Multiplication	?	Reserved; do not use
+	Addition and positive sign	!	Reserved; do not use
-	Subtraction, minus sign		

Non-printable special characters

LF	End-of-block character
Blank	Separator between words; blank
Tab character	Reserved; do not use

9.1.6 Overview of instructions - Turning

Functions marked with ** are not available for SINUMERIK 802D sl value.

The functions marked with * take effect at the start of the program (CNC variant for the "turning" technology unless otherwise programmed and provided that the machine manufacturer default settings have not been changed).

Address	Significance	Value assignments	Information	Programming
D	Tool offset number	0 ... 9, only integer, no sign	Contains compensation data for a particular tool T... ; D0->compensation values= 0, max. 9 D numbers for one tool	D...
F	Feedrate	0.001 ... 99 999.999	Path velocity of a tool/workpiece; unit: mm/min or mm/revolution depending on G94 or G95	F...
F	Dwell time (block with G4)	0.001 ... 99 999.999	Dwell time in seconds	G4 F...; separate block
F	Thread lead change (block containing G34, G35)	0.001 ... 99 999.999	in mm/rev2	See G34, G35
G	G function (preparatory function)	Only integer, specified values	The G functions are divided into G groups. Only one G function of a group can be programmed in a block. A G function can be either modal (until it is canceled by another function of the same group) or only effective for the block in which it is programmed (non-modal).	G... or symbolic name, e.g.: CIP
			G group:	
G0	Linear interpolation at rapid traverse rate		1: Motion commands	G0 X... Z...
G1 *	Linear interpolation at feedrate		(type of interpolation)	G1 X...Z... F...
G2	Circular interpolation clockwise			G2 X... Z... I... K... F... ;Center and end point G2 X... Z... CR=... F... ;Radius and end point G2 AR=... I... K... F... ;opening angle and center point G2 AR=... X... Z... F... ;opening angle and end point
G3	Circular interpolation counter-clockwise			G3 ... ; otherwise as for G2
CIP	Circular interpolation through intermediate point			CIP X... Z... I1=... K1=... F... ;I1, K1 is intermediate point
CT	Circular interpolation; tangential transition			N10 ... N20 CT Z... X... F... ;circle, tangential transition to the previous path segment N10

Address	Significance	Value assignments	Information	Programming
G33	Thread cutting with constant lead		modally effective	;Constant lead G33 Z... K... SF=... ; cylindrical thread G33 X... I... SF=... ; face thread G33 Z... X... K... SF=... ; taper thread, in Z axis path larger than in the X axis G33 Z... X... I... SF=... ; taper thread, in X axis path larger than in the Z axis
G34	Thread cutting, increasing lead			G33 Z... K... SF=... ; cylindrical thread, constant lead G34 Z... K... F17.123 ; lead increasing with ; 17.123 mm/rev2
G35	Thread cutting, decreasing lead			G33 Z... K... SF=... ; cylindrical lead G35 Z... K... F7.321 ; lead decreasing with ; 7.321 mm/rev2
G331	Thread interpolation			N10 SPOS=... ; Spindle in position control N20 G331 Z... K... S... ; tapping without compensating chuck e.g. in Z axis ; RH or LH thread is defined via the sign of the lead (e.g. K+): + : as with M3 - : as with M4
G332	Thread interpolation - retraction			G332 Z... K... ;tapping without compensating chuck, e.g. in Z axis, retraction motion ; sign of lead as for G331
G4	Dwell time		2: Special motions, dwell time non-modal	G4 F...;separate block, F: Time in seconds or G4 S.... ;separate block, S: in spindle revolutions
G74	Reference point approach			G74 X1=0 Z1=0 ;separate block, (machine axis identifier!)
G75	Fixed point approach			G75 X1=0 Z1=0 ;separate block, (machine axis identifier!)
TRANS	translation, programmable		3: Write memory	TRANS X... Z... ;separate block

Address	Significance	Value assignments	Information	Programming
SCALE	Programmable scaling factor		non-modal	SCALE X... Z... ; scaling factor in the direction of the specified axis, separate block
ROT	rotation, programmable			ROT RPL=... ;rotation in the current plane G17 to G19, separate block
MIRROR	Programmable mirroring			MIRROR X0 ; coordinate axis whose direction is changed, separate block
ATRANS	additive translation, programming			ATRANS X... Z... ; separate block
ASCALE	Additive programmable scaling factor			ASCALE X... Z... ; scaling factor in the direction of the specified axis, separate block
AROT	additive programmable rotation			AROT RPL=... ; rotation in the current plane G17 to G19, separate block
AMIRROR	additive programmable mirroring			AMIRROR X0 ; coordinate axis whose direction is changed, separate block
G25	Lower spindle speed limitation or lower working area limitation			G25 S... ; separate block G25 X... Z... ; separate block
G26	Upper spindle speed limitation or upper working area limitation			G26 S... ; separate block G26 X... Z... ; separate block
G17	X/Y plane (when center-drilling, TRANSMIT milling required)		6: Plane selection	
G18 *	Z/X plane (standard turning)			
G19	Y/Z plane (required for TRACYL milling)			
G40 *	Tool radius compensation OFF		7: Tool radius compensation modally effective	
G41	Tool radius compensation left of contour			
G42	Tool radius compensation right of contour			
G500 *	Settable work offset OFF		8: Settable work offset modally effective	
G54	1. Settable work offset			
G55	2. settable work offset			
G56	3. settable work offset			
G57	4. settable work offset			
G58	5. settable work offset			
G59	6. settable work offset			

Address	Significance	Value assignments	Information	Programming
G53	Non-modal skipping of the settable work offset		9: Suppressing the settable work offset non-modal	
G153	Non-modal skipping of the settable work offset including base frame			
G60 *	Exact stop		10: Approach behavior	
G64	Continuous-path mode		modally effective	
G62	Corner deceleration at inside corners when tool radius offset is active (G41, G42)		Only in conjunction with continuous-path mode.	G62 Z... G1
G9	Non-modal exact stop		11: Non-modal exact stop non-modal	
G601 *	Exact stop window, fine, with G60, G9		12: Exact stop window	
G602	Exact stop window, coarse, with G60, G9		modally effective	
G621	Corner deceleration at all corners		Only in conjunction with continuous-path mode.	G621 AIDS=...
G70	Inch dimension input		13: Inch/metric dimension data	
G71 *	Metric dimension data input		modally effective	
G700	Inch dimension data input; also for feedrate F			
G710	Metric dimension data input; also for feedrate F			
G90 *	Absolute dimension data input		14: Absolute / incremental dimension	
G91	Incremental dimension input		modally effective	
G94	Feed F in mm/min		15: Feedrate / spindle	
G95 *	Feedrate F in mm/spindle revolutions		modally effective	
G96	Constant cutting rate ON (F in mm/rev., S in m/min)			G96 S... LIMS=... F...
G97	Constant cutting speed OFF			
G450 *	Transition circle		18: Behavior at corners when working with tool radius compensation	
G451	Point of intersection		modally effective	
BRISK *	Jerking path acceleration		21: Acceleration profile	
SOFT	Jerk-limited path acceleration		modally effective	
FFWOF *	Feedforward control OFF		24: Feedforward control	
FFWON	Feedforward control ON		modally effective	
WALIMON	Working area limitation ON		28: Working area limitation modally effective	; applies to all axes activated via setting data; values set via G25, G26
WALIMOF *	Working area limitation OFF			
DIAMOF	Radius dimensioning		29: Dimension Radius / diameter	
DIAMON *	Diameter dimensioning		modally effective	

Address	Significance	Value assignments	Information	Programming
G290 *	SIEMENS mode		47: External NC languages modally effective	
G291	External mode (not with 802D-bl)			
The functions marked with an asterisk (*) are active when the program is started (in the delivery condition of the control system, unless otherwise programmed and if the machine manufacturer has preserved the default setting for the "Turning" technology).				
H H0= to H9999=	H function	± 0.0000001 ... 9999 9999 (8 decimal places) or specified as an exponent: ± (10-300 ... 10+300)	Value transfer to the PLC; significance defined by the machine manufacturer	H0=... H9999=... e.g.: H7=23.456
I	Interpolation parameters	±0.001 ... 99 999.999 Thread: 0.001 ... 2000.000	Belongs to the X axis; meaning dependent on G2,G3 ->circle center or G33, G34, G35 G331, G332 -> thread lead	See G2, G3 and G33, G34, G35
K	Interpolation parameters	±0.001 ... 99 999.999 Thread: 0.001 ... 2000.000	Belongs to the Z axis; otherwise, as with I	See G2, G3 and G33, G34, G35
I1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the X axis; specification for circular interpolation with CIP	See CIP
K1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the Z axis; specification for circular interpolation with CIP	See CIP
L	Subroutine; name and call	7 decimals; integer only, no sign	Instead of a free name, it is also possible to select L1 ...L9999999; this also calls the subroutine (UP) in a separate block, Please note: L0001 is not always equal to L1. The name "LL6" is reserved for the tool change subroutine.	L.... ;separate block
M	Additional function	0 ... 99 only integer, no sign	For example, for initiating switching actions, such as "coolant ON", maximum five M functions per block.	M...
M0	Programmed stop		The machining is stopped at the end of a block containing M0; to continue, press NC START.	
M1	Optional stop		As with M0, but the stop is only performed if a special signal (Program control: "M01") is present.	
M2	End of main program with return to beginning of program		Can be found in the last block of the processing sequence	
M30	End of program (as M2)		Can be found in the last block of the processing sequence	
M17	End of subroutine		Can be found in the last block of the processing sequence	

Address	Significance	Value assignments	Information	Programming
M3	CW rotation of spindle (for master spindle)			
M4	CCW rotation of spindle (for master spindle)			
M5	Spindle stop (for master spindle)			
Mn=3**	CW rotation of spindle (for spindle n)		n = 1 or = 2	M2=3 ; CW rotation stop for spindle 2
Mn=4**	CCW rotation of spindle (for spindle n)		n = 1 or = 2	M2=4 ; CCW rotation stop for spindle 2
Mn=5**	Spindle stop (for spindle n)		n = 1 or = 2	M2=5 ; Spindle stop for spindle 2
M6	Tool change		Only if activated with M6 via the machine control panel; otherwise, change directly using the T command	
M40	Automatic gear stage switching (for master spindle)			
Mn=40	Automatic gear stage switching (for spindle n)		n = 1 or = 2	M1=40 ; automatic gear stage ; for spindle 1
M41 to M45	Gear stage 1 to gear stage 5 (for master spindle)			
Mn=41 to Mn=45	Gear stage 1 to gear stage 5 (for spindle n)		n = 1 or = 2	M2=41; 1st gear stage for spindle 2
M70, M19	-		Reserved; do not use	
M...	Remaining M functions		Functionality is not defined by the control system and can therefore be used freely by the machine manufacturer	
N	Block number - subblock	0 ... 9999 9999 only integer, no sign	Can be used to identify blocks with a number; is written at the beginning of a block.	N20
:	Block number of a main block	0 ... 9999 9999 only integer, no sign	Special block identification, used instead of N... ; such a block should contain all instructions for a complete subsequent machining step.	:20
P	Number of subroutine passes	1 ... 9999 only integer, no sign	Is used if the subroutine is run several times and is contained in the same block as the call	L781 P... ;separate block N10 L871 P3 ; three cycles

R0
to
R299
	Arithmetic parameters	± 0.0000001 ...
		9999 9999
		(8 decimal places)
		or specified as an exponent:
		± (10-300 ... 10+300)
				R1=7.9431 R2=4
				with specification of an exponent:
				R1=-1.9876EX9; R1=-1 987 600 000

Address	Significance	Value assignments	Information	Programming
Arithmetic functions			In addition to the 4 basic arithmetic functions using the operands + - * /, there are the following arithmetic functions:	
SIN( )	Sine	Degrees		R1=SIN(17.35)
COS()	Cosine	Degrees		R2=COS(R3)
TAN()	Tangent	Degrees		R4=TAN(R5)
ASIN()	Arc sine			R10=ASIN(0.35) ; R10: 20.487 degrees
ACOS()	Arc cosine			R20=ACOS(R2) ; R20: ... Degrees
ATAN2(,)	Arctangent2		The angle of the sum vector is calculated from 2 vectors standing vertically one on another. The 2nd vector specified is always used for angle reference. Result in the range: -180 to +180 degrees	R40=ATAN2(30.5,80.1) ; R40: 20.8455 degrees
SQRT()	Square root			R6=SQRT(R7)
POT()	Square			R12=POT(R13)
ABS()	Absolute value			R8=ABS(R9)
TRUNC()	Truncate to integer			R10=TRUNC(R2)
LN()	Natural logarithm			R12=LN(R9)
EXP()	Exponential function			R13=EXP(R1)
RET	Subroutine end		Used instead of M2 - to maintain the continuous-path mode	RET ;separate block
S...	Spindle speed (master spindle)	0.001 ... 99 999.999	Unit of measurement of the spindle r.p.m.	S...
S1=...	Spindle speed for spindle 1	0.001 ... 99 999.999	Unit of measurement of the spindle r.p.m.	S1=725 ; speed 725 r.p.m. for spindle 1
S2=...**	Spindle speed for spindle 2	0.001 ... 99 999.999	Unit of measurement of the spindle r.p.m.	S2=730 ; speed 730 r.p.m. for spindle 2
S	Cutting rate with G96 active	0.001 ... 99 999.999	Cutting rate unit m/min with G96; for master spindle only	G96 S...
S	Dwell time in block with G4	0.001 ... 99 999.999	Dwell time in spindle revolutions	G4 S... ;separate block
T	Tool number	1 ... 32 000 only integer, no sign	The tool change can be performed either directly using the T command or only with M6. This can be set in the machine data.	T...
X	Axis	±0.001 ... 99 999.999	Positional data	X...
Y	Axis	±0.001 ... 99 999.999	Positional data, e.g. with TRACYL, TRANSMIT	Y...
Z	Axis	±0.001 ... 99 999.999	Positional data	Z...

Address	Significance	Value assignments	Information	Programming
AC	Absolute coordinate	-	The dimension can be specified for the end or center point of a certain axis, irrespective of G91.	N10 G91 X10 Z=AC(20) ;X -incremental dimension, Z - absolute dimension
ACC[*axis*]	Percentage acceleration override	1 ... 200, integer	Acceleration override for an axis or spindle; specified as a percentage	N10 ACC[X]=80 ;for the X axis 80% N20 ACC[S]=50;for the spindle: 50%
ACP	Absolute coordinate; approach position in the positive direction (for rotary axis, spindle)	-	It is also possible to specify the dimensions for the end point of a rotary axis with ACP(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=ACP(45.3) ;approach absolute position of the A axis in the positive direction N20 SPOS=ACP(33.1) ;position spindle
ACN	Absolute coordinate; approach position in the negative direction (for rotary axis, spindle)	-	It is also possible to specify the dimensions for the end point of a rotary axis with ACN(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=ACN(45.3) ;approach absolute position of the A axis in the negative direction N20 SPOS=ACN(33.1) ;position spindle
ANG	Angle for the specification of a straight line for the contour definition	±0.00001 ... 359.99999	Specified in degrees; one possibility of specifying a straight line when using G0 or G1 if only one end-point coordinate of the plane is known or if the complete end point is known with contour ranging over several blocks	N10 G1 X... Z.... N11 X... ANG=... or contour over several blocks: N10 G1 X... Z... N11 ANG=... N12 X... Z... ANG=...
AR	Aperture angle for circular interpolation	0.00001 ... 359.99999	Specified in degrees; one possibility of defining the circle when using G2/G3	See G2, G3
CALL	Indirect cycle call	-	Special form of the cycle call; no parameter transfer; the name of the cycle is stored in a variable; only intended for cycle-internal use	N10 CALL VARNAME ; variable name
CHF	Chamfer; general use	0.001 ... 99 999.999	Inserts a chamfer of the specified chamfer length between two contour blocks	N10 X... Z.... CHF=... N11 X... Z...
CHR	Chamfer; in the contour definition	0.001 ... 99 999.999	Inserts a chamfer of the specified leg length between two contour blocks	N10 X... Z.... CHR=... N11 X... Z...
CR	Radius for circular interpolation	0.010 ... 99 999.999 Negative sign - for selecting the circle: greater than semicircle	One possibility of defining a circle when using G2/G3	See G2, G3

Address	Significance	Value assignments	Information	Programming
CYCLE...	Machining cycle	Only specified values	The call of the machining cycles requires a separate block; the appropriate transfer parameters must be loaded with values. Special cycle calls are also possible with an additional MCALL or CALL.	
CYCLE81	Drilling, centering			N5 RTP=110 RFP=100 .... ; Assign with values N10 CYCLE81(RTP, RFP, ...); separate part program block
CYCLE82	Drilling, counterboring			N5 RTP=110 RFP=100 .... ;assign with values N10 CYCLE82(RTP, RFP, ...) ;separate block
CYCLE83	Deep-hole drilling			N10 CYCLE83(110, 100, ...) ;or transfer values directly , separate block
CYCLE84	Rigid tapping			N10 CYCLE84(...) ;separate block
CYCLE840	Tapping with compensating chuck			N10 CYCLE840(...) ;separate block
CYCLE85	Reaming 1			N10 CYCLE85(...) ;separate block
CYCLE86	Boring			N10 CYCLE86(...) ;separate block
CYCLE87	Drilling with stop 1			N10 CYCLE87(...); separate part program block
CYCLE88	Drilling with stop 2			N10 CYCLE88(...) ;separate block
CYCLE89	Reaming 2			N10 CYCLE89(...); separate part program block
HOLES1	Row of holes			N10 HOLES1(...); separate part program block
HOLES2	Circle of holes			N10 HOLES2(...); separate part program block
CYCLE93	Recess			N10 CYCLE93(...) ;separate block
CYCLE94	Undercut DIN76 (forms E and F), finishing			N10 CYCLE94(...) ;separate block
CYCLE95	Stock removal with relief cutting			N10 CYCLE95(...) ;separate block
CYCLE96	Thread undercut			N10 CYCLE96(...); separate part program block
CYCLE97	Thread cutting			N10 CYCLE97(...) ;separate block

Address	Significance	Value assignments	Information	Programming
CYCLE98	Side-by-side thread mounting			N10 CYCLE98(...); separate part program block
DC	Absolute coordinate; approach position directly(for rotary axis, spindle)	-	It is also possible to specify the dimensions for the end point of a rotary axis with DC(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=DC(45.3) ;Approach absolute position of the A axis directly N20 SPOS=DC(33.1); Position spindle
DEF	Definition instruction		Defining a local user variable of the type BOOL, CHAR, INT, REAL, directly at the beginning of the program	DEF INT VARI1=24, VARI2 ; 2 variables of the type INT ; name defined by user
DITS	Run-in path with thread G33	-1 ... < 0, 0, > 0	Starting with configured axis acceleration; starting with sudden acceleration; run-in path specified, if necessary with axis overload	N10 G33 Z50 K5 DITS=4
DITE	Run-out path with thread G33	-1 ... < 0, 0, > 0	Braking with configured axis acceleration. Braking with sudden acceleration, specification of run-out path, with rounding	N10 G33 Z50 K5 DITE=4
FRC **	Non-modal feedrate for chamfer/rounding	0, >0	When FRC=0, feedrate F will act	For the unit, see F and G94, G95; for chamfer/rounding, see CHF, CHR, RND
FRCM **	Modal feedrate for chamfer/rounding	0, >0	When FRCM=0, feedrate F will act	For the unit, see F and G94, G95; for rounding/modal rounding, see RND, RNDM

FXS [*axis*]**	Travel to fixed stop	=1: select
		=0: Deselection
			Axis: Use the machine identifier	N20 G1 X10 Z25
				FXS[Z1]=1 FXST[Z1]=12.3
				FXSW[Z1]=2 F...

FXST [*axis*] **	Clamping torque, travel to fixed stop	> 0.0 ... 100.0	in %, max. 100% from the max. torque of the drive, *axis:* Use the machine identifier	N30 FXST[Z1]=12.3
FXSW [*axis*] **	Monitoring window, travel to fixed stop	> 0.0	Unit of measurement mm or degrees, axis-specific, *axis:* Use the machine identifier	N40 FXSW[Z1]=2.4
GOTOB	GoBack instruction	-	A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the program start.	N10 LABEL1: ... ... N100 GOTOB LABEL1
GOTOF	GoForward instruction	-	A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the end of the program.	N10 GOTOF LABEL2 ... N130 LABEL2: ...
IC	Coordinate specified using incremental dimensions	-	The dimension can be specified for the end or center point of a certain axis irrespective of G90.	N10 G90 X10 Z=IC(20) ;Z - incremental dimension, X - absolute dimension

Address	Significance	Value assignments	Information	Programming
IF	Jump condition	-	If the jump condition is fulfilled, the GoTo operation to the block with the following *label is performed;* , otherwise, the next instruction/block will follow.In one block, several IF instructions are possible. Relational operators: = = equal, <> not equal > greater than, < less than >= greater than or equal to <= less than or equal to	N10 IF R1>5 GOTOF LABEL3 ... N80 LABEL3: ...
LIMS	Upper limit speed of the spindle with G96, G97	0.001 ... 99 999.999	Limits the spindle speed with the G96 function enabled - constant cutting rate and G97	See G96
MEAS **	Measurement with deletion of distance-to-go	+1 -1	=+1: Measuring input 1, rising edge =-1: Measuring input1, falling edge	N10 MEAS=-1 G1 X... Z... F...
MEAW **	Measurement without deletion of distance-to-go	+1 -1	=+1: Measuring input 1, rising edge =-1: Measuring input1, falling edge	N10 MEAW=1 G1 X... Z... F...
$A_DBB[n] $A_DBW[n] $A_DBD[n] $A_DBR[n]	Data byte Data word Data double-word Real data		Reading and writing PLC variables	N10 $A_DBR[5]=16.3 ; Write Real variables ; with offset position 5 ; (position, type and meaning are agreed between NC and PLC)
$A_MONIFACT **	Factor for tool life monitoring	> 0.0	Initialization value: 1.0	N10 $A_MONIFACT=5.0 ; Tool life elapsed 5 times faster
$AA_FXS [*axis*] **	Status, travel to fixed stop	-	Values: 0 ... 5 *Axis*: Machine axis identifier	N10 IF $AA_FXS[X1]==1 GOTOF ....
$AA_MM[*axis*] **	Measurement result for an axis in the machine coordinate system	-	*Axis*: Identifier of an axis (X, Z) traversing when measuring	N10 R1=$AA_MM[X]
$AA_MW[*axis*] **	Measurement result for an axis in the workpiece coordinate system	-	*Axis*: Identifier of an axis (X, Z) traversing when measuring	N10 R2=$AA_MW[X]
$AC_MEA[1] **	Measuring task status	-	Default condition: 0: Default condition, probe did not switch 1: Probe switched	N10 IF $AC_MEAS[1]==1 GOTOF .... ; Continue program when probe has switched ...

Address	Significance	Value assignments	Information	Programming
$A..._..._ TIME **	Timer for runtime: $AN_SETUP_TIME $AN_POWERON_TIME $AC_OPERATING_TIME $AC_CYCLE_TIME $AC_CUTTING_TIME	0.0 ... 10+300 min (value read-only) min (value read-only) s s s	System variable: Time since the control system has last booted Time since the control system has last booted normally Total runtime of all NC programs Runtime of the NC program (only of the selected program) Tool action time	N10 IF $AC_CYCLE_TIME==50.5 ....
$AC_..._ PARTS **	Workpiece counter: $AC_TOTAL_PARTS $AC_REQUIRED_PARTS $AC_ACTUAL_PARTS $AC_SPECIAL_PARTS	0 ... 999 999 999, integer	System variable: Total actual count Set number of workpiece Current actual count Count of workpieces - specified by the user	N10 IF $AC_ACTUAL_PARTS==15 ....
$AC_ MSNUM	Number of the active master spindle		read-only	
$P_ MSNUM	Number of programmed master spindle		read-only	
$P_NUM_ SPINDLES	Number of configured spindles		read-only	
$AA_S[n]	Actual speed of spindle n		Spindle number n =1 or =2, read-only	
$P_S[n]	Last programmed speed of spindle n		Spindle number n =1 or =2, read-only	
$AC_ SDIR[n]	Current direction of rotation of spindle n		Spindle number n =1 or =2, read-only	
$P_ SDIR[n]	Last programmed direction of rotation of spindle n		Spindle number n =1 or =2, read-only	
$P_ TOOLNO	Number of the active tool T	-	read-only	N10 IF $P_TOOLNO==12 GOTOF ....
$P_TOOL	Active D number of the active tool	-	read-only	N10 IF $P_TOOL==1 GOTOF ....
$TC_MOP1 [t,d] **	Tool life warning limit	0.0 ...	in minutes, writing or reading values for tool t, D number d	N10 IF $TC_MOP1[13,1]<15.8 GOTOF ....
$TC_MOP2 [t,d] **	Residual tool life	0.0 ...	in minutes, writing or reading values for tool t, D number d	N10 IF $TC_MOP2[13,1]<15.8 GOTOF ....

Address	Significance	Value assignments	Information	Programming
$TC_MOP3 [t,d] **	Warning limit for count	0 ... 999 999 999, integer	Writing or reading values for tool t, D number d	N10 IF $TC_MOP3[13,1]<15 GOTOF ....
$TC_MOP4 [t,d] **	Residual unit quantity	0 ... 999 999 999, integer	Writing or reading values for tool t, D number d	N10 IF $TC_MOP4[13,1]<8 GOTOF ....
$TC_MOP11[t,d] **	Tool life setpoint	0.0 ...	in minutes, writing or reading values for tool t, D number d	N10 $TC_MOP11[13,1]=247.5
$TC_MOP13[t,d] **	Unit quantity setpoint	0 ... 999 999 999, integer	Writing or reading values for tool t, D number d	N10 $TC_MOP13[13,1]=715
$TC_TP8[t] **	Status of the tool	-	Default status - coding by bits for tool t, (bit 0 to bit 4)	N10 IF $TC_TP8[1]==1 GOTOF ....
$TC_TP9[t] **	Type of monitoring of the tool	0 ... 2	Monitoring type for tool t, writing or reading 0: No monitoring, 1: Tool life, 2: Quantity	N10 $TC_TP9[1]=2 ; Select count monitoring
MSG ()	Signal	max. 65 characters	Message text in inverted commas	MSG("MESSAGE TEXT") ; separate block ... N150 MSG() ; Clear previous message
OFFN	Slot width for TRACYL, otherwise, dimension specification	-	Only effective with the tool radius compensation G41, G42 active	N10 OFFN=12.4
RND	Rounding	0.010 ... 99 999.999	Inserts a rounding with the specified radius value tangentially between two contour blocks	N10 X... Z.... RND=... N11 X... Z...
RNDM	Modal rounding	0.010 ... 99 999.999 0	- Inserts roundings with the specified radius value tangentially at the following contour corners; special feedrate possible: FRCM= ... - Modal rounding OFF	N10 X... Y.... RNDM=.7.3 ;modal rounding ON N11 X... Y... .... N100 RNDM=.0 ;modal rounding OFF
RPL	Angle of rotation with ROT, AROT	±0.00001 ... 359.9999	Specification in degrees; angle for a programmable rotation in the current plane G17 to G19	see ROT, AROT

SET(, , ,)
REP()
	Set values for the variable fields		SET: Various values, from the specified element up to: according to the number of values
			REP: the same value, from the specified element up to the end of the field
				DEF REAL VAR2[12]=REP(4.5) ; all elements value 4.5
				N10 R10=SET(1.1,2.3,4.4) ; R10=1.1, R11=2.3, R4=4.4

SETMS(n) SETMS **	Define spindle as master spindle	n= 1 or n= 2	n: Number of the spindle, if only SETMS is set, the default master spindle comes into effect	N10 SETMS(2) ; separate block, 2nd spindle = master
SF	Thread starting point when using G33	0.001 ... 359.999	Specified in degrees; the thread starting point with G33 will be offset by the specified value	See G33

Address	Significance	Value assignments	Information	Programming
SPI(n)	converts the spindle number n into the axis identifier		n =1 or =2, axis identifier: e.g. "SP1" or "C"	
SPOS SPOS(n)	spindle position	0.0000 ... 359.9999	specified in degrees; the spindle stops at the specified position (to achieve this, the spindle must provide the appropriate technical prerequisites: position control) Spindle number n: 1 or 2	N10 SPOS=.... N10 SPOS=ACP(...) N10 SPOS=ACN(...) N10 SPOS=IC(...) N10 SPOS=DC(...)
SPOSA	Spindle position	0.0000 ... 359.9999	SPOS and SPOSA have the same functionality but differ in their block change behavior: With SPOS, the NC block is only enabled once the position has been reached. With SPOSA, the block is enabled even if the position has not been reached.	SPOSA=<value> / SPOSA [<n>] = <value>/
STOPFIFO	Stops the fast machining step		Special function; filling of the buffer memory until STARTFIFO, "Buffer memory full" or "End of program" is detected.	STOPFIFO; separate block, start of filling N10 X... N20 X...
STARTFIFO	Start of fast machining step		Special function; the buffer memory is filled at the same time.	N30 X... STARTFIFO ;separate block, end of filling
STOPRE	Preprocessing stop		Special function; the next block is only decoded if the block before STOPRE is completed.	STOPRE ; separate block
TRACYL(d) **	Milling of the peripheral surface	d: 1.000 ... 99 999.999	Kinematic transformation (only available if configured accordingly)	TRACYL(20.4) ; separate block ; cylinder diameter: 20.4 mm TRACYL(20.4,1) ; also possible
TRANSMIT **	Milling of the face end	-	Kinematic transformation (only available if configured accordingly)	TRANSMIT ; separate block TRANSMIT(1) ; also possible
TRAFOOF **	Disable TRANSMIT, TRACYL	-	Disables all kinematic transformations	TRAFOOF ; separate block

TRAILOF	Coupled motion in synchronism with the axis OFF
	(trailing OFF)
			modally effective	TRAILOF(<following axis>,<leading axis>,<leading axis 2>)
				TRAILOF(<following axis>)

TRAILON	Coupled motion in synchronism with the axis ON (trailing OFF)		modally effective	TRAILON(<following axis>,<leading axis>,<coupling factor>)

Address	Significance	Value assignments	Information	Programming
MASLDEF	Define master/slave axis grouping			`MASLDEF(Slv1,Slv2,..., master axis)`
MASLDEL	Decouple master/slave axis grouping and delete grouping definition			`MASLDEL(Slv1,Slv2,..., )`
MASLOF	Deactivation of a temporary coupling			`MASLOF(Slv1,Slv2,..., )`
MASLOFS	Deactivation of a temporary coupling with automatic slave axis stop			`MASLOFS(Slv1, Slv2, ..., )`
MASLON	Activation of a temporary coupling			`MASLON(Slv1,Slv2,..., )`

9.2 Positional data

9.2.1 Programming dimensions

In this section you will find descriptions of the commands, with which you can directly program dimensions taken from a drawing. This has the advantage that no extensive calculations have to be made for NC programming.

Note

The commands described in this section stand in most cases at the start 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 the following sections is to illustrate the conventional structure of an NC program.

Overview of typical dimensions

The basis of most NC programs is a drawing with concrete dimensions.

When implementing in a NC program, it is helpful to take over exactly the dimensions of a workpiece drawing into the machining program. These can be:

- Absolute dimension, G90 modally effective applies for all axes in the block, up to revocation by G91 in a following block.
- Absolute dimension, X=AC(value) only this value applies only for the stated axis and is not influenced by G90/G91. This is possible for all axes and also for SPOS, SPOSA spindle positionings, and interpolation parameters I, J, K.
- Absolute dimension, X=CC(value) directly approaching the position by the shortest route, only this value applies only for the stated rotary axis and is not influenced by G90/G91. Is also possible for SPOS, SPOSA spindle positionings.
- Absolute dimension, X=ACP(value) approaching the position in positive direction, only this value is set only for the rotary axis, the range of which is set to 0... < 360 degrees in the machine data.
- Absolute dimension, X=ACN(value) approaching the position in negative direction, only this value is set only for the rotary axis, the range of which is set to 0... < 360 degrees in the machine data.
- Incremental dimension, G91 modally effective applies for all axes in the block, until it is revoked by G90 in a following block.
- Incremental dimension, X=IC(value) only this value applies exclusively for the stated axis and is not influenced by G90/G91. This is possible for all axes and also for SPOS, SPOSA spindle positionings, and interpolation parameters I, J, K.
- Inch dimension, G70 applies for all linear axes in the block, until revoked by G71 in a following block.
- Metric dimension, G71 applies for all linear axes in the block, until revoked by G70 in a following block.
- Inch dimension as for G70, but applies also for feedrate and length-related setting data.

- Metric dimension as for G71, but applies also for feedrate and length-related setting data.
- Diameter programming, DIAMON on
- Diameter programming, DIAMOF off

Diameter programming, DIAM90 for traversing blocks with G90. Radius programming for traversing blocks with G91.

9.2.2 Absolute / incremental dimensioning: G90, G91, AC, IC

Functionality

With the instructions G90/G91, the written positional data X, Z, ... are evaluated as a coordinate point (G90) or as an axis position to traverse to (G91). G90/91 applies for all axes.
Irrespective of G90/G91, certain positional data can be specified for certain blocks in absolute/incremental dimensions using AC/IC.

These instructions do **not determine the path** by which the end points are reached; this is provided by a G group (G0, G1, G2 and G3... see Chapter "Axis Movements").

Programming

```
G90         ; Absolute dimension data
G91         ; Incremental dimension data

Z=AC(...)   ; Absolute dimensioning for a certain axis (here: Z axis), non-modal
Z=IC(...)   ; Absolute dimensioning for a certain axis (here: Z axis), non-modal
```

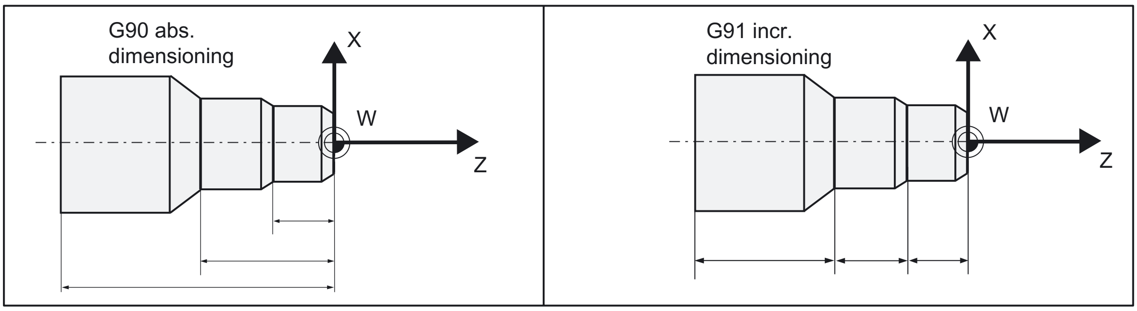


Figure 9-3 Different dimensioning types in the drawing

Absolute dimensioning G90

With absolute dimensioning, the dimensioning data refers to the **zero of the coordinate system currently active** (workpiece or current workpiece coordinate system or machine coordinate system). This is dependent on which offsets are currently active: programmable, settable, or no offsets.

Upon program start, G90 is active for **all axes** and remains active until it is deselected in a subsequent block by G91 (incremental dimensioning data) (modally active).

Incremental dimensioning G91

With incremental dimensioning, the numerical value of the path information corresponds to the **axis path to be traversed**. The leading sign indicates the **traversing direction**.

G91 applies to all axes and can be deselected in a subsequent block by G90 (absolute dimensioning).

Specification with =AC(...), =IC(...)

After the end point coordinate, write an equality sign. The value must be put in round brackets.
Absolute dimensioning is also possible for center points using =AC(...). Otherwise, the reference point for the circle center is the circle starting point.

Programming example

```
N10 G90 X20 Z90           ; Absolute dimensions
N20 X75 Z=IC(-32)         ; X-dimensions remain absolute, incremental Z dimension
...
N180 G91 X40 Z2           ; Switch-over to incremental dimensioning
N190 X-12 Z=AC(17)        ; X-remains incremental dimensioning, Z-absolute
```

9.2.3 Dimensions in metric units and inches: G71, G70, G710, G700

Functionality

If workpiece dimensions that deviate from the base system settings of the control are present (inch or mm), the dimensions can be entered directly in the program. The required conversion into the base system is performed by the control system.

Programming

G70	; Inch dimensions
G71	' Metric dimensions
G700	; Inch dimensions, also for feedrate F
G710	; Metric dimensions, also for feedrate F

Programming example

```
N10 G70 X10 Z30           ; Inch dimensions
N20 X40 Z50               ;G70 continues to act
...
N80 G71 X19 Z17.3         ; metric dimensioning from this point on
...
```

Information

Depending on the **default setting** you have chosen, the control system interprets all geometric values as either metric **or** inch dimensions. Tool offsets and settable work offsets including their displays are also to be understood as geometrical values; this also applies to the feed F in mm/min or inch/min.
The default setting can be set in machine data.
All examples provided in this Manual assume the **metric default setting.**

G70 or G71 evaluates all geometrical data that directly refer to the **workpiece**, either as inches or metric units, for example:

- Positional data X, Z, ... for G0,G1,G2,G3,G33, CIP, CT
- Interpolation parameters I, K (also thread lead)
- Circle radius CR
- **Programmable** work offset (TRANS, ATRANS)

All remaining geometric parameters that are not direct workpiece parameters, such as feedrates, tool offsets, and **settable** work offsets, are not affected by **G70/G71**.

G700/G710 however, also affects the feedrate F (inch/min, inch/rev. or mm/min, mm/rev.).

9.2.4 Radius / diameter dimensions: DIAMOF, DIAMON, DIAM90

Functionality

For machining parts, the positional data for the **X–axis** (transverse axis) is programmed as diameter dimensioning. When necessary, it is possible to switch to radius dimensioning in the program.

DIAMOF or DIAMON assesses the end point specification for the X axis as radius or diameter dimensioning. The actual value appears in the display accordingly for the workpiece coordinate system.

For DIAM90, irrespective of the traversing method (G90/G91), the actual value of the transverse axis is always displayed as a diameter. This also applies to reading of actual values in the workpiece coordinate system with MEAS, MEAW, $P_EP[x] and $AA_IW[x].

Programming

DIAMOF	; Radius dimensioning
DIAMON	; Diameter dimensioning
DIAM90	; diameter dimensioning for G90, radius dimensioning for G91

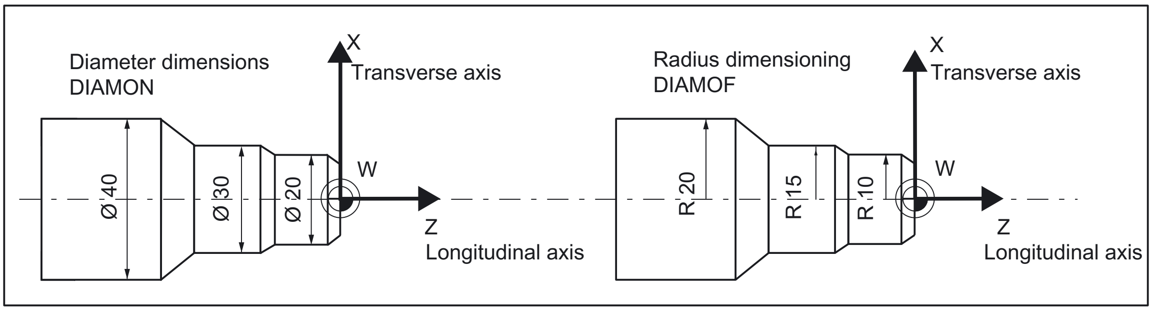


Figure 9-4 Diameter and radius dimensioning for the transverse axis

Programming example

```
N10 G0 X0 Z0                 ;Approach starting point
N20 DIAMOF                   Diameter input off
N30 G1 X30 S2000 M03 F0.8    ; X-axis = traverse axis 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 dimension and
                             ; radius programming for incremental dimension
N80 G91 X10 Z-20             Incremental dimension
N90 G90 X10                  Absolute dimensions
```

```
N100 M30                          ;End of program
```

Note

A programmable offset with TRANS X... or ATRANS X... is always evaluated as radius dimensioning. Description of this function: see the next section.

9.2.5 Programmable work offset: TRANS, ATRANS

Functionality

The programmable work offset can be used:

- for recurring shapes/arrangements in various positions on the workpiece
- when selecting a new reference point for the dimensioning
- as a stock allowance when roughing

This results in the **current workpiece coordinate system**. The rewritten dimensions use this as a reference.

The offset is possible in all axes.

Note

In the X-axis, the workpiece zero should be in the turning center due to the functions of diameter programming (DIAMON) and constant cutting speed (G96). For this reason, use no offset or only a small offset (e.g. as allowance) in the X axis.

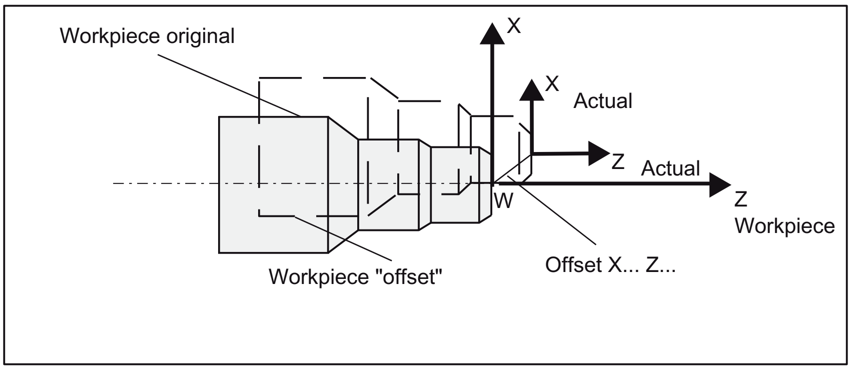


Figure 9-5 Effect of the programmable offset

Programming

TRANS Z...	; programmable offset, deletes old instructions for offsetting, rotation, scaling factor, mirroring
ATRANS Z...	; programmable offset, additive to existing instructions
TRANS	; without values: clears old instructions for offset, rotation, scaling factor, mirroring

The instructions that contain TRANS or ATRANS each require a separate block.

Programming example

```
N10 ...
N20 TRANS Z5              ; programmable offset, 5 mm in Z-axis
N30 L10                   ; Subroutine call; contains the geometry to be offset
...
N70 TRANS                 ; offset cleared
...
```

Subroutine call - see Section "Subroutine technique "

9.2.6 Programmable scaling factor: SCALE, ASCALE

Functionality

A scale factor can be programmed for all axes with SCALE, ASCALE. The path is enlarged or reduced by this factor in the axis specified.

The currently set coordinate system is used as the reference for the scale change.

Programming

SCALE X... Z...	; Programmable scaling factor, clears old instructions for offset, rotation, scaling factor, mirroring
ASCALE X... Z...	; Programmable scaling factor, additive to existing instructions
SCALE	; Without values: clears old instructions for offset, rotation, scaling factor, mirroring

The instructions that contain SCALE or ASCALE each require a separate block.

Notes

- For circles, the same factor should be used in both axes.
- If an ATRANS is programmed with SCALE/ASCALE active, these offset values are also scaled.

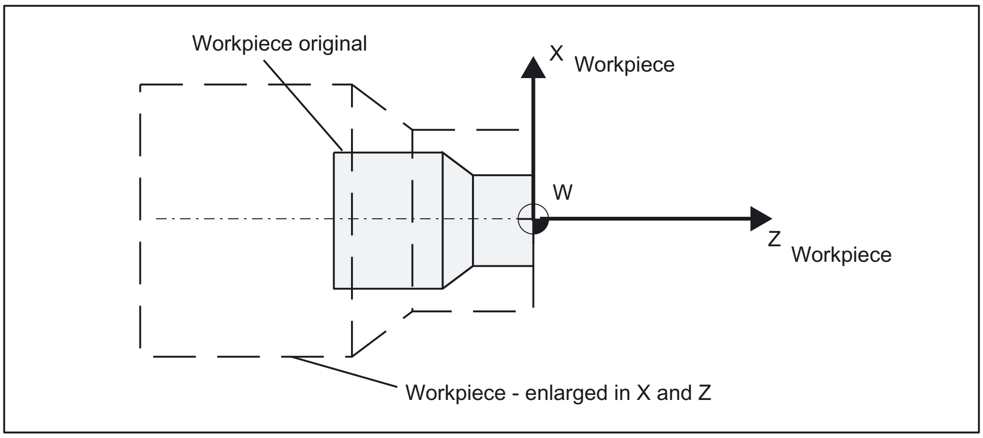


Figure 9-6 Example of a programmable scaling factor

Programming example

```
N20 L10                    ; Programmed contour original
N30 SCALE X2 Z2            ; contour in X and Z enlarged 2 times
N40 L10
...
```

Subroutine call - see section "Subroutine technique"

Information

In addition to the programmable offset and the scale factor, the following functions exist:

- Programmable rotation ROT, AROT and
- programmable mirroring, MIRROR, AMIRROR.

These functions are primarily used in milling. On turning machines, this is possible with TRANSMIT.

Examples of rotation and mirroring: see Section "List of instructions"

9.2.7 Workpiece clamping - settable work offset: G54 to G59, G500, G53, G153

Functionality

The adjustable work offset specifies the position of the workpiece zero on the machine (offset of the workpiece zero with respect to the machine zero point). This offset is determined upon clamping of the workpiece into the machine and must be entered in the corresponding data field by the operator. The value is activated by the program by selection from six possible groupings: G54 to G59.

For information on operation, see chapter "Setting/changing the work offset"

Programming

G54 to G59	; 1. to 6th settable work offset
G507 to G554	; 7. to 54th settable work offset
G500	; Settable work offset OFF - modal
G53	; settable work offset OFF non-modal, also suppresses programmable offset
G153	; As with G53; additionally suppresses base frame

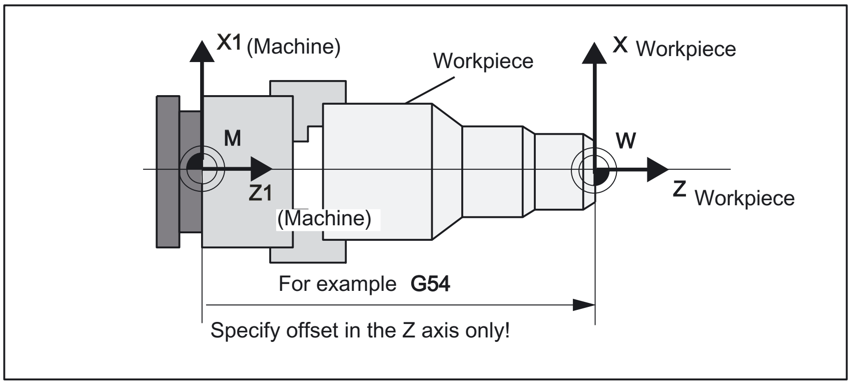


Figure 9-7 Settable work offset

Programming example

```
N10 G54 ...              ; 1st call settable work offset
N20 X... Z...            ; Machine the workpiece
...
N90 G500 G0 X...         ; Deactivate settable work offset
```

9.2.8 Programmable working area limitation: G25, G26, WALIMON, WALIMOF

Functionality

With G25, G26, a working area can be defined for all axes in which it is possible to traverse, however, not outside this area. With the tool length compensation active, the tool tip is decisive; The coordinate parameters are machine-based.

In order to be able to use the working area limitation, it must be activated for the respective axis. This is done via the input screen under "Offset Param" > "Setting data" > "Working area limit."

There are two options for defining the working area:

- Entering values via the input screen of the control system under the operating area <OFFSET PARAM> > "Setting data" > "Working area limit".

  This makes the working area limitation effective in JOG mode as well.

- Programing with G25/G26

  The values for the individual axes can be changed in the part program. The values that were entered in the input screen operating area <OFFSET PARAM> > "Setting data" > "Working area limit" are overwritten.

The working area limitation is enabled/disabled in the program by WALIMON/WALIMOF.

Programming

```
G25 X... Z...          ; Lower working area limitation
G26 X... Z...          ; Upper working area limitation

WALIMON                ; Working area limitation ON
WALIMOF                ; Working area limitation OFF
```

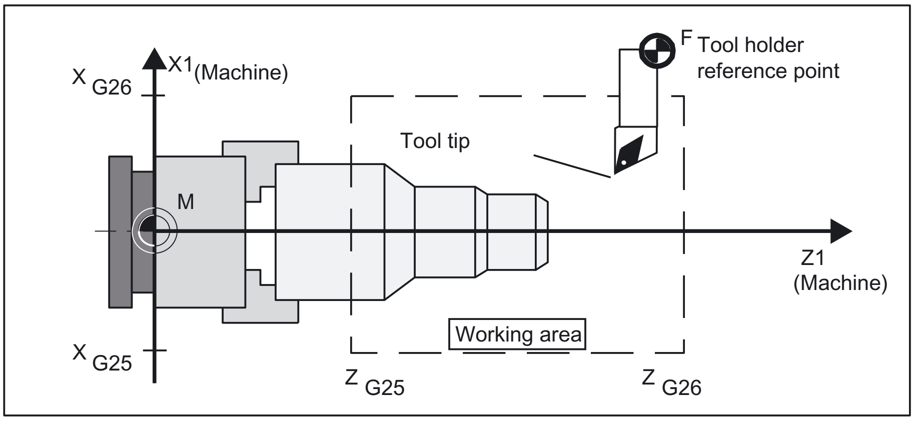


Figure 9-8 Programmable working area limitation

Notes

- The channel axis identifier from MD 20080 $MC_AXCONF_CHANAX_NAME_TAB should be used for G25, G26.
  With SINUMERIK 802D sl, kinematic transformations (TRAANG) are possible. In some cases, different axis identifiers are configured for MD 20080 and for the geometry axis identifiers MD20060 $MC_AXCONF_GEOAX_NAME_TAB.
- G25, G26 is also used in connection with the address S for the spindle speed limitation.
- A working area limitation can only be activated if the reference point for the relevant axes has been approached.

Programming example

```
N10 G25 X0 Z40              ; Values of the lower working area limitation
N20 G26 X80 Z160            ; Values of the upper working area limitation
N30 T1
N40 G0 X70 Z150
N50 WALIMON                 ; Working area limitation ON
...                         ; only within the working area
N90 WALIMOF                 ; Working area limitation OFF
```

9.3 Axis movements

9.3.1 Linear interpolation with rapid traverse: G0

Functionality

The rapid traverse movement G0 is used for fast positioning of the tool, however, **not for direct workpiece machining.**
All axes can be traversed simultaneously - on a straight path.

For each axis, the maximum speed (rapid traverse) is defined in machine data. If only one axis traverses, it uses its rapid traverse. If two axes are traversed simultaneously, the path velocity (resulting velocity) is selected to achieve the **maximum possible path velocity** in consideration of both axes.

Any programmed feedrates (F word) are not relevant for G0.
G0 remains active until canceled by another instruction from this G group (G0, G1, G2, G3, ...).

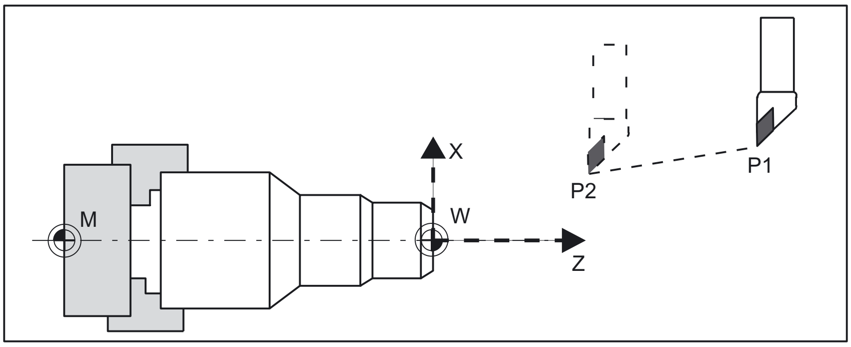


Figure 9-9 Linear interpolation with rapid traverse from point P1 to P2

Programming example

```
N10 G0 X100 Z65
```

Note

Another option for linear programming is available with the angle specification ANG=. (see Chapter "Contour definition programming")

Information

Another group of G functions exists for moving into the position (see Section "Exact stop/continuous-path mode: G60, G64"). For G60 exact stop, a window with various

precision values can be selected with another G group. For exact stop, an alternative instruction with non-modal effectiveness exists: G9.
You should consider these options for adaptation to your positioning tasks.

9.3.2 Linear interpolation with feedrate: G1

Functionality

The tool moves from the starting point to the end point along a straight path. For the **path velocity,** is determined by the programmed **F word** .
All the axes can be traversed simultaneously.
G1 remains active until canceled by another instruction from this G group (G0, G2, G3, ...).

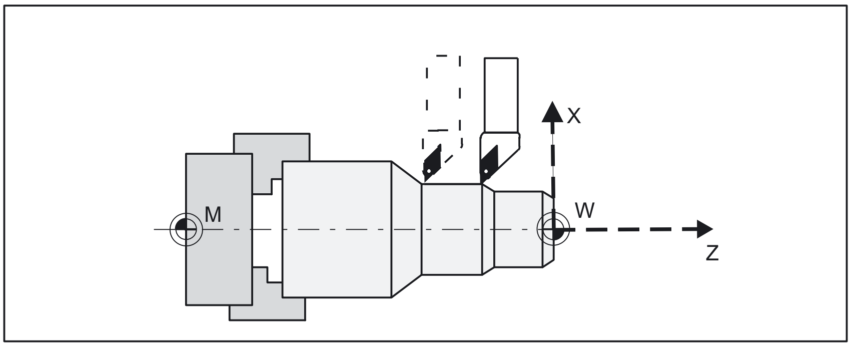


Figure 9-10 Linear interpolation with G1

Programming example

```
N05 G54 G0 G90 X40 Z200 S500 M3        ; The tool traverses in rapid traverse,
                                       spindle speed = 500 r.p.m., clockwise
N10 G1 Z120 F0.15                      ; Linear interpolation with feedrate 0.15
                                       mm/revolution
N15 X45 Z105
N20 Z80
N25 G0 X100                            ; Retraction in rapid traverse
N30 M2                                 ; End of program
```

Note: Another option for linear programming is available with the angle specification ANG=.

9.3.3 Circular interpolation: G2, G3

Functionality

The tool moves from the starting point to the end point along a circular path. The direction is determined by the G function:

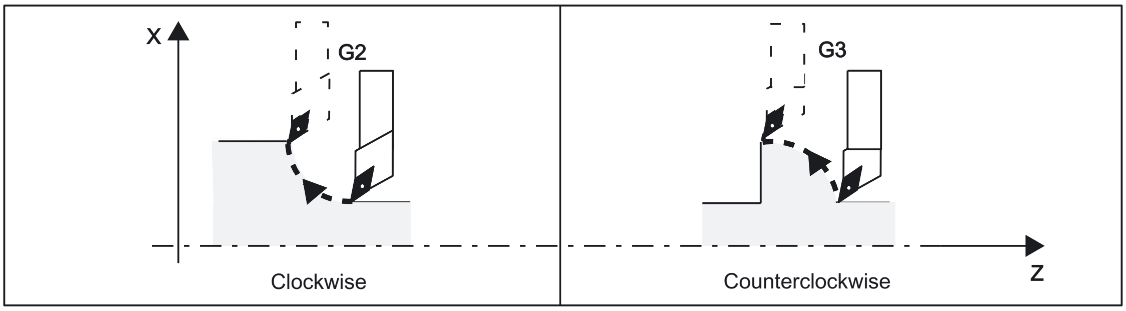


Figure 9-11 Definition of the circular direction of rotation G2-G3

The description of the desired circle can be given in various ways:

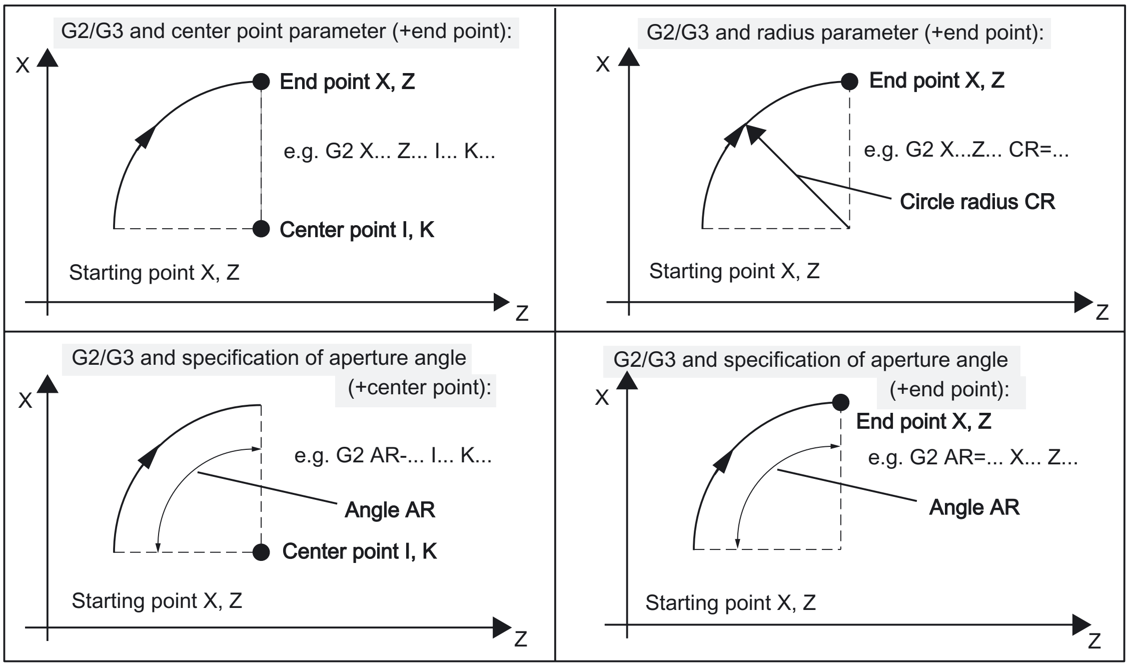


Figure 9-12 Options for circular path programming with G2/G3, with G2 as an example

G2/G3 remains active until canceled by another instruction from this G group (G0, G1, ...). The **path velocity** is determined by the programmed **F word.**

Programming

G2/G3 X... Y... I... J...	; Center and end points
G2/G3 CR=... X... Y...	; Circle radius and end point
G2/G3 AR=... I... J...	; Opening angle and center point
G2/G3 AR=... X... Y...	; Opening angle and end point
G2/G3 AP=... RP=...	; Polar coordinates, circle around the pole

Note

Further possibilities for circle programming result from:

CT - circle with tangential connection and

CIP - circle via intermediate point (see next sections).

Input tolerances for the circle

Circles are only accepted by the control system with a certain dimensional tolerance. The circle radius at the starting and end points are compared here. If the difference is within the tolerance, the center point is exactly set internally. Otherwise, an alarm message is issued.

The tolerance value can be set via machine data (see "Operating Instructions" 802Dsl).

Programming example: Definition of center point and end point

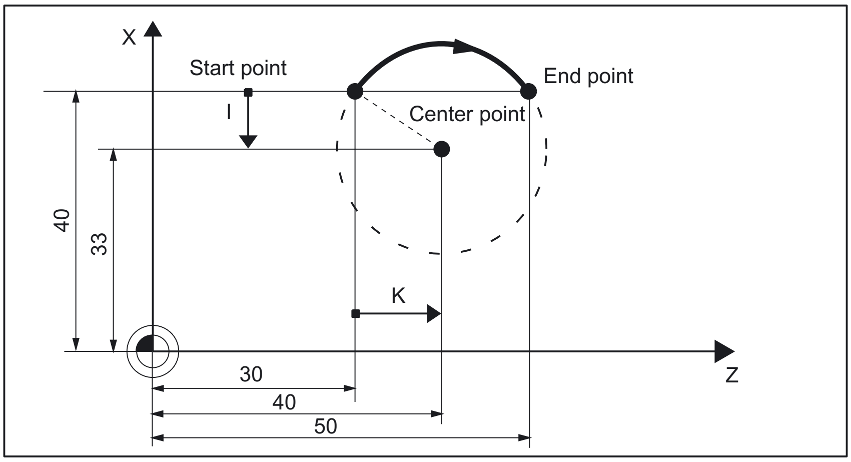


Figure 9-13 Example for center point and end point specification

```
N5 G90 Z30 X40              ; Starting point circle for N10
N10 G2 Z50 X40 K10 I-7      ; End point and center point
```

Note

Center point values refer to the circle starting point!

Programming example: End point and radius specification

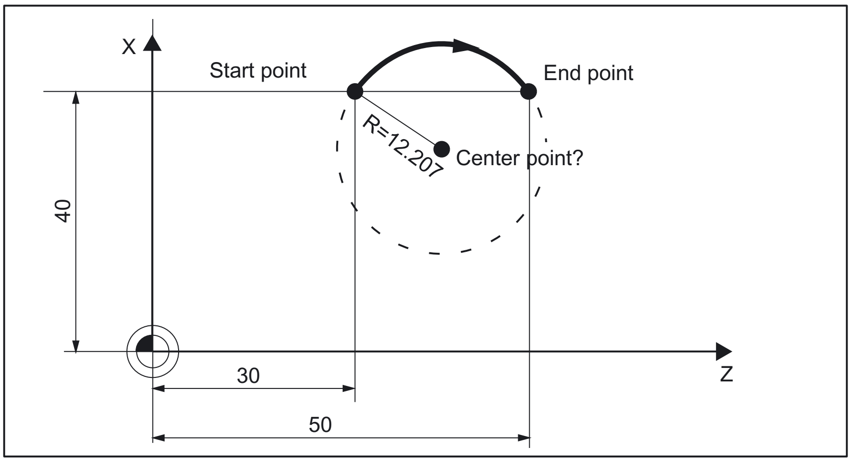


Figure 9-14 Example for end point and radius specification

```
N5 G90 Z30 X40              ; Starting point circle for N10
N10 G2 Z50 X40 CR=12.207    ; End point and radius
```

Note

With a negative leading sign for the value with CR=-..., a circular segment larger than a semicircle is selected.

Programming example: Definition of end point and aperture angle

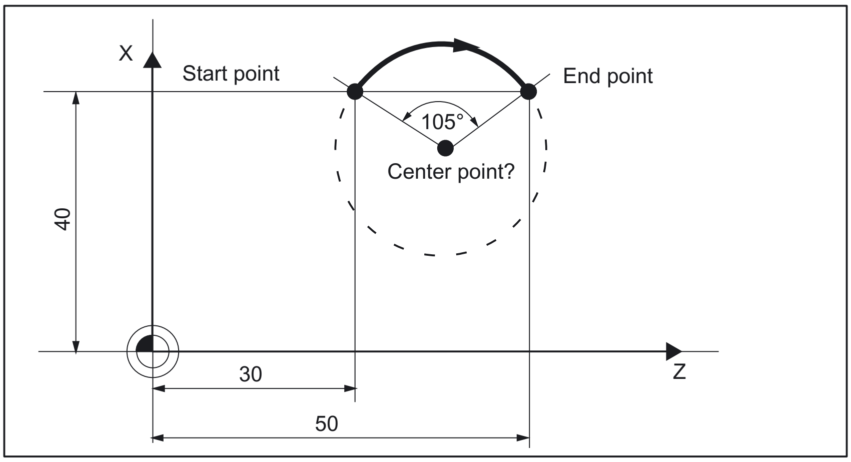


Figure 9-15 Example for end point and aperture angle specification

```
N5 G90 Z30 X40                  ; Starting point circle for N10
N10 G2 Z50 X40 AR=105           ; Opening angle and end point
```

Programming example: Definition of center point and aperture angle

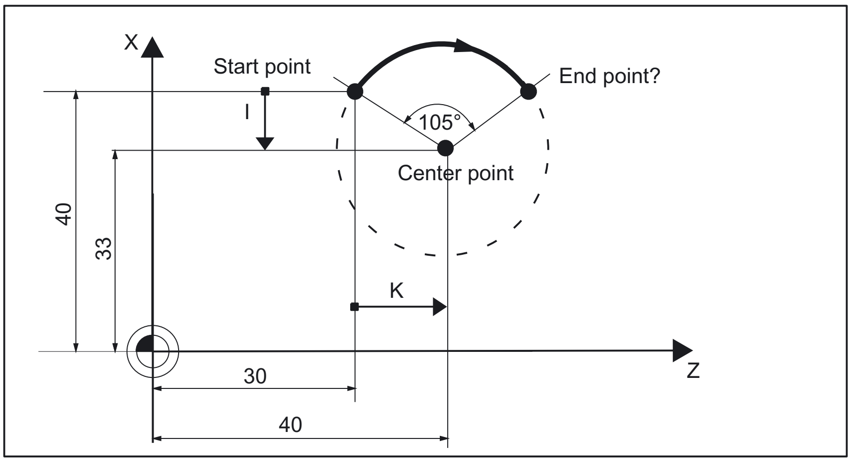


Figure 9-16 Example for center point and aperture angle specification

```
N5 G90 Z30 X40                  ; Starting point circle for N10
N10 G2 K10 I-7 AR=105           ; Opening angle and center point
```

Note

Center point values refer to the circle starting point!

9.3.4 Circular interpolation via intermediate point: CIP

Functionality

The direction of the circle results here from the position of the intermediate point (between starting and end points). Specification of intermediate point: I1=... for the X axis, K1=... for the Z axis.
CIP remains active until canceled by another instruction from this G group (G0, G1, ...).

The configured dimensional data G90 or G91 applies to the end point **and** the intermediate point.

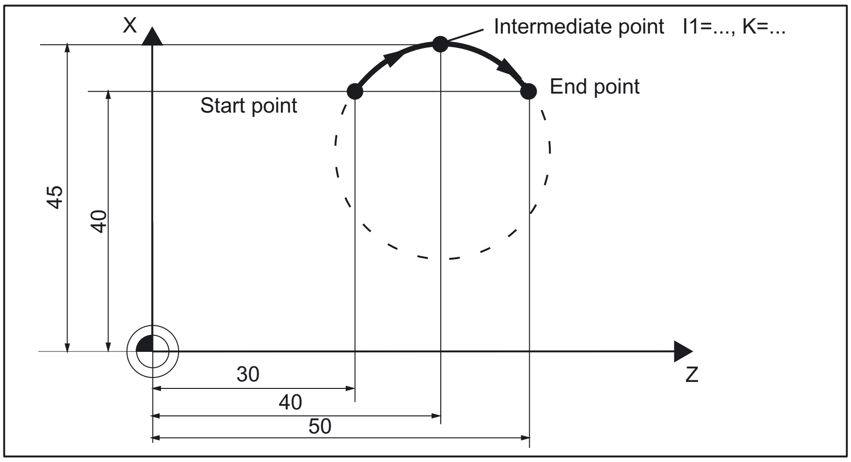


Figure 9-17 Circle with end point and intermediate point specification using the example of G90

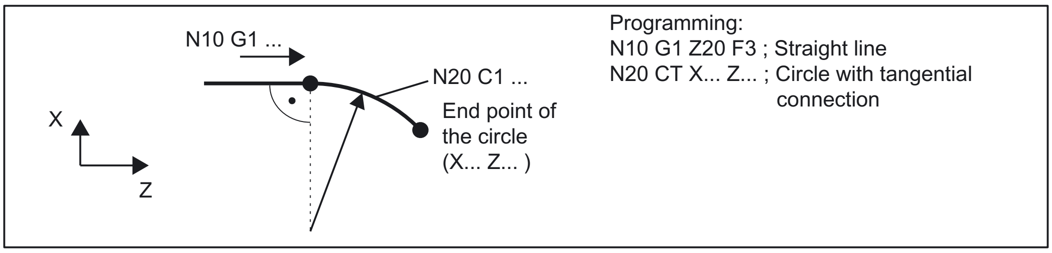
Programming example

```
N5 G90 Z30 X40                ; Starting point circle for N10
N10 CIP Z50 X40 K1=40 I1=45   ; End point and intermediate point
```

9.3.5 Circle with tangential transition: CT

Functionality

With CT and the programmed end point in the current plane (G18: Z/X plane), a circle is produced which tangentially connects to the previous path segment (circle or straight line). This defines the radius and center point of the circle from the geometric relationships of the previous path section and the programmed circle end point.



```
Programming:
N10 G1 Z20 F3 ; Straight line
N20 CT X... Z... ; Circle with tangential
                   connection
```

Figure 9-18 Circle with tangential transition to the previous path section

9.3.6 Thread cutting with constant lead: G33

Functionality

The function G33 can be used to machine threads with constant lead of the following type:

- Thread on cylindrical structures
- Thread on tapered structures
- External thread
- Single- and multiple-start thread
- Multi-block thread (series of threads)

This requires a spindle with position measuring system.

G33 remains active until canceled by another instruction from this G group (G0, G1, G2, G3, ...).

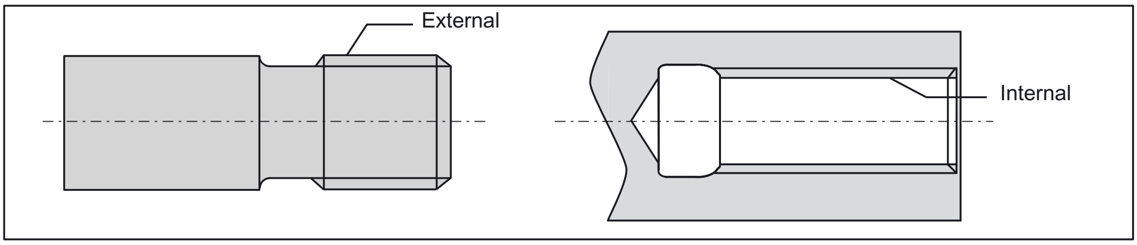


Figure 9-19 External / internal thread with cylindrical thread as an example

Right-hand or left-hand thread

Right-hand or left-hand thread is set with the rotation direction of the spindle (M3 right, M4 left). To do this, the rotation value must be programmed under address S or a rotation speed must be set.

Programming

Remark: Run-in and run-out paths must be taken into account for the thread lengths.

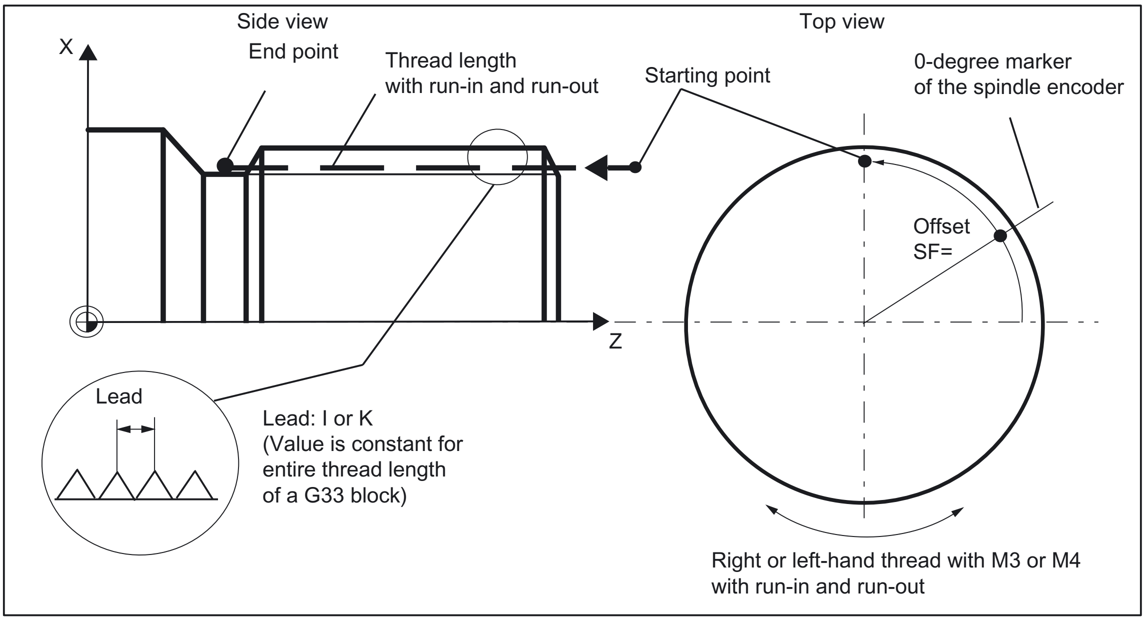


Figure 9-20 Programmable values for the thread with G33

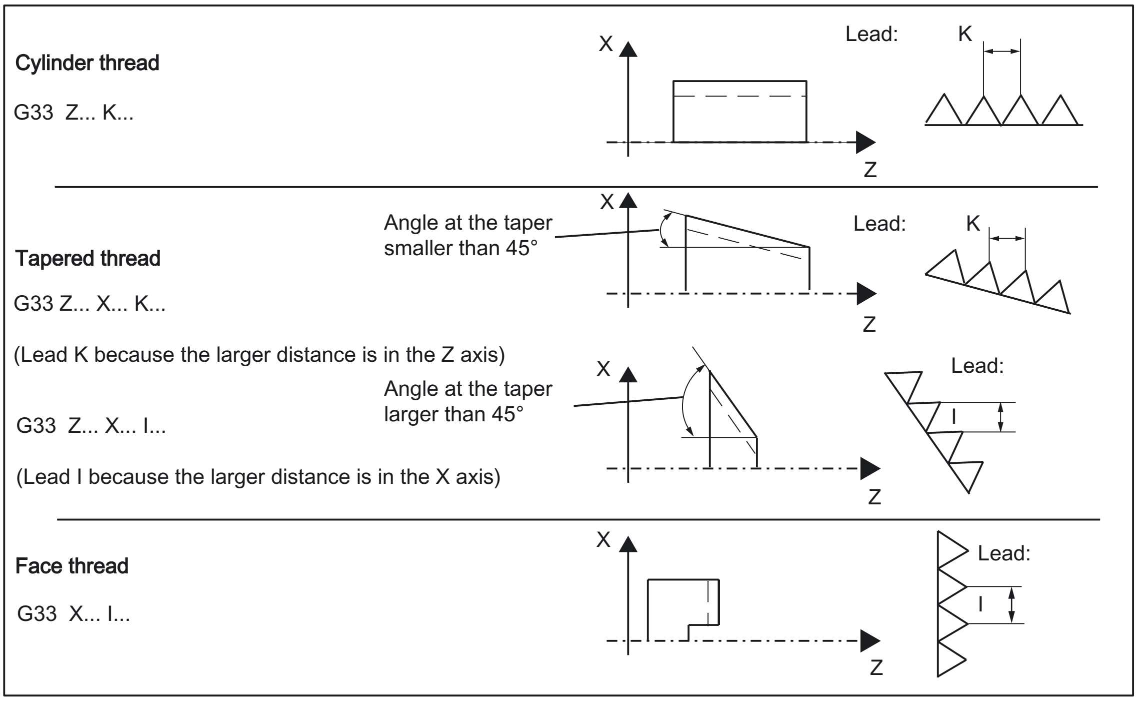

Cylinder thread
G33 Z... K...
X
Z
Lead:
K
Tapered thread
G33 Z... X... K...
(Lead K because the larger distance is in the Z axis)
Angle at the taper smaller than 45°
Angle at the taper larger than 45°
G33 Z... X... I...
(Lead I because the larger distance is in the X axis)
Lead:
I
Face thread
G33 X... I...
Lead:
I

Figure 9-21 Lead assignment for cylindrical, tapered and transversal thread

Tapered thread

For tapered threads (2 axis values required), the required lead address **I** or K of the axis with the **larger travel** (longer thread) must be used. A second lead is not defined.

Starting point offset SF=

A starting point offset is required for the spindle if a multiple-start thread or a thread in offset sections is to be machined. The starting point offset is programmed in the thread block with G33 under the address **SF** (absolute position).
If no starting point offset SF is written, the value from the setting data "Starting angle of thread" is active (SD 4200: THREAD_START_ANGLE) is active.
Please note: A programmed value for SF must always be entered in the setting data.

Programming example

Cylindrical thread, double-thread, starting point offset 180 degrees, thread length (including run-in and run-out) 100 mm, thread lead 4 mm/rev.

```
N10 G54 G0 G90 X50 Z0 S500 M3          ; Approach starting point, clockwise
                                       spindle rotation
N20 G33 Z-100 K4 SF=0                  ;Lead: 4 mm/rev
N30 G0 X54
N40 Z0
N50 X50
```

```
N60 G33 Z-100 K4 SF=180                    ; 2nd thread, offset by 180 degrees
N70 G0 X54 ...
```

Multi-block thread

If multiple thread blocks are programmed consecutively (multi-block thread), it only makes sense to define a starting point offset in the 1st thread block. The value is only used here.

Multi-block threads are connected automatically in G64 continuous path mode.

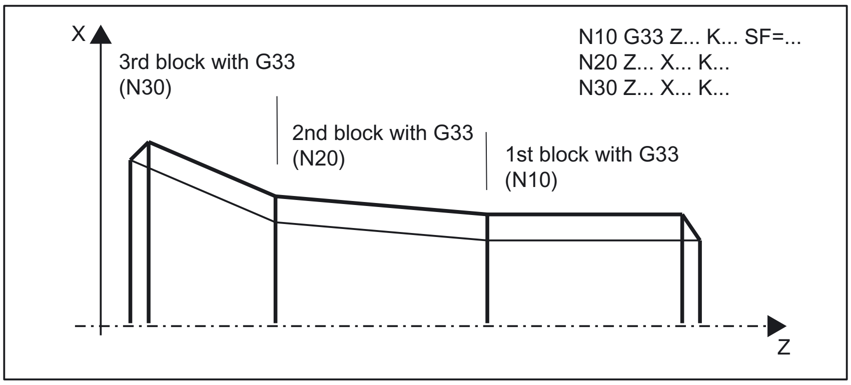


Figure 9-22 Example of multi-block thread (thread chaining)

Axis velocity

With G33 threads, the velocity of the axes for the thread length is determined on the basis of the spindle speed and the thread lead. The **feedrate F is not relevant.** It is, however, stored. However, the maximum axis velocity (rapid traverse) defined in the machine data can not be exceeded. This will result in an alarm.

Information

Important

- The spindle speed override switch should remain unchanged for thread machining.
- The feedrate override switch has no meaning in this block.

9.3.7 Programmable run-in and run-out path for G33: DITS, DITE

Functionality

The run-in and run-out path must also be traversed to the required thread with thread G33. The starting and braking of the axis (both axes in case of a tapered thread) are performed in these areas. This path depends on the thread lead, spindle speed, and the axis dynamics (configuration).
If the available path for run-in or run-out is limited, it may be necessary to reduce the spindle speed so that this path is sufficient.
In this case, the run-in and run-out paths can be specified separately in the program to achieve favorable cutting values and short machining times or to simplify the handling of this issue. If no values are specified, the values from the setting data (SD) apply. The specifications in the program are written into SD42010: THREAD_RAMP_DISP[0] ... [1].
If this path is not sufficient for the traversing with the configured axis acceleration, the axis is overloaded in terms of acceleration. Alarm 22280 ("Programmed run-in path too short") is then issued for the thread run-in. The alarm is purely for information and has no effect on part program execution.
The run-out path acts as a rounding clearance at the end of the thread. This achieves a smooth change in the axis movement when retracting.

Programming

```
DITS=...                    ; Run-in path of thread for G33
DITE=...                    ; Run-out path of thread for G33
```

Table 9- 3 Values for DITS and DITE or SD42010: THREAD_RAMP_DISP

-1 ... < 0:	Starting/braking of the feed axis is carried out with the configured acceleration. Jerk according to current BRISK/SOFT programming.
0:	Abrupt starting/braking of the feedrate axis on thread cutting.
> 0:	The run-in / run-out path of the thread is predefined for G33. To avoid alarm 22280, the acceleration limits of the axis must be observed in case of very small run-in and run-out paths.

Note: The value of SD42010 after reset / program start is -1.

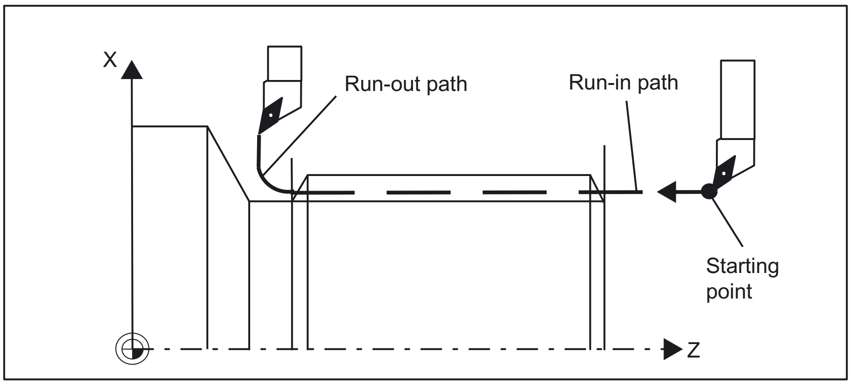


Figure 9-23 Run-in path and run-out path with corner rounding for thread G33

Programming example

```
...
N40 G90 G0 Z100 X10 M3 S500
N50 G33 Z50 K5 SF=180 DITS=4 DITE=2        ; run-in 4 mm, run-out 2 mm
N60 G0 X30
...
```

9.3.8 Thread cutting with variable lead: G34, G35

Functionality

Threads with variable lead can be produced in one block with G34 or G35:

- G34 ; Thread with (linearly) increasing lead
- G35 ; Thread with (linearly) decreasing lead.

Both functions otherwise have the same functionality as G33 and have the same prerequisites.

G34 or G35 remain active until canceled by another instruction from this G group (G0, G1, G2, G3, G33, ...).

Thread lead:

- I or K; Starting thread lead in mm/rev., associated with X or Z axis

Lead change:
In the block with G34 or G35, the address F contains the meaning of the lead change:
The lead (mm per revolution) changes per revolution.

- F ; lead change in mm/$_{\text{rev.}}$^{2}.

Note: Outside of G34, G35, the address F also indicates the feed or the dwell time for G4. The values programmed there remain saved.

Determining F

If you already know the starting and final lead of a thread, you can calculate the thread lead change F to be programmed according to the following equation:

$$F = \frac{\left| K_e{}^2 - K_a{}^2 \right|}{2 \times L_G} \left[mm / U^2 \right]$$

Explanation:
K_e Thread lead of the axis end point coordinate [mm/rev]
K_a Inititial thread lead (under I, K progr.) [mm/U]
$L_G$thread length in [mm]

Programming

```
G34 Z... K... F...          ;Cylindrical thread with increasing lead
G35 X... I... F...          ;Face thread with decreasing lead
G35 Z... X... K... F...     ;Taper thread with decreasing lead
```

Programming example

Table 9- 4 Cylindrical thread, subsequently with decreasing lead

```
N10 M3 S40                    ; Switch on spindle
N20 G0 G54 G90 G64 Z10 X60    ; Approach starting point
N30 G33 Z-100 K5 SF=15        ;Thread, constant lead 5mm/rev,
                              ; Activation point at 15 degrees
N40 G35 Z-150 K5 F0.16        ;Starting pitch 5 mm/rev,
                              ; Pitch decrease 0.16 mm/rev,
                              ; Thread length 50 mm,
                              ;Desired lead at end of block 3 mm/rev
N50 G0 X80                    ; Retraction in X
N60 Z120
N100 M2
```

9.3.9 Thread interpolation: G331, G332

Functionality

Use of this function for grinding machines is intended mainly for a second spindle (driven tool) - see section "2nd spindle".

A position-controlled spindle with position measuring system is required.
With G331/G332, threads **without** compensating chuck can be drilled, if the dynamic response of the spindle and the axis allow this.
If a compensating chuck is used nevertheless, the position differences to be compensated for by the compensating chuck are reduced. This allows thread grinding at higher spindle speeds.

G331 applies for grinding, G332 for grinding in opposite direction.
The grinding depth is specified through the axis, e.g. Z; the thread lead through the corresponding interpolation parameter (here: K).
For G332, the same lead is programmed as for G331. The spindle direction of rotation is automatically reversed.
The spindle speed is programmed with S; without M3/M4.
Before thread grinding with G331/G332, the spindle must be brought into the closed-loop position controlled mode using SPOS=... .

Right-hand or left-hand thread

The **sign of the thread lead** determines the direction of spindle rotation:
positive: right-hand (as with M3)
Negative: Left-hand (as with M4)

Remark:

A complete thread tapping cycle with thread interpolation is provided with the standard cycle CYCLE84.

Axis velocity

For G331/G332, the velocity of the axis for the thread length results from the spindle speed and the thread lead. The **feedrate F is not relevant**. It is, however, stored. However, the maximum axis velocity (rapid traverse) defined in the machine data can not be exceeded. This will result in an alarm.

Programming example

Metric thread 5,
lead according to the table: 0.8 mm/rev., hole already premachined:

```
N5 G54 G0 G90 X10 Z5          ; Approach starting point
N10 SPOS=0                    ;Spindle in position control
N20 G331 Z-25 K0.8 S600       ; Thread grinding, K positive = Clockwise rotation of
                              spindle, end point -25 mm

N40 G332 Z5 K0.8              ; Retraction
N50 G0 X... Z...
```

9.3.10 Fixed point approach: G75

Functionality

By using G75, a fixed point on the machine, e.g. tool change point, can be approached. The position is stored permanently in the machine data for all axes. A maximum of four fixed points can be defined for each axis.

No offset is effective. The velocity of each axis is its rapid traverse.
G75 requires a separate block and acts non-modal. The machine axis identifier must be programmed!
In the part program block after G75, the previous G command of the "Interpolation type" group (G0, G1,G2, ...) is active again.

Programming

```
G75 FP=<n> X1=0 Z1=0
```

Note

FPn is referencing with axis machine date MD30600 $MA_FIX_POINT_POS[n-1]. If no FP is programmed, then the first fixed point is selected.

Table 9- 5 Explanation

Command	Significance
G75	Fixed point approach
FP=<n>	Fixed point that is to be approached. The fixed point number is specified: <n> Value range of <n>: 1, 2, 3, 4 If no fixed point number is specified, fixed point 1 is approached automatically.
X1=0 Z1=0	Machine axes to be traversed to the fixed point. Specify the axes with value "0" with which the fixed point is to be approached simultaneously. Each axis is traversed with the maximum axial velocity.

Programming example

```
N05 G75 FP=1 X1=0                          ; Approach fixed point 1 in X
N10 G75 FP=2 Z1=0                          ; Approach fixed point 2 in Z, e. g. for
                                           tool change
N30 M30                                    ; End of program
```

Note

The programmed position values for X1, Z1 (any value, here = 0) are ignored, but must still be written.

9.3.11 Reference point approach: G74

Functionality

The reference point can be approached in the NC program with G74. Richtung und Geschwindigkeit jeder Achse sind in Maschinendaten hinterlegt.
G74 erfordert einen eigenen Satz und wirkt satzweise. The machine axis identifier must be programmed!
In the block after G74, the previous G command of the "Interpolation type" group (G0, G1,G2, ...) is active again.

Programming example

```
N10 G74 X1=0 Z1=0
```

Remark: The programmed position values for X1, Z1 (here = 0) are ignored, but must still be written.

9.3.12 Measuring with touch-trigger probe: MEAS, MEAW

Functionality

The function is available for SINUMERIK 802D sl plus and pro.

If the instruction MEAS=... or MEAW=... is in a block with traversing movements of axes, the positions of the traversed axes for the switching flank of a connected measuring probe are registered and stored. The measurement result can be read in the program for each axis. For MEAS, the movement of the axes is halted when the selected switching flank of the probe appears and the remaining distance to go is deleted.

Programming

MEAS=1	G1 X... Z... F...	; Measuring with rising edge of the probe, clearing the distance to go
MEAS=-1	G1 X... Z... F...	; Measuring with falling edge of the probe, clearing the distance to go
MEAW=1	G1 X... Z... F...	; Measuring with rising edge of the probe, without clearing the distance to go
MEAW=-1	G1 X... Z... F...	; Measuring with falling edge of the probe, without clearing the distance to go

CAUTION
For MEAW: Measuring probe travels to the programmed position even after is has triggered. Risk of destruction!

Measuring job status

If the probe has switched, the variable $AC_MEA[1] has the value=1 after the measuring block; otherwise the value =0.
When a measuring block is started, the variable is set to =0.

Measuring result

When the probe is successfully activated, the result of the measurement is available after the measuring block with the following variables for the axes traversed in the measuring block:
in the machine coordinate system: $AA_MM[*axis*]
in the workpiece coordinate system: $AA_MW[*axis*]
axis stands for X or Z.

Programming example

```
N10 MEAS=1 G1 X300 Z-40 F4000            ; Measurement with deletion of distance
                                         to go, rising edge
N20 IF $AC_MEA[1]==0 GOTOF MEASERR       ; measuring error?
N30 R5=$AA_MW[X] R6=$AA_MW[Z]            ; Processing of the measured values
..
N100 MEASERR: M0                         ; measuring error
```

Note: IF instruction - see Section "Conditional program jumps"

9.3.13 Feedrate F

Functionality

The feed F is the **path velocity** and represents the value of the geometric sum of the velocity components of all axes involved. The axis velocities are determined from the share of the axis path in the overall path.

The feedrate F is effective for the interpolation types G1, G2, G3, CIP, and CT and is retained until a new F word is written.

Programming

F...

Remark: For **integer values**, the decimal point is not required, e.g.: F300

Unit of measure for F with G94, G95

The dimension unit for the F word is determined by G functions:

- G94 **F** as the feedrate in**mm/min**
- G95 **F** as feedrate in **mm/rev.** of the spindle (only meaningful if the spindle is turning!)

Remark:
This unit of measure applies to metric dimensions. According to Section "Metric and inch dimensioning", settings with inch dimensioning are also possible.

Programming example

```
N10 G94 F310               ; Feedrate in mm/min
...
N110 S200 M3               ; Spindle rotation
N120 G95 F15.5             ; Feedrate in mm/revolution
```

Remark: Write a new F word if you change G94 - G95.

Information

The G group with G94, G95 also contains the functions G96, G97 for the constant cutting rate. These functions also influence the S word.

9.3.14 Exact stop / continuous-path control mode: G9, G60, G64

Functionality

G functions are provided for optimum adaptation to different requirements to set the traversing behavior at the block borders and for block advancing. For example, you would like to quickly position with the axes or you would like to machine path contours over multiple blocks.

Programming

```
G60                        ; Exact stop, modal
G64                        ; Continuous-path mode

G9                         ; Exact stop, non-modal

G601                       ; Exact stop window fine
G602                       ; Exact stop window coarse
```

Exact stop G60, G9

If the exact stop function (G60 or G9) is active, the velocity for reaching the exact end position at the end of a block is decelerated to zero.

Another modal G group can be used here to set when the traversing movement of this block is considered ended and the next block is started.

- G601 Exact stop window fine

  Block advance takes place when all axes have reached the "Exact stop window fine" (value in the machine data).

- G602 Exact stop window coarse

  Block advance takes place when all axes have reached the "Exact stop window coarse" (value in the machine data).

The selection of the exact stop window has a significant influence on the total time if many positioning operations are executed. Fine adjustments require more time.

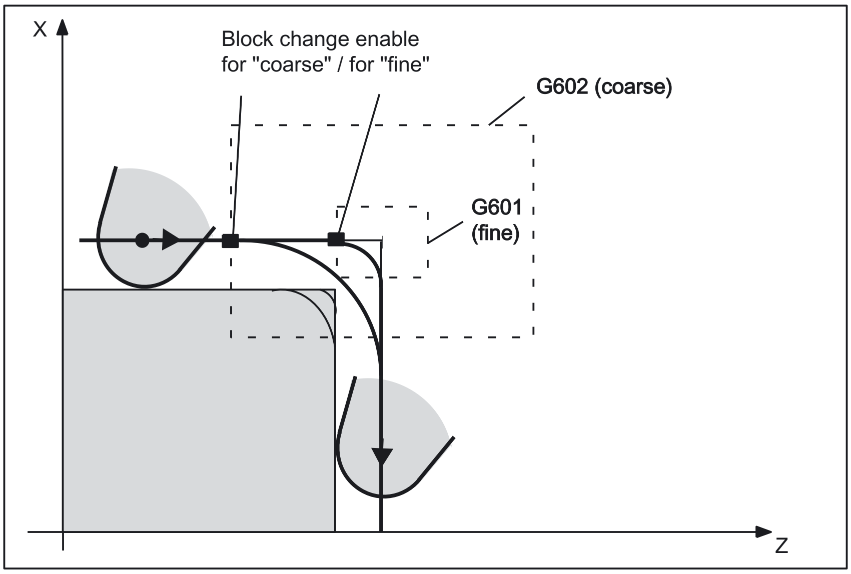


Figure 9-24 Exact stop window coarse or fine, in effect for G60-G9; enlarged display of the windows

Programming example

```
N5 G602                      ; Exact stop window coarse
N10 G0 G60 Z...              ; Exact stop modal
N20 X... Z...                ;G60 continues to act
...
N50 G1 G601 ...              ; Exact stop window fine
N80 G64 Z...                 ;Switching over to continuous-path mode
...
N100 G0 G9 Z...              ; Exact stop acts only in this block
N111 ...                     ;Again continuous-path mode
```

Remark: The G9 command only generates exact stop for the block in which it is programmed; G60, however, is effective until it is canceled by G64.

Continuous-path control mode G64

The objective of the continuous-path control mode is to avoid deceleration at the block boundaries and to switch **to the next block** with **a path velocity as constant as possible** (in the case of tangential transitions). The function works with **look-ahead velocity control** over several blocks.

For non-tangential transitions (corners), the velocity can be reduced rapidly enough so that the axes are subject to a relatively high velocity change over a short period of time. This may lead to a significant jerk (acceleration change). The size of the jerk can be limited by activating the SOFT function.

Programming example

```
N10 G64 G1 Z... F...        ; Continuous-path mode
N20 X...                    ; Continuous-path control mode continues to be active
...
N180 G60 ...                ; Switching over to exact stop
```

Look-ahead velocity control

In the continuous-path control mode with G64, the control system automatically determines the velocity control for several NC blocks in advance. This enables acceleration and deceleration across multiple blocks with approximately tangential transitions. For paths that consist of short travels in the NC blocks, higher velocities can be achieved than without look ahead.

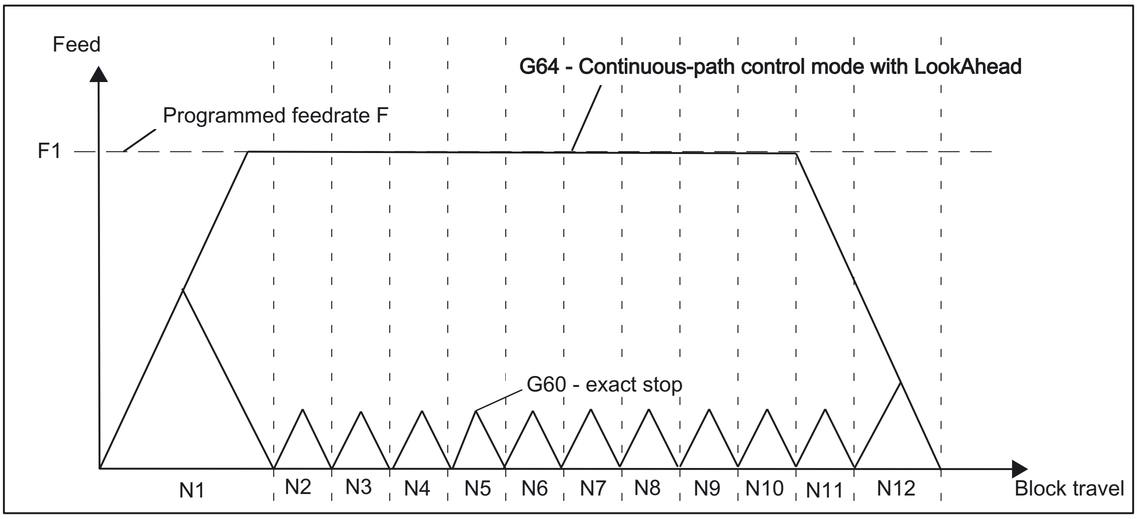


Figure 9-25 Comparison of the G60 and G64 velocity behavior with short travels in the blocks

9.3.15 Acceleration pattern: BRISK, SOFT

BRISK

The axes of the machine change their velocities using the maximum permissible acceleration value until reaching the final velocity. BRISK allows time-optimized working. The set velocity is reached in a short time. However, jumps are present in the acceleration pattern.

SOFT

The axes of the machine accelerate with nonlinear, constant curves until reaching the final velocity. With this jerk-free acceleration, SOFT allows for reduced machine load. The same behavior can also be applied to braking procedures.

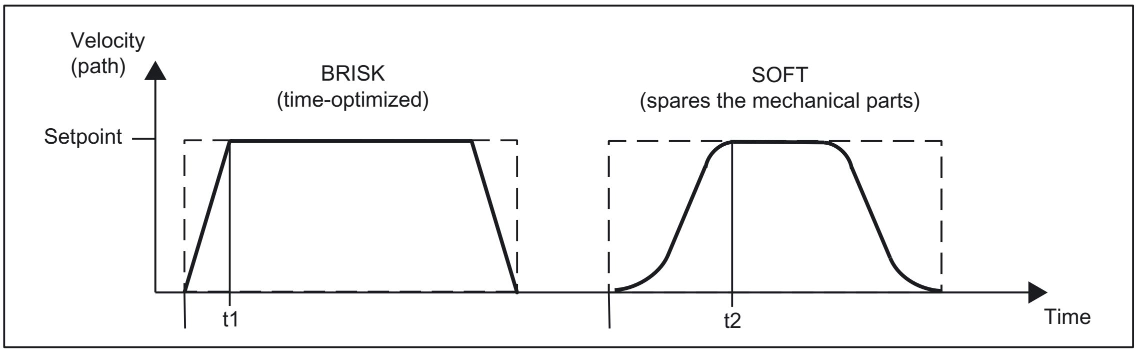


Figure 9-26 Principle course of the path velocity when using BRISK/SOFT

Programming

BRISK	; Jerking path acceleration
SOFT	; Jerk-limited path acceleration

Programming example

```
N10 SOFT G1 X30 Z84 F6.5     ; Jerk-limited path acceleration
...
N90 BRISK X87 Z104           ; continuing with jerking path acceleration
...
```

9.3.16 Percentage acceleration override: ACC

Functionality

Certain program sections can require the axis and spindle acceleration set via the machine data to be changed using the program. This programmable acceleration is a percentage acceleration override.

For each axis (e.g. X) or spindle (S), a percentage value >0% and ≤200% can be programmed. The axis interpolation is then carried out with this proportional acceleration.

The reference value (100%) is the valid machine data value for the acceleration of the axis or spindle. For the spindle, the reference value is also dependent upon:

- the gear stage
- the selected mode (positioning mode or speed mode).

Programming

ACC[axis name] = percentage	; for axis
ACC[S] = percentage	; for spindle

Programming example

```
N10 ACC[X]=80                   ; 80% acceleration for the x axis
N20 ACC[S]=50                   ; 50% acceleration for the spindle
...
N100 ACC[X]=100                 ; Deactivate the override for the X-axis
```

Activation

The limitation is active in all types of interpolation of the AUTOMATIC and MDA modes but **not** in JOG mode and for reference point approach.

The value assignment ACC[...] = 100 deactivates the override; likewise, as do RESET and End of program.

The programmed override value is also active with dry run feedrate.

⚠ CAUTION

A value greater than 100% may only be programmed if this load is permissible for the machine mechanics and the drives have the corresponding reserves. Failure to adhere to the limits can lead to damage to the mechanical parts and/or error messages.

9.3.17 Traversing with feedforward control: FFWON, FFWOF

Functionality

Through feedforward control, the following error in the traversing path is almost zero. Traversing with feedforward control permits greater path accuracy and thus better production results.

Programming

FFWON	; Feedforward control ON
FFWOF	; Feedforward control OFF

Programming example

```
N10 FFWON                      ; Feedforward control ON
N20 G1 X... Z... F9
...
N80 FFWOF                      ; Feedforward control OFF
```

9.3.18 3. and 4th axis.

Prerequisite

The control system must be designed for 3 or 4 axes.

Functionality

Depending on the machine design, a 3rd and even a 4th can be required. These axes can be implemented as linear or rotary axes. The identifier for these axes is defined by the machine manufacturer (e.g. B).

For rotary axes, the traversing range can be configured between 0 ...<360 degrees (modulo behavior) or -360 degrees/+360 degrees if there is no modulo axis is present.

With an appropriate machine design, a 3rd or 4th axis can be traversed linear simultaneously with the remaining axes. If the axis is traversed in a block with G1 or G2/G3 with the remaining axes (X, Z), it does not receive a component of the feedrate F. Its speed conforms to the path time of axes X, Z. It movement begins and ends with the remaining path axes. However, the speed cannot exceed the defined limit value.
If a block is programmed with this 3rd axis only, the axis will traverse using the active feedrate F when the G1 function is executed. If the axis is a rotary axis, the unit of measurement for F is degrees/min with G94 or degrees/rev. of the spindle with G95.

For these axes, offsets can be set (G54 ... G59) and programmed (TRANS, ATRANS).

Programming example

```
The 3rd axis is a swivel axis with the axis identifier B
N5 G94                      ; feedrate F in mm/min or degrees/min
N10 G0 X10 Z30 B45          ; X-Z traverse path with rapid traverse, B at the same
                            time
N20 G1 X12 Z33 B60 F400     ; X-Z traverse path at 400 mm/min, B at the same time
N30 G1 B90 F3000            ; Axis B traverses alone to position 90 degrees at a
                            speed of 3000 degrees/min
```

Special instructions for rotary axes: DC, ACP, ACN

```
For example, for rotary axis A:
A=DC(...)                   ; Absolute dimensions, approach position directly (on
                            the shortest path)
A=ACP(...)                  ; Absolute dimensions, approach position in positive
                            direction
A=ACN(...)                  ; Absolute dimensions, approach position in negative
                            direction
Example:
N10 A=ACP(55.7)             ; approach absolute position 55.7 degrees in positive
                            direction
```

9.3.19 Dwell Time: G4

Functionality

Between two NC blocks you can interrupt the machining process for a defined period by inserting your own block with G4; e.g. for relief cutting.
Words with F... or S... are only used for times in this block. Any previously programmed feedrate F or a spindle speed S remain valid.

Programming

```
G4 F...                        ; Dwell time in seconds
G4 S...                        ; Dwell time in spindle revolutions
```

Programming example

```
N5 G1 F3.8 Z-50 S300 M3      ;Feed F; spindle speed S
N10 G4 F2.5                  ; Dwell time 2.5 seconds
N20 Z70
N30 G4 S30                   ;dwelling 30 revolutions of the spindle, corresponds at
                             ; S=300 rpm and 100 % speed override to: t=0.1 min
N40 X...                     ;Feed and spindle speed remain effective
```

Remark

G4 S.. is only possible if a controlled spindle is available (if the speed specifications are also programmed via S...).

9.3.20 Travel to fixed stop

Functionality

This function is available for SINUMERIK 802D sl plus and 802D sl pro.

The travel to fixed stop (FXS = Fixed Stop) function can be used to establish defined forces for clamping workpieces, such as those required for spindle sleeves and grippers. The function can also be used for the approach of mechanical reference points. With sufficiently reduced torque, it is also possible to perform simple measurement operations without connecting a probe.

Programming

FXS[axis]=1	; Select travel to fixed stop
FXS[axis]=0	; Deselect travel to fixed stop
FXST[axis]=...	; Clamping torque, specified in % of the max. torque of the drive
FXSW [axis]=…	; Width of the window for fixed-stop monitoring in mm/degrees

Remark: The machine axis identifier should be used as the axis identifier (e.g.: X1). The channel axis identifier (e.g. X) is only permitted, if e.g. no coordinate rotation is active and this axis is directly assigned to a machine axis.

The commands are modal. The traversing path and the selection of the function FXS[axis]=1 must be programmed in one block.

Programming example - selection

```
N10 G1 G94 ...
N100 X250 Z100 F100 FXS[Z1]=1      ; selected for machine axis Z1 FXS function,
FXST[Z1]=12.3                      ; Clamping torque 12.3%,
FXSW[Z1]=2                         ; window width 2 mm
```

Notes

- When selected, the fixed stop must be located between the start and end positions.
- The parameters for torque FXST[ ]= and window width FXSW[ ]= are optional. If these are not written, the values from existing setting data (SD) are in effect. Programmed values are imported to the setting data. At the start, the setting data are loaded with values from machine data. FXST[ ]=... or FXSW[ ]=... can be changed in the program at any time. The changes are applied before traversing movements in the block.

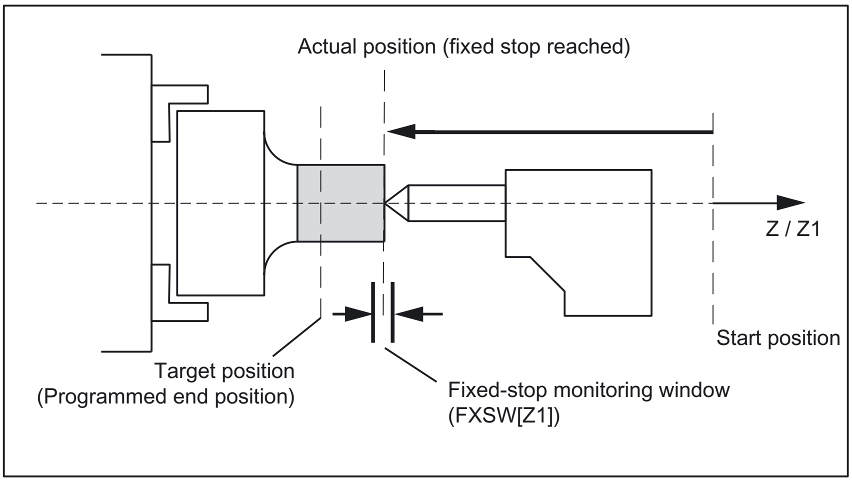

Actual position (fixed stop reached)
Z / Z1
Start position
Target position
(Programmed end position)
Fixed-stop monitoring window
(FXSW[Z1])

Figure 9-27 Example for travel to fixed stop, a quill is pressed onto the workpiece

Other programming examples

```
N10 G1 G94 ...
N20 X250 Z100 F100 FXS[X1]=1        ; selected for machine axis X1 FXS,
                                    ; clamping torque and window width from SDs
N20 X250 Z100 F100 FXS[X1]=1        ; selected for machine axis X1 FXS,
FXST[X1]=12.3                       ; clamping torque 12.3% and window width from SD
N20 X250 Z100 F100 FXS[X1]=1        ; selected for machine axis X1 FXS,
FXST[X1]=12.3 FXSW[X1]=2            ; clamping torque 12.3% and window width 2 mm
N20 X250 Z100 F100 FXS[X1]=1        ; selected for machine axis X1 FXS,
FXSW[X1]=2                          ; clamping torque from SD, window width 2 mm
```

Fixed stop reached

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 FXST[ ]=... or the value from SD and then remains constant.
- The monitoring of the fixed stop is active within the specified window width (FXSW[ ]=... or value from SD).

Deselecting the function

Deselection of the function triggers a preprocessing stop. The block with FXS[X1]=0 must contain traversing movements.

Example:

```
N200 G1 G94 X200 Y400 F200         Axis X1 is retracted from the fixed stop to
FXS[X1] = 0                        position
                                    X= 200 mm.
```

⚠ CAUTION
The traversing movement to the retraction position must lead away from the fixed stop; otherwise, damage to the fixed 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 once the torque limit has been deactivated.

Further notes

- "Measuring with deletion of distance-to-go" ("MEAS" command) and "Travel to fixed stop" cannot be programmed in the same block.
- Contour monitoring is not performed while "Travel to fixed stop" is active.
- 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. Ensure that the axis can still follow. For this reason, it must be verified that the contour deviation is not larger than that with unlimited torque.
- 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. when inserting a spindle sleeve).

System variable for status: $AA_FXS[*axis*]

This system variable provides the "Travel to fixed stop" status for the axis specified:

Value =	0:	Axis not at fixed stop
	1:	Fixed stop has been approached successfully (axis is within fixed stop monitoring window)
	2:	Approach to fixed stop has failed (axis is not at fixed stop)
	3:	Travel to fixed stop activated
	4:	Fixed stop detected
	5:	Travel to fixed stop is deselected. The deselection is not yet completed.

Query of the system variables in the parts program initiates a preprocessing stop.

For SINUMERIK 802D sl, only the static states can be detected before and after selection/deselection.

Alarm suppression

The issuing of the following alarms can be suppressed with machine data:

- 20091 "Fixed stop not reached"
- 20094 "Fixed stop aborted"

Reference

SINUMERIK 802D sl Function Manual for Turning, Milling, Nibbling; Travel to Fixed Stop

9.3.21 Feed reduction with corner deceleration (FENDNORM, G62, G621)

Function

With automatic corner deceleration the feed rate is reduced according to a bell curve before reaching the corner. It is also possible to parameterize the extent of the tool behavior relevant to machining via setting data. These are:

- Start and end of feed rate reduction
- Override with which the feed rate is reduced
- Detection of a relevant corner

Relevant corners are those whose inside angle is less than the corner parameterized in the setting data.

Default value `FENDNORM` deactivates the function of the automatic corner override.

Reference

ISO Dialects for SINUMERIK Description of Functions

Programming

```
FENDNORM
G62 G41
or
G621
```

Parameters

FENDNORM	; Automatic corner deceleration OFF
G62	; Corner deceleration at inside corners when tool radius offset is active
G621	; Corner deceleration at all inside corners when tool radius compensation is active

G62 only applies to inside corners with

- active tool radius offset `G41`, `G42` and
- active continuous-path mode `G64`, `G641`

The corner is approached at a reduced feed rate resulting from:

F * (override for feed rate reduction) * feed rate override

The maximum possible feed rate reduction is attained at the precise point where the tool is to change directions at the corner, with reference to the center path.

G621 applies analogously with G62 at each corner of the axes defined by `FGROUP`.

9.3.22 Coupled axes

9.3.22.1 Coupled motion (TRAILON, TRAILOF)

Functionality

When a defined leading axis is moved, the coupled motion axes (= following axes) assigned to it traverse through the distances described by the leading axis, allowing for a coupling factor.

Together, the leading axis and following axis represent coupled axes.

Application examples

- Traversal of an axis by means of a simulated axis. The leading axis is a simulated axis and the coupled axis a real axis. In this way, the real axis can be traversed as a function of the coupling factor.
- Two-sided machining with 2 coupled motion groups:

  1. Leading axis Y, coupled motion axis V

  2. Leading axis Z, coupled motion axis W

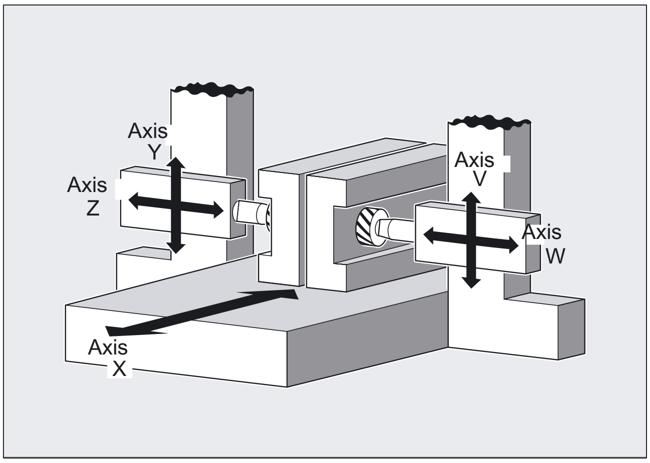


Figure 9-28 Coupled motion, example two-side machining

Programming

```
TRAILON(<following axis>,<leading axis>,<coupling factor>)
TRAILOF(<following axis>,<leading axis>,<leading axis 2>)
TRAILOF(<following axis>)
```

Significance

`TRAILON`

Command for activating and defining a coupled axis grouping

Effectiveness: modal

`<following axis>`

Parameter 1: Axis name of trailing axis

Note:
A coupled-motion axis can also act as the leading axis for other coupled-motion axes. In this way, it is possible to create a range of different coupled axis groupings.

`<leading axis>`

Parameter 2: Axis name of trailing axis

`<coupling factor>`

Parameter 3: Coupling factor

The coupling factor specifies the desired relationship between the paths of the coupled-motion axis and the leading axis:

<coupling factor> = path of coupled-motion axis/path of leading axis

Type: REAL

Default: 1

The input of a negative value causes the master and coupled axes to traverse in opposition.

If a coupling factor is not programmed, then coupling factor 1 automatically applies.

`TRAILOF`

Command for deactivating a coupled axis grouping

Effectiveness: modal

`TRAILOF` with 2 parameters deactivates only the coupling to the specified leading axis:

```
TRAILOF(<following axis>,<leading axis>)
```

If a coupled-motion axis has 2 leading axes, `TRAILOF` can be called with 3 parameters to deactivate both couplings.

```
TRAILOF(<following axis>,<leading axis>,<leading axis 2>)
```

Programming `TRAILOF` without specifying a leading axis produces the same result:

```
TRAILOF(<following axis>)
```

Note

Coupled axis motion is always executed in the base coordinate system (BCS).

The number of coupled axis groupings which may be simultaneously activated is limited only by the maximum possible number of combinations of axes on the machine.

Programming example

The workpiece is to be machined on two sides with the axis configuration shown in the diagram. To do this, you create two combinations of coupled axes.

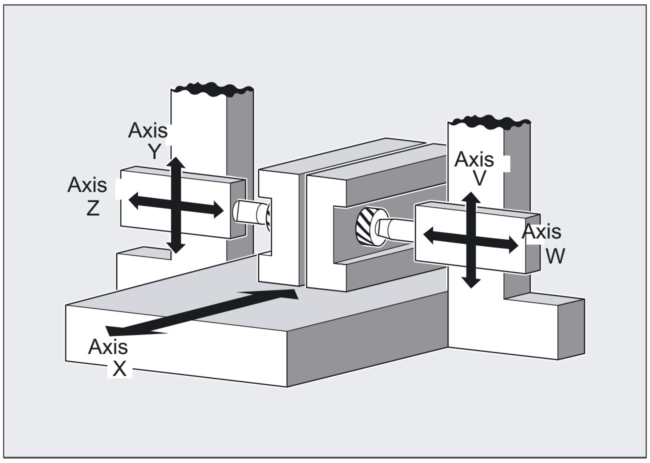


Figure 9-29 Coupled motion, programming example

```
…
N100 TRAILON(V,Y)          ; Activation of 1st coupled axis grouping
N110 TRAILON(W,Z,-1)       ; Activation of 2nd coupled axis grouping, Negative
                             coupling factor: Coupled-motion axis traverses in the
                             opposite direction from leading axis.
N120 G0 Z10                ; Infeed of Z and W axes in opposite axial directions.
N130 G0 Y20                ; Infeed of Y and V axes in same axis  direction.
…
N200 G1 Y22 V25 F200       ; Overlaying of a dependent and independent movement of
                             coupled motion axis V.
…
TRAILOF(V,Y)               ; Deactivation of 1st coupled axis grouping.
TRAILOF(W,Z)               ; Deactivation of 2nd coupled axis grouping.
```

Further Information

Axis types

A coupled axis grouping can consist of any desired combinations of linear and rotary axes. A simulated axis can also be defined as a leading axis.

Coupled-motion axes

Up to two leading axes can be assigned simultaneously to a trailing axis. The assignment is made in different combinations of coupled axes.

A coupled-motion axis can be programmed with the full range of available motion commands (`G0`, `G1`, `G2`, `G3`, etc.). The coupled axis not only traverses the independently defined paths, but also those derived from its leading axes on the basis of coupling factors.

Dynamics limit

The dynamics limit is dependent on the type of activation of the coupled axis grouping:

- Activation in part program

  If activation is performed in the part program and all leading axes are active as program axes in the activated channel, the dynamic response of all coupled-motion axes is taken into account during traversing of the leading axis to avoid overloading the coupled-motion axes.

  If activation is performed in the part program with leading axes that are not active as program axes in the activating channel ($AA_TYP ≠ 1), then the dynamic response of the coupled-motion axes is not taken into account during traversing of the leading axis. This can cause the overloading of coupled-motion axes with a dynamic response which is less than that required for the coupling.

- Activation in synchronized action

  If activation is performed in a synchronized action, the dynamic response of the coupled-motion axes is not taken into account during traversing of the leading axis. This can cause the overloading of coupled-motion axes with a dynamic response which is less than that required for the coupling.

> ⚠CAUTION
>
> If a coupled axis grouping is activated:
>
> - in synchronized actions
> - in the part program with leading axes that are not program axes in the channel of the coupled-motion axes
>
> it is the specific responsibility of the user/machine manufacturer to take suitable action to ensure that the traversing of the leading axis will not cause the overloading of the coupled-motion axes.

Coupling status

The coupling status of an axis can be checked in the part program with the system variable:

$AA_COUP_ACT[<axis>]

Value	Significance
0	No coupling active
8	Coupled motion active

9.3.22.2 Master/slave group (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

Functionality

The speed/torque coupling function (master-slave) is used for mechanically-coupled axes that are driven by two separate motors.

The master-slave coupling allows the following:

- To couple slave axes to their master axis, only when the axes involved are at a standstill.
- Coupling and decoupling **rotating**, speed-controlled spindles and the dynamic configuration.

References

SINUMERIK 802D sl Function Manual for Turning, Milling, Nibbling; Speed/Torque Coupling, Master-Slave

Programming

`MASLON(Slv1,Slv2,..., )`	
`MASLOF(Slv1,Slv2,..., )`	
`MASLDEF(Slv1,Slv2,..., master axis)`	Extension for dynamic configuration
`MASLDEL(Slv1,Slv2,..., )`	Extension for dynamic configuration
`MASLOFS(Slv1, Slv2, ..., )`	Extension for slave spindle

Note

For `MASLOF/MASLOFS`, the implicit preprocessing stop is not required. Because of the missing preprocessing stop, the $P system variables for the slave axes do not provide updated values until next programming.

Significance

General

`MASLON`	Close (switch-in) a temporary coupling.
`MASLOF`	Open an active coupling. The extensions for spindles must be observed.

Dynamic configuration extension

`MASLDEF`	Coupling user-defined using machine data or also create/change from the part program.
`MASLOFS`	Open the coupling analog to `MASLOF` and automatically brake the slave spindle.
`MASLDEL`	Uncouple master/slave axis group and delete group definition.
`Slv1, Slv2, ...`	Slave axes led by a master axis.
`Master axis`	Axis, that controls defined slave axes in a master/slave group.

Programming examples

Example 1: Dynamic configuration of a master/slave coupling

Dynamic configuration of a master/slave coupling from the part program:

The axis relevant after axis container rotation must become the master axis.

```
MASLDEF(AUX,S3)      ; S3 master for AUX
MASLON(AUX)          ; Close coupling for AUX
M3=3 S3=4000         ; Clockwise direction of rotation
MASLDEL(AUX)         ; Delete configuration and  open the coupling
AXCTSWE(CT1)         ; Container rotation
```

Examples

Example 2: Actual-value coupling of a slave axis

Actual value coupling of a slave axis to the same value of the master axis using `PRESETON`.

For a permanent master/slave coupling, the actual value at the SLAVE axis is to be changed using `PRESETON`.

```
N37262 $MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0           ; Briefly switch-out the
                                                        permanent coupling.
N37263 NEWCONF
N37264 STOPRE
MASLOF(Y1)                                            ; Temporary coupling open.
N5 PRESETON(Y1,0,Z1,0,B1,0,C1,0,U1,0)                 ; Set the actual value of
                                                        the non-referenced slave
                                                        axes as these are
                                                        activated with power on.
N37262 $MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1           ; Activate permanent
                                                        coupling.
N37263 NEWCONF
```

Further Information

General

`MASLOF` This instruction is executed directly for spindles in speed control mode. The slave spindles rotating at this instant keep their speeds until a new speed is programmed.

Dynamic configuration extension

`MASLDEF` Definition of a master/slave group from the part program. Beforehand, the definition was made exclusively using machine data.

`MASLDEL` The instruction cancels assignment of the slave axes to the master axis and simultaneously opens the coupling, like MASLOF.

The master/slave definitions specified in the machine data are kept.

`MASLOFS` MASLOFS can be used to automatically brake slave spindles when the coupling is opened.

For axes and spindles in the positioning mode, the coupling is only closed and opened at standstill (zero speed).

Note

For the slave axis, the actual value can be synchronized to the same value of the master axis using `PRESETON`. To do this, the permanent/slave coupling must be briefly switched-out in order to set the actual value of the non-referenced slave axis to the value of the master/axis with power on. Then the coupling is permanently re-established.

The permanent master/slave coupling is activated using the MD setting
MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1 – and this has no effect on the language commands of the temporary coupling.

Coupling behavior for spindles!

For spindles in the open-loop speed controlled mode, the coupling behavior of `MASLON`, `MASLOF`, `MASLOFS` and `MASLDEL` are explicitly defined using machine data
MD37263 $MA_MS_SPIND_COUPLING_MODE.

For the default setting with MD37263 = 0, the slave axes are coupled-in and coupled-out only when the axes involved are at standstill. `MASLOFS` corresponds to `MASLOF`.

For MD37263 = 1, the coupling instruction is immediately executed and therefore also the motion. For `MASLON` the coupling is immediately closed and for `MASLOFS` or `MASLOF` immediately opened. The slave spindles rotating at this instant in time are for `MASLOFS` automatically braked and for `MASLOF` keep their speed until a new speed is programmed.

9.4 Spindle movements

9.4.1 Spindle speed S, directions of rotation

Functionality

The spindle speed is programmed under the address S in revolutions per minute, if the machine has a controlled spindle.
The direction of rotation and the beginning or end of the movement are specified via M commands.

Programming

```
M3                          ; Spindle clockwise
M4                          ; Spindle counterclockwise
M5                          ; Spindle stop
```

Remark: For integer S values, the decimal point can be omitted, e.g. S270.

Information

If you write M3 or M4 in a **block with axis movements**, the M commands become active **before** the axis movements.
Standard setting: Axis movements will only start once the spindle has accelerated to speed (M3, M4). M5 is also issued before the axis movement. However, it does not wait for the spindle to stop. Axis motion already starts before the spindle comes to a standstill.
The spindle is stopped with the end of the program or RESET.
At the beginning of the program, the spindle speed is zero (S0).
Note: Other settings can be configured via machine data.

Programming example

```
N10 G1 X70 Z20 F3 S270 M3      ; before the axis traversing X, Z the spindle
                               accelerates to 270 r.p.m., clockwise
...
N80 S450 ...                   ; Speed change
...
N170 G0 Z180 M5                ; Z movement, spindle comes to a stop
```

9.4.2 Spindle speed limitation: G25, G26

Functionality

In the program, you can limit the limit values that would otherwise apply by writing G25 or G26 and the spindle address S with the speed limit value. At the same time the values in the setting data are overwritten.
G25 or G26 only need one separate block each. A previously programmed speed S is maintained.

Programming

G25 S...	; Lower spindle speed limitation
G26 S...	; Upper spindle speed limitation

Information

The outmost limits of the spindle speed are set in machine data. By entering via the operator panel, setting data for further limitations can be activated.
For the function G96 -constant cutting speed, an additional upper limit (LIMS) can be programmed/entered.

Programming example

```
N10 G25 S12              ; Lower spindle speed limitation : 12 rev/min
N20 G26 S700             ; Upper spindle speed limitation : 700 rev/min
```

9.4.3 Spindle positioning

9.4.3.1 Spindle positioning (SPOS, SPOSA, M19, M70, WAITS)

Functionality

`SPOS`, `SPOSA` or `M19` can be used to set spindles to specific angular positions, e.g. during tool change.

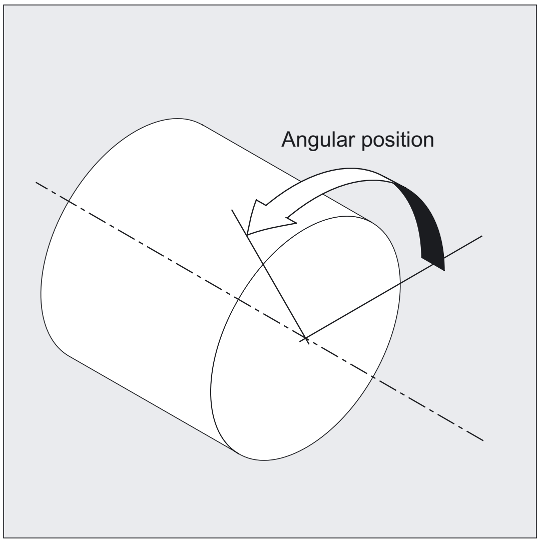


Figure 9-30 Angular position

`SPOS`, `SPOSA` and `M19` induce a temporary switchover to position-controlled mode until the next `M3`/`M4`/`M5`/`M41` to `M45`.

Positioning in axis mode

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.

End of positioning

The end-of-motion criterion when positioning the spindle can be programmed using `FINEA`, `CORSEA`, `IPOENDA` or `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.

Synchronization

In order to synchronize spindle movements, `WAITS` can be used to wait until the spindle position is reached.

Conditions

The spindle to be positioned must be capable of operation in position-controlled mode.

Programming

Position spindle:

`SPOS=<value>/SPOS[<n>]=<value>`

`SPOSA=<value>/SPOSA[<n>]=<value>`

`M19/M<n>=19`

Switch spindle over to axis mode:

`M70/M<n>=70`

Define end-of-motion criterion:

`FINEA/FINEA[S<n>]`

`COARSEA/COARSEA[S<n>]`

`IPOENDA/IPOENDA[S<n>]`

`IPOBRKA/IPOBRKA(<axis>[,<instant in time>])` ; Programming in a separate NC block.

Synchronize spindle movements:

`WAITS/WAITS(<n>,<m>)` ; Programming in a separate NC block.

Significance

`SPOS/SPOSA`: Set spindle to specified angle

`SPOS` and `SPOSA` have the same functionality but differ in their block change behavior:

- `SPOS` delays the enabling of the NC block until the position has been reached.
- `SPOSA` enables the NC block even if the position has not been reached.

`<n>`: Number of the spindle to be positioned.

If a spindle number is not specified or if the spindle number is set to "0", `SPOS` or `SPOSA` will be applied to the master spindle.

`<value>`: Angular position to which the spindle is to be set.

Unit: degrees

Type: REAL

The following options are available about programming the position approach mode:

`=AC(<value>)`: Absolute dimensions

Range of values: 0 … 359,9999

`=IC(<value>)`: Incremental dimensions

Range of values: 0 … ±99 999,999

`=DC(<value>)`: Approach absolute value directly

`=ACN(<value>)`: Absolute dimension, approach in negative direction

`=ACP(<value>)`: Absolute dimension, approach in positive direction

`=<value>`: as `DC(<value>)`

`M<n>=19`: Set the master spindle (`M19` or `M0=19`) or spindle number <n> (`M<n>=19`) to the angular position preset with SD43240 $SA_M19_SPOS with the position approach mode preset in SD43250 $SA_M19_SPOSMODE.

The NC block is not enabled until the position has been reached.

`M<n>=70`: Switch the master spindle (`M70` or `M0=70`) or spindle number <n> (`M<n>=70`) over to axis mode.

No defined position is approached. The NC block is enabled after the switchover has been performed.

`FINEA`: Motion end when "Exact stop fine" reached

`COARSEA`: Motion end when "Exact stop coarse" reached

`IPOENDA`: End of motion on reaching "interpolator stop"

`S<n>`: Spindle for which the programmed end-of-motion criterion is to be effective

`<n>`: Spindle number

If a spindle is not specified in `[S<n>]` or a spindle number of "0" is specified, the programmed end-of-motion criterion will be applied to the master spindle.

`IPOBRKA`: A block change is possible in the braking ramp.

`<axis>`: Channel axis identifier

`<instant in time>`: Instant in time of the block change with reference to the braking ramp

Unit:	Percent
Range of values:	100 (application point of the braking ramp) to 0 (end of the braking ramp)

If a value is not assigned to the `<instant in time>` parameter, the current value of the setting data is applied:

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

Note:

`IBOBRKA` with an instant in time of "0" is identical to `IPOENDA`.

`WAITS`: Synchronization command for the specified spindle(s)

The subsequent blocks are not processed until the specified spindle(s) programmed in a previous NC block with `SPOSA` has (have) reached its (their) end position(s) (with exact stop fine).

`WAITS` after `M5`: Wait for the specified spindle(s) to come to a standstill.

`WAITS` after `M3`/`M4`:	Wait for the specified spindle(s) to reach their setpoint speed.
`<n>,<m>`:	Numbers of the spindles to which the synchronization command is to be applied. If a spindle number is not specified or if the spindle number is set to "0", `WAITS` will be applied to the master spindle.

Note

Three spindle positions are possible for each NC block.

Note

With incremental dimensions `IC(<value>)`, spindle positioning can take place over several revolutions.

Note

If position control was activated with `SPCON` prior to `SPOS`, this remains active until `SPCOF` is issued.

Note

The control detects the transition to axis mode automatically from the program sequence. Explicit programming of M70 in the part program is, therefore, essentially no longer necessary. However, M70 can continue to be programmed, e.g to increase the legibility of the part program.

Programming examples

Example 1: Position spindle with negative direction of rotation

Spindle 2 is to be positioned at 250° with negative direction of rotation:

```
N10 SPOSA[2]=ACN(250)        ; The spindle is decelerated if necessary and
                               accelerated in the opposite direction to that of the
                               positioning movement.
```

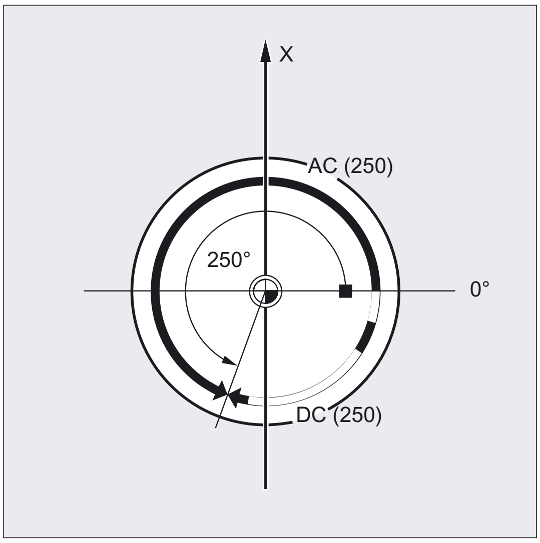


Figure 9-31 Position specified in degrees

Example 2: Spindle positioning in axis mode

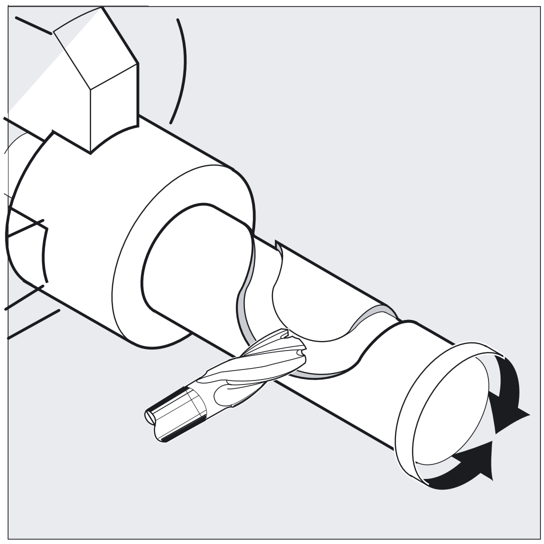


Figure 9-32 Spindle positioning

Program variant 1:

```
...
N10 M3 S500
...
N90 SPOS[2]=0                ; Position control on, spindle 2 positioned to 0, axis
                               mode can be used in the next block.
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.
```

Program variant 2:

```
...
N10 M3 S500
...
N90 M2=70                    ; Spindle 2 switches 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.
```

Example 3: Drill cross holes in turned part

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.

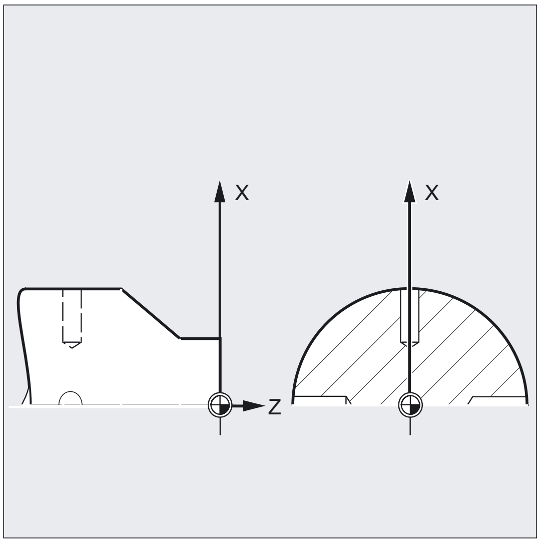


Figure 9-33 Rotating part, spindle

```
....
N110 S2=1000 M2=3            ; Switch on cross drilling attachment.
N120 SPOSA=DC(0)             ; Set main spindle to 0° immediately,
                               the program will advance to the next block straight
                               away.
```

```
N125 G0 X34 Z-35            ; Switch on the drill while the spindle is taking up
                              position.
N130 WAITS                  ; Wait for the main spindle to reach 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.)
N140G0 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° relative 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
...
```

Further information

Positioning with SPOSA

The block step enable or program execution is not affected by SPOSA. The spindle positioning can be performed during execution of subsequent NC 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).

NOTICE
If a command, which implicitly causes a preprocessing stop, is read in a following block, execution of this block is delayed until all positioning spindles are stationary.

Positioning with SPOS/M19

The block step enabling condition is met when all functions programmed in the block reach their end-of-block criterion (e.g. all auxiliary functions acknowledged by the PLC, all axes at their end point) and the spindle reaches the programmed position.

Velocity of the movements:

The velocity and the delay response for positioning are stored in the machine data. The configured values can be modified by programming or by synchronized actions.

Specification of spindle positions:

As the G90/G91 commands are not effective here, the corresponding dimensions apply explicitly, e.g. AC, IC, DC, ACN, ACP. If no specifications are made, traversing automatically takes place as for DC.

Synchronize spindle movements with WAITS

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 reach their positions.

Example:

```
N10 SPOSA[2]=180 SPOSA[3]=0
...
N40 WAITS(2,3)                    ; The block waits until spindles 2 and 3 have
                                    reached the positions specified in block N10.
```

WAITS can be used after M5 to wait until the spindle(s) has (have) stopped. WAITS can be used after M3/M4 to wait until the spindle(s) has (have) reached the specified speed/direction of rotation.

Note

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).

Position spindle from rotation (M3/M4)

When M3 or M4 is active, the spindle comes to a standstill at the programmed value.

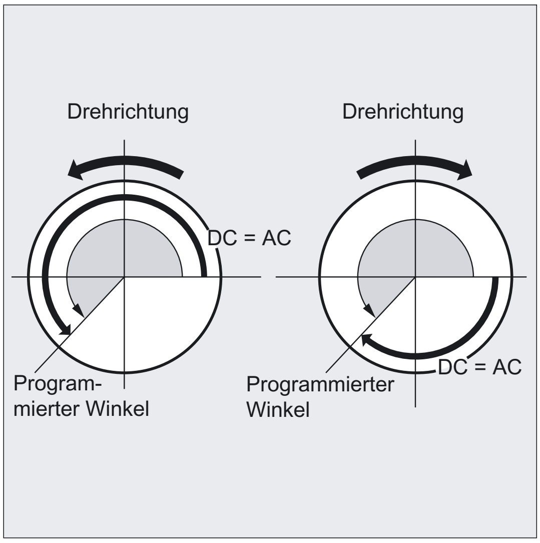


Figure 9-34 Direction of spindle rotation

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 taken. With IC, the spindle rotates additionally to the specified value starting at the current spindle position.

Position a spindle from standstill (M5)

The exact programmed distance is traversed from standstill (M5).

9.4.4 Gear stages

Functionality

Up to 5 gear stages can be configured for a spindle for speed / torque adaptation.

Programming

The relevant gear stage is selected in the program via M commands:

```
M40             ; Automatic gear stage selection
M41 to M45      ; Gear stages 1 to 5
```

9.4.5 2. Spindle

Function

With SINUMERIK 802D sl plus and 802D sl pro, a 2nd spindle is provided.

With these controls, the kinematic transformation functions TRANSMIT and TRACYL are possible for milling on turning machines. These functions require a 2nd spindle for the driven milling tool.
In these functions, the main spindle is operated as rotary axis.

Master spindle

A series of functions is associated with the master spindle that can only be used with this spindle:

```
G95                     ; Rev. feedrate
G96, G97                ; Constant cutting rate
LIMS                    ; upper speed limit for G96, G97
G33, G34, G35, G331,    ; Thread cutting, thread interpolation
G332
M3, M4, M5, S...        ; simple specifications for direction of rotation, stop and
                        speed
```

The master spindle is defined via configuration (machine data). Generally it is the main spindle (spindle 1). A different spindle can be defined as master spindle in the program:

```
SETMS(n)                ; spindle n (= 1 or 2) is the master spindle as of now.
```

Switching back can also be performed via:

```
SETMS                   ; configured master spindle is now master spindle again
SETMS (1)               ; Spindle 1 is now master spindle again.
```

The definition of the master spindle changed in the program is only valid until End of program/program abort. Thereafter, the configured master spindle is again active.

Programming via spindle number

Some spindle functions can also be selected via the spindle number:

```
S1=..., S2=...              ; Spindle speed for spindle 1 or 2
M1=3, M1=4, M1=5            ; Specifications for direction of rotation, stop for
                            spindle 1
M2=3, M2=4, M2=5            ; Specifications for direction of rotation, stop for
                            spindle 2
M1=40, ..., M1=45           ; gear stages for spindle 1 (if available)
M2=40, ..., M2=45           ; gear stages for spindle 2 (if available)
SPOS [n]                    ; Position spindle n
SPI(n)                      ; Converts spindle number n to axis identifier,
                            ; e.g. "SP1" or "CC"
                            ; n must be a valid spindle number (1 or 2)
                            ; The functions of spindle identifiers SPI(n) and Sn are
                            identical.
$P_S[n]                     ; Last programmed speed of spindle n
$AA_S[n]                    ; Actual speed of spindle n
$P_SDIR[ n ]                ; Last programmed direction of rotation of spindle n
$AC_SDIR[ n ]               ; Current direction of rotation of spindle n
```

2 spindles installed

The following can be interrogated in the program via the system variable:

```
$P_NUM_SPINDLES             ; Number of configured spindles (in the channel)
$P_MSNUM                    ; Number of programmed master spindle
$AC_MSNUM                   ; Number of the active master spindle
```

9.5 Special turning functions

9.5.1 Constant cutting rate: G96, G97

Functionality

Requirement: A controlled spindle must be present.

With activated G96 function, the spindle speed is adapted to the currently machined workpiece diameter (transverse axis) such that a programmed cutting rate S remains constant on the tool edge:
Spindle speed times diameter = constant.

The S word is evaluated as the cutting rate as of the block with G96. G96 is modally effective until cancellation by another G function of the group (G94, G95, G97).

Programming

```
G96 S... LIMS=... F...     ; Constant cutting speed ON
G97                        ; Constant cutting speed OFF

S                          ; Cutting rate, unit of measurement m/min.
LIMS=                      ; Upper limit speed of the spindle with G96, G97 effective
F                          ; Feedrate in mm/revolution – as for G95
```

Remark:
If G94 instead of G95 was active before, a new appropriate F value must be written!

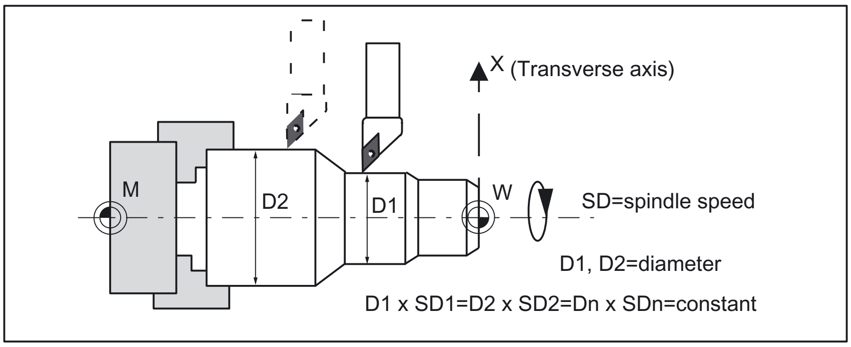


Figure 9-35 Constant cutting rate G96

Rapid traverse

With rapid traverse G0, there is no change in speed.

Exception: If the contour is approached at rapid traverse and the next block contains an interpolation type G1 or G2, G3, CIP, CT (contour block), then the speed for the contour block is applied already in the approach block with G0.

Upper speed limit LIMS=

During machining from large to small diameters, the spindle speed can increase significantly. In this case, it is recommended the upper spindle speed limitation LIMS=... . LIMS is only effective with G96 and G97.
By programming LIMS=..., the value entered into the setting data (SD 43230: SPIND_MAX_VELO_LIMS) is overwritten. This SD takes effect when LIMS is not written.
The upper limit speed programmed with G26 or defined via machine data cannot be overwritten with LIMS=.

Deactivate constant cutting rate: G97

The function "Constant cutting rate" is deactivated by G97. If G97 is active, a programmed S word is given in RPM as the spindle speed .
If no new S word is programmed, the spindle turns at the last defined speed with G96 function active.

Programming example

```
N10 ... M3                ; Spindle's direction of rotation
N20 G96 S120 LIMS=2500    ; Activate constant cutting speed, 120 m/min, speed
                          limit 2,500 r.p.m.
N30 G0 X150               ; no change in speed, because block N31 with G0
N31 X50 Z...              ; no change in speed, because block N32 with G0
N32 X40                   ; Approach on contour, new speed is automatically set as
                          is required for the beginning of block N40
N40 G1 F0.2 X32 Z...      ; Feedrate 0.2 mm/revolution
...
N180 G97 X... Z...        ; Deactivating constant cutting rate
N190 S...                 ; new spindle speed, r.p.m.
```

Information

The G96 function can also be deactivated with G94 or G95 (same G group). In this case, the last programmed spindle speed S is active for the remaining machining sequence if no new S word is programmed.

The programmable offset TRANS or ATRANS (see section of that name) should not be used on the transverse axis X or used only with low values. The workpiece zero point should be located at the turning center. Only then is the exact function of G96 guaranteed.

9.5.2 Rounding, chamfer

Functionality

You can insert the chamfer (CHF or CHR) or rounding (RND) elements into a contour corner. If you wish to round several contour corners sequentially by the same method, use "Modal rounding" (RNDM).

You can program the feedrate for the chamfer/rounding with FRC (non-modal) or FRCM (modal). If FRC/FRCM is not programmed, the normal feedrate F is applied.

Programming

CHF=...	; Insert chamfer, value: Length of chamfer
CHR=...	; Insert chamfer, value: Side length of the chamfer
RND=...	; Insert rounding, value: Radius of chamfer
RNDM=...	; Modal rounding:
	Value >0: Radius of chamfer, modal rounding ON
	This rounding is inserted in all contour corners.
	Value = 0: Modal rounding OFF...
FRC=...	; Non-modal feedrate for chamfer/rounding
	Value >0, feedrate in mm/min (G94) or mm/rev. (G95)
FRCM=...	; Modal feedrate for chamfer/rounding:
	Value >0: Feedrate in mm/min (G94) or mm/rev. (G95),
	Modal feedrate for chamfer/rounding ON
	Value = 0: Modal feedrate for chamfer/rounding OFF
	Feedrate F applies to the chamfer/rounding.

Information

The chamfer/rounding functions are executed in the current planes G17 to G19.

The appropriate instruction CHF= ... or CHR=... or RND=... or RNDM=... is written in the block with axis movements leading to the corner.

The programmed value for chamfer and rounding is automatically reduced if the contour length of an involved block is insufficient.

No chamfer/rounding is inserted, if

- more than three blocks in the connection are programmed that do not contain any information for traversing in the plane,
- or a plane change is carried out.

F, FRC,FRCM are not active when a chamfer is traversed with G0.

If the feedrate F is active for chamfer/rounding, it is by default the value from the block which leads away from the corner. Other settings can be configured via machine data.

Chamfer CHF or CHR

A linear contour element is inserted between **linear and circle contours** in any combination. The edge is broken.

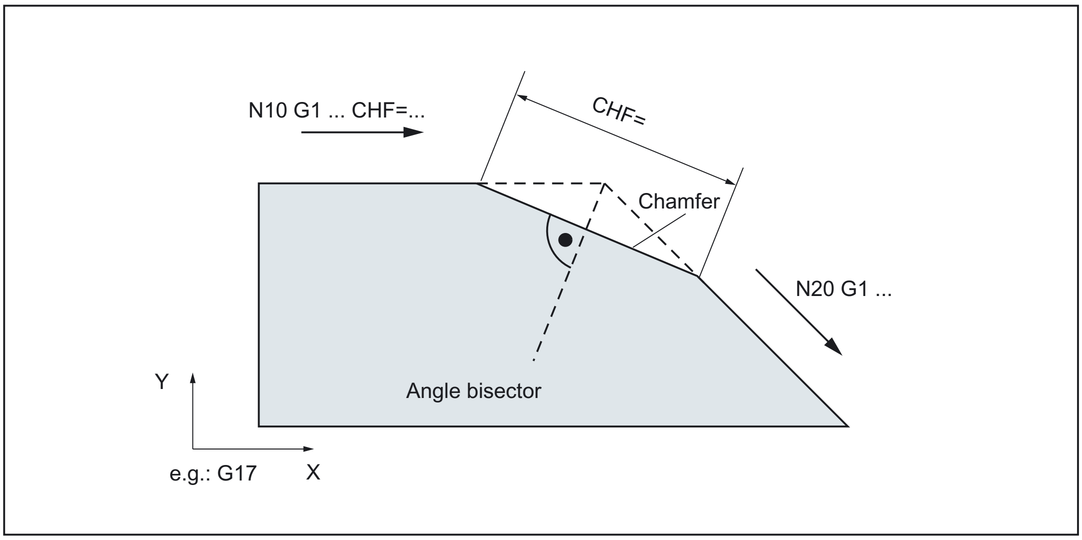


Figure 9-36 Inserting a chamfer with CHF using the example: Between two straight lines

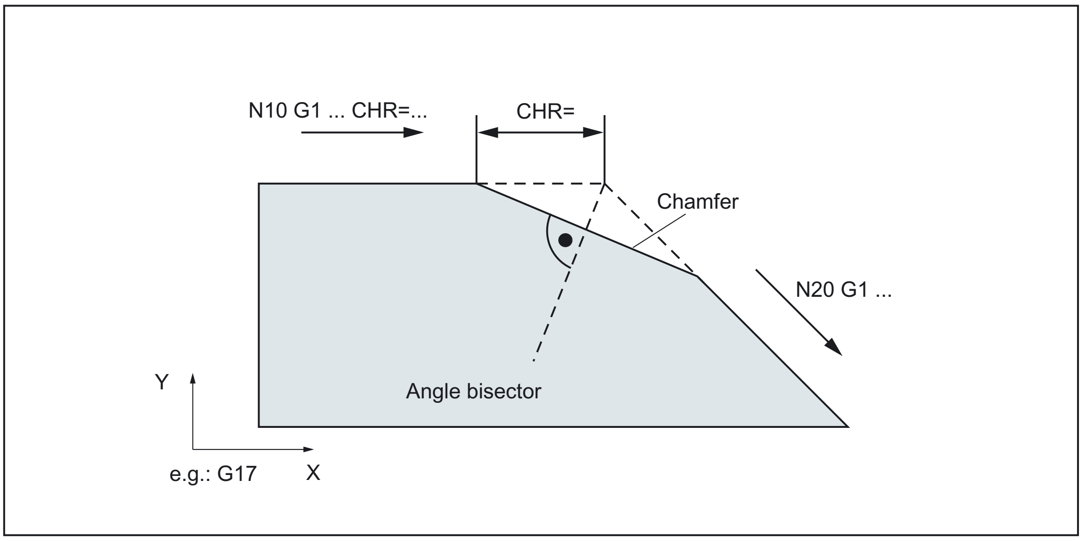


Figure 9-37 Inserting a chamfer with CHR using the example: Between two straight lines

Programming examples of chamfer

```
N5 G17 G94 F300 ...
N10 G1 X... CHF=5             ; Insert chamfer with chamfer length of 5 mm
N20 X... Y...
...
N100 G1 X... CHR=7            ; Insert chamfer with leg length of 7 mm
N110 X... Y...
...
N200 G1 FRC=200 X... CHR=4    ; Insert chamfer with feedrate FRC
N210 X... Y...
```

Rounding RND or RNDM

A circle contour element can be inserted with tangential connection between the linear and circle contours in any combination.

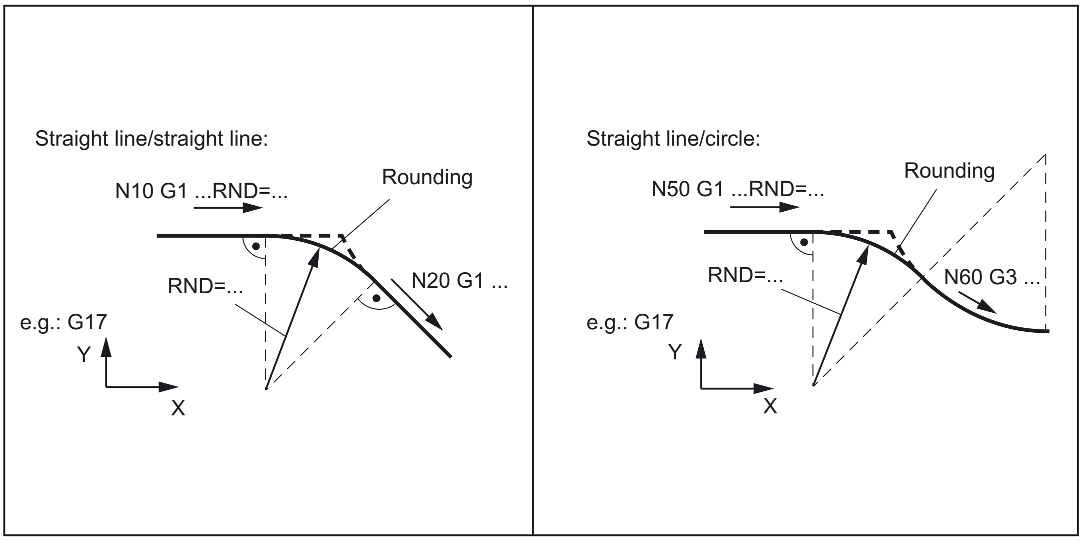


Figure 9-38 Examples for inserting roundings

Programming examples for rounding

```
N5 G17 G94 F300 ...
N10 G1 X... RND=8             ; Insert 1 rounding with radius 8 mm, feedrate F
N20 X... Y...
...
N50 G1 X... FRCM= 200 RNDM=7.3  ; Modal rounding, radius 7.3 mm with special
                                feedrate FRCM (modal)
N60 G3 X... Y...              ; continue inserting this rounding - to N70
N70 G1 X... Y... RNDM=0       ; Modal rounding OFF
...
```

9.5.3 Contour definition programming

Functionality

If direct end point values for the contour are not visible in a machining drawing, angle values can also be used for straight line determination. In a contour corner, you can insert the elements chamfer or rounding. The corresponding instruction CHR= ... or RND=... is written in the block that leads to the corner.
Contour definition programming can be used in blocks with G0 or G1.
Theoretically, any number of straight-line blocks can be combined and a rounding or chamfer inserted in between. Every straight line must be clearly identified by point values and/or angle values.

Programming

ANG=...	; Specification of angle to define a straight line
RND=...	; Insert rounding, value: Radius of chamfer
CHR=...	; Insert chamfer, value: Side length of the chamfer

Information

If radius and chamfer are programmed in one block, only the radius is inserted regardless of the programming sequence.

Angle ANG=

An angle can be entered to uniquely define the straight line path if only one end point coordinate of the plane is known for a straight line or for contours across multiple blocks the cumulative end point. The angle is always referred to the Z axis (normal case: G18 active). Positive angles are aligned counterclockwise.

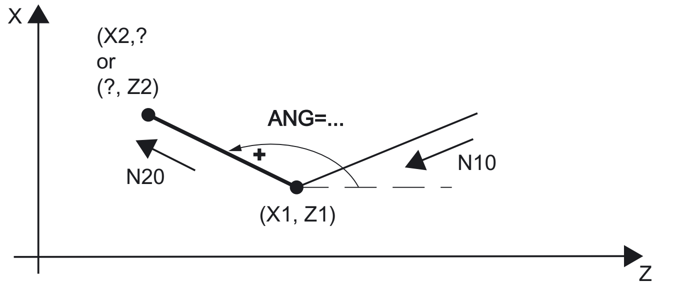
Contour	Programming
 X (X2,? or (?, Z2) ANG=... + N20 N10 (X1, Z1) Z	End point in N20 not fully known N10 G1 X1 Z1 N20 X2 ANG=... or: N10 G1 X1 Z1 N20 Z2 ANG=... The values are only symbolic

Figure 9-39 Angle value for determination of a straight line

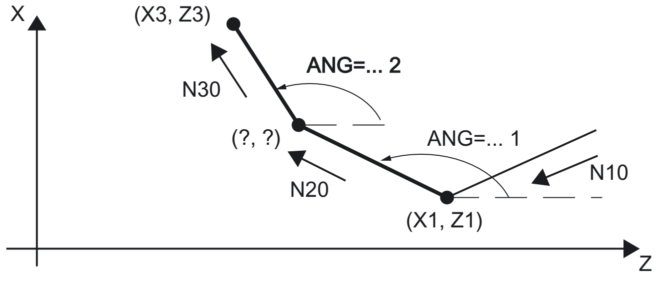
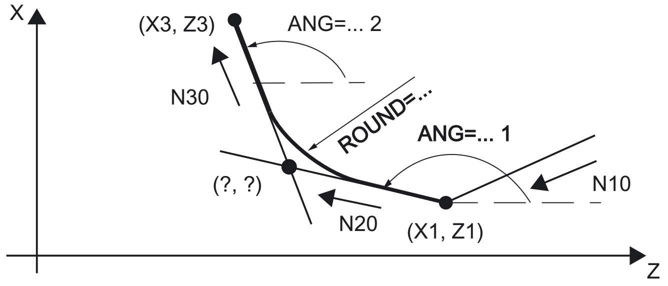
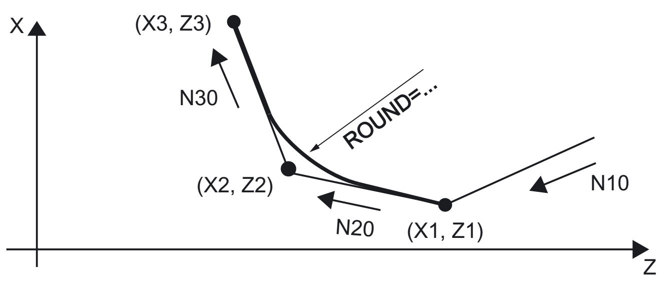
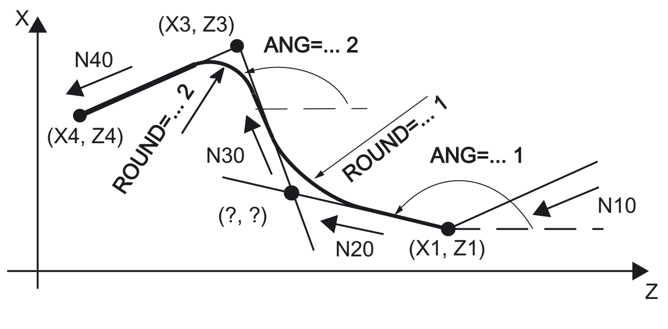
Contour	Programming
 X (X3, Z3) ANG=... 2 N30 (?, ?) ANG=... 1 N10 N20 (X1, Z1) Z	End point in N20 unknown N10 G1 X1 Z1 N20 ANG=... 1 N30 X3 Z3 ANG=... 2 The values are only symbolic
 X (X3, Z3) ANG=... 2 N30 ROUND=... ANG=... 1 (?, ?) N10 N20 (X1, Z1) Z	End point in N20 unknown Insert a rounding: N10 G1 X1 Z1 N20 ANG=... 1 RND=... analog Insert a chamfer: N10 G1 X1 Z1 N20 ANG=... 1 CHR=... N30 X3 Z3 ANG=... 2
 X (X3, Z3) N30 ROUND=... (X2, Z2) N10 N20 (X1, Z1) Z	End point in N20 known Insert a rounding: N10 G1 X1 Z1 N20 X2 Z2 RND=... N30 X3 Z3 analog Insert a chamfer: N10 G1 X1 Z1 N20 X2 Z2 CHR=... N30 X3 Z3 N30 X3 Z3 ANG=... 2
 X (X3, Z3) N40 ANG=... 2 (X4, Z4) ROUND=... 2 N30 ROUND=... 1 ANG=... 1 (?, ?) N10 N20 (X1, Z1) Z	End point in N20 unknown Insert a rounding: N10 G1 X1 Z1 N20 ANG=... 1 RND=... 1 N30 X3 Z3 ANG=... 2 RND=... 2 N40 X4 Z4 analog Insert a chamfer: N10 G1 X1 Z1 N20 ANG=... 1 CHR=... 1 N30 X3 Z3 ANG=... 2 CHR=... 2 N40 X4 Z4

Figure 9-40 Examples of multi-block contours

9.6 Tool and tool offset

9.6.1 General information (turning)

Functionality

During program creation for the workpiece machining, you do not have to take tool lengths or cutting radius into consideration. You program the workpiece dimensions directly, e.g. according to the drawing.

The tool data must be entered separately in a special data area.
In the program, you will merely call the required tool with its offset data. The control system performs the required path compensations based on this data to create the described workpiece.

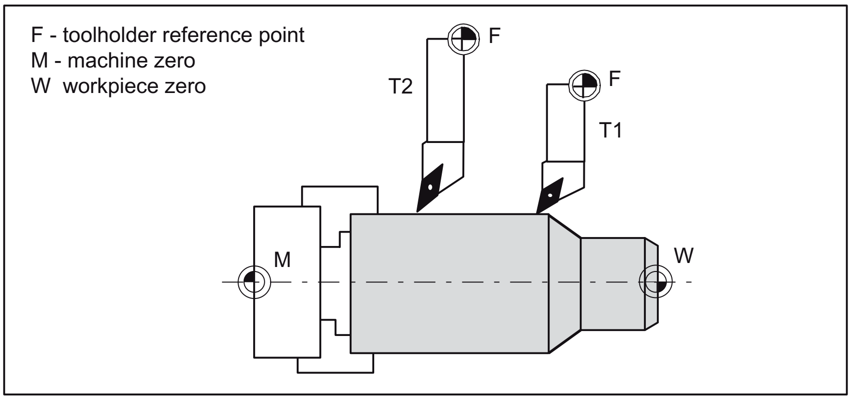


Figure 9-41 Machining a workpiece with different tool dimensions

See also

Entering tools and tool offsets (Page 36)

9.6.2 Tool T (turning)

Functionality

The tool selection takes place when the T word is programmed. Whether this is a **tool change** or only a **preselection**, is defined in the machine data:

- A tool change (tool call) takes place directly with the T word (e.g. typical for tool turrets on turning machines) or
- the change takes place after the preselection with the T word by an additional instruction **M6.**

  Note:

  If a certain tool was activated, it remains stored as an active tool even beyond the end of the program and after switching off / switching on the control system.

  If you change a tool manually, input the change also in the control system so that the control system 'knows' the correct tool. For example, you can start a block with the new T word in MDA mode.

Programming

```
T...                        ; Tool number: 1 ... 32 000
```

Note

The following is the maximum that can be stored simultaneously in the control system:

- SINUMERIK 802D sl value: 32 tools
- SINUMERIK 802D sl plus: 64 tools
- SINUMERIK 802D sl pro: 128 tools

Programming example

```
Tool change without M6:
N10 T1                    ; Tool 1
...
N70 T588                  ; Tool 588
```

9.6.3 Tool offset number D (turning)

Functionality

It is possible to assign 1 to 9 data fields with different tool offset blocks (for multiple cutting edges) to a specific tool. If a special cutting edge is required, then it can be programmed using D and the appropriate number.
If a D word is not written, **D1 is automatically** effective.
If **D0** is programmed, the offsets for the tool are **ineffective**.

Programming

```
D...                    ; Tool offset number: 1 ... 9, D0: No offsets active!
```

Note

The following maximum values for tool offset blocks can be stored simultaneously in the control system:

- SINUMERIK 802D sl value: 32 data fields (D numbers)
- SINUMERIK 802D sl plus: 64 data fields (D numbers)
- SINUMERIK 802D sl pro:128 data fields (D numbers).

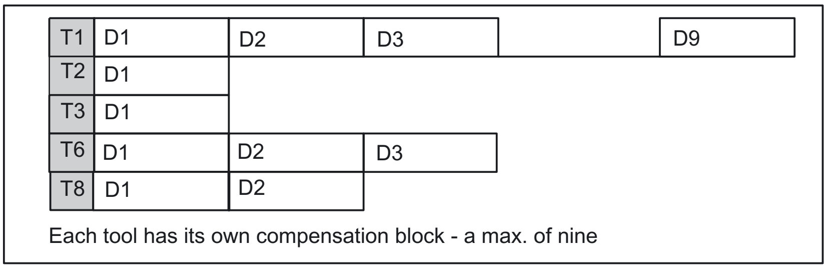


Figure 9-42 Examples for assigning tool offset numbers to tools

Information

Tool length compensations become effective **immediately** when the tool is active; when no D number was programmed with the values of D1.
The compensation is retracted with the first programmed traversing of the associated length compensation axis.

A **tool radius compensation** must also be activated by G41/G42.

Programming example

Tool change:

```
N10 T1          ; Tool 1 is activated with the associated D1
N11 G0 X...     ; The length offset compensation is overlaid here
Z...
N50 T4 D2       ; Load tool 4, D2 from T4 is active
...
N70 G0 Z...     ; D1 for tool 4 active, only cutting edge changed
D1
```

Contents of an compensation memory

- Geometrical dimensions: Length, radius.

  They consist of several components (geometry, wear). The control takes into account the components to obtain a resulting dimension (e.g. overall length 1, total radius). The respective overall dimension becomes active when the offset memory is activated.
  The way in which these values are computed in the axes is determined by the tool type and the current plane G17, G18, G19.
- Tool type

  The tool type (drill, turning tool or milling tool) determines which geometry data are required and how they will be calculated.
- Cutting edge position

  For the "turning tool" tool type, you must also enter the cutting edge position.

The following figures provide information on the required tool parameters for the respective tool type.

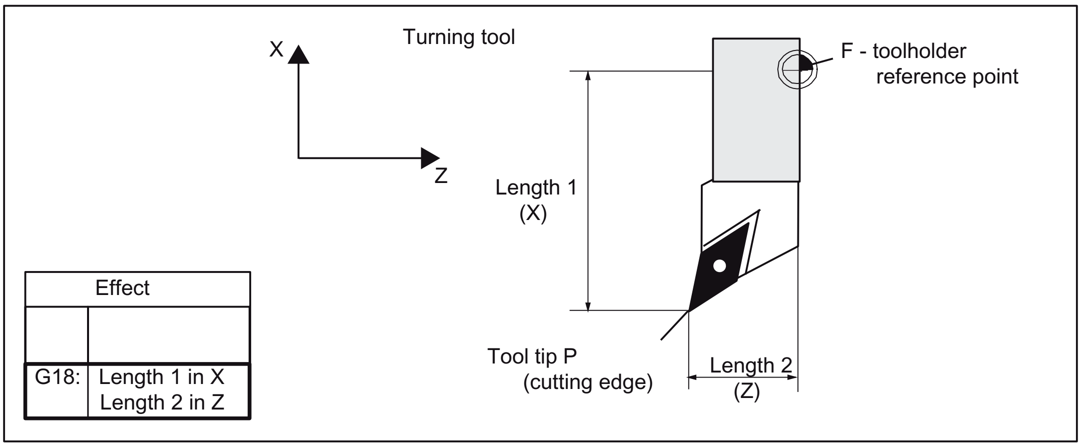

X
Z
Turning tool
F - toolholder reference point
Length 1 (X)
Tool tip P (cutting edge)
Length 2 (Z)
Effect
G18: Length 1 in X Length 2 in Z

Figure 9-43 Tool length compensation values for turning tools

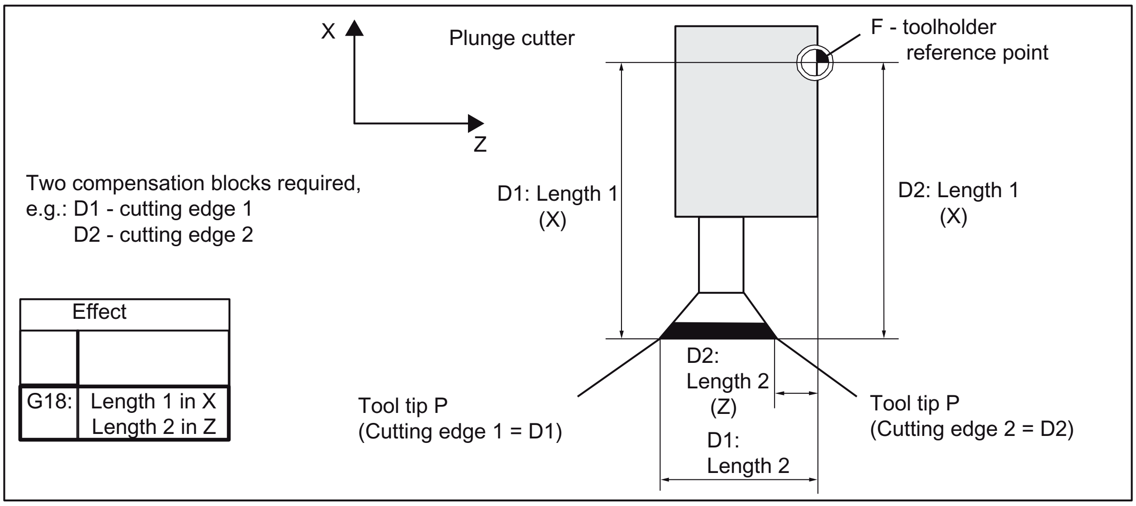
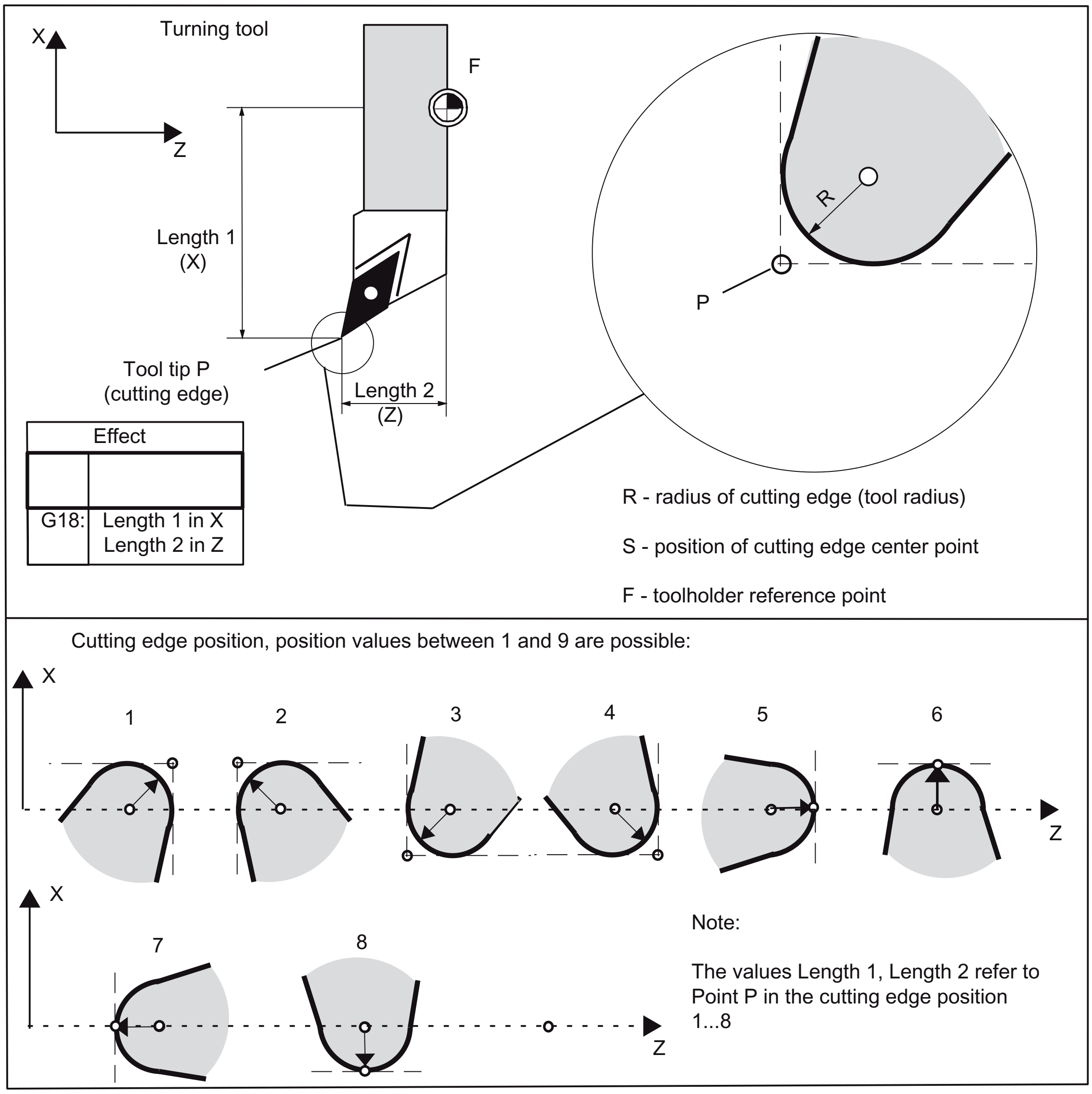

X
Z
Plunge cutter
F - toolholder reference point
Two compensation blocks required,
e.g.: D1 - cutting edge 1
D2 - cutting edge 2
D1: Length 1 (X)
D2: Length 1 (X)
Effect
G18:
Length 1 in X
Length 2 in Z
D2: Length 2 (Z)
Tool tip P (Cutting edge 1 = D1)
Tool tip P (Cutting edge 2 = D2)
D1: Length 2

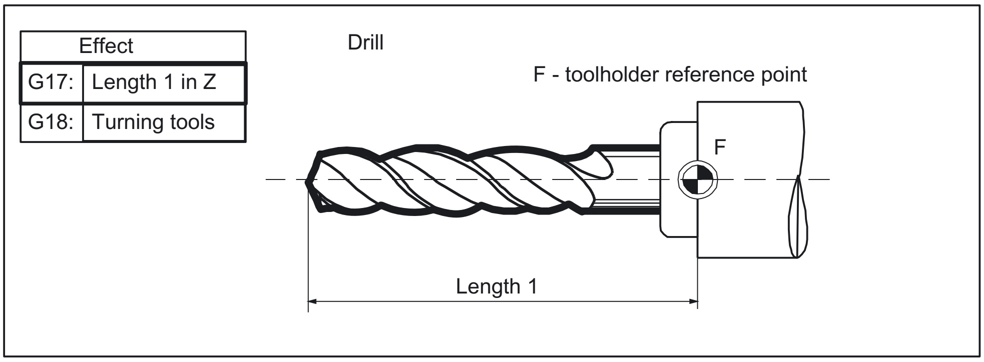
Figure 9-44 Turning tool with two cutting edges D1 and D2 - Length compensation



Effect	
G18:	Length 1 in X Length 2 in Z

X
Z
Turning tool
F
Length 1
(X)
Tool tip P
(cutting edge)
Length 2
(Z)
R
P
R - radius of cutting edge (tool radius)
S - position of cutting edge center point
F - toolholder reference point
Cutting edge position, position values between 1 and 9 are possible:
X
1
2
3
4
5
6
Z
X
7
8
Z
Note:
The values Length 1, Length 2 refer to Point P in the cutting edge position 1...8

Figure 9-45 Compensations for turning tool with tool radius compensation



Effect	
G17:	Length 1 in Z
G18:	Turning tools

Figure 9-46 Effect of the compensation for the drill

Center hole

Switch to G17 for application of a center hole. This makes the length compensation take effect for the drill in the Z axis. After drilling, the normal compensation for turning tools takes effect again with G18.

Programming example

```
N10 T...                                   ;Drill
N20 G17 G1 F... Z...                       ; Tool length offset effective in Z axis
N30 Z...
N40 G18 ....                               ; Drilling terminated
```

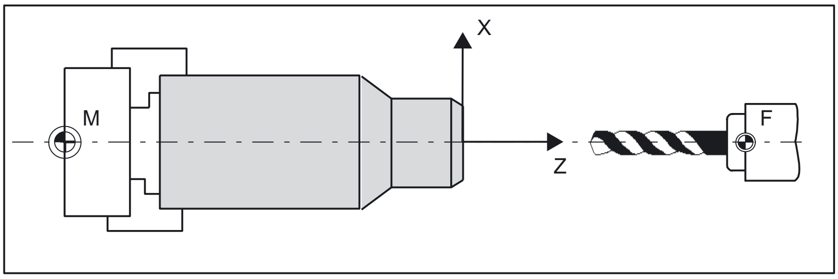


Figure 9-47 Application of a center hole

9.6.4 Selecting the tool radius compensation: G41, G42

Functionality

A tool with a corresponding D number must be active. The tool radius offset (cutting edge radius offset) is activated by G41/G42. The controller automatically calculates the required equidistant tool paths for the programmed contour for the respective current tool radius. G18 must be active.

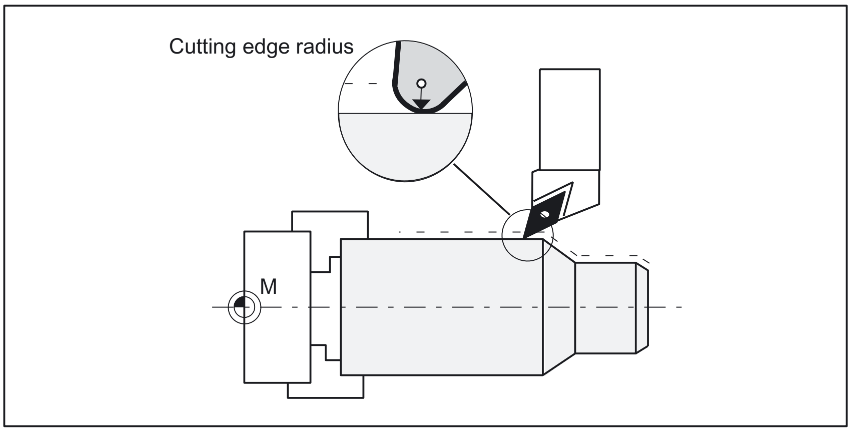


Figure 9-48 Tool radius compensation (cutter radius compensation)

Programming

G41 X... Z...	; Tool radius compensation left of contour
G42 X... Z...	; Tool radius compensation right of contour

Remark: The selection can only be made for linear interpolation (G0, G1).

Program both axes. If you only specify one axis, the second axis is automatically completed with the last programmed value.

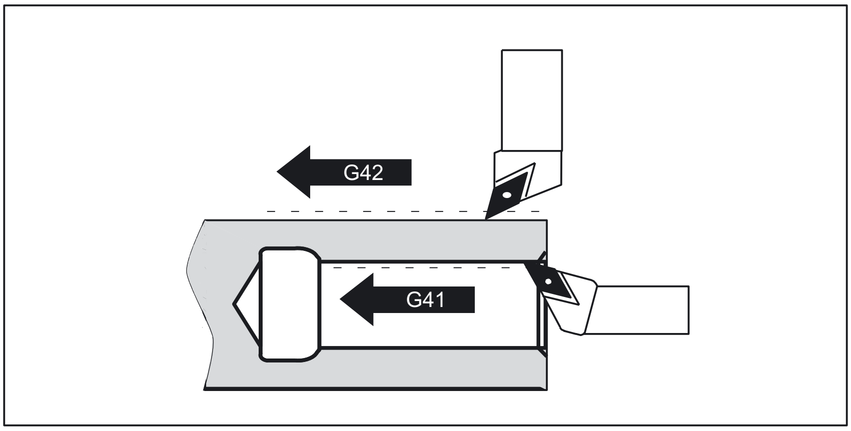


Figure 9-49 Compensation to the right/left of the contour

Starting the compensation

The tool approaches the contour on a straight line and positions itself vertically to the path tangent in the starting point of the contour.
Select the start point so as to ensure collision-free traversing.

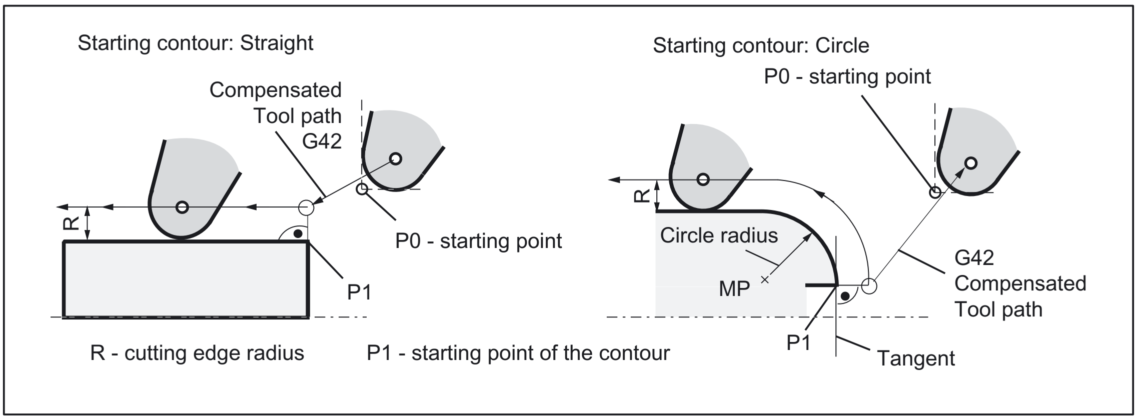


Figure 9-50 Start of the tool radius compensation with the example G42, tool point direction =3

Information

As a rule, the block with G41/G42 is followed by the block with the workpiece contour. However, the contour description may be interrupted by an intervening block that does not contain information for the contour path, e.g. only M command.

Programming example

```
N10 T... F...
N15 X... Z...               ; P0 - starting point
N20 G1 G42 X... Z...        ; Selection right of contour, P1
N30 X... Z... ;             ; Starting contour, circle or straight line
```

9.6.5 Corner behavior: G450, G451

Functionality

Using the functions G450 and G451, you can set the behavior for non-continuous transition from one contour element to another contour element (corner behavior) when G41/G42 is active.
Internal and external corners are detected by the control system automatically. For internal corners, the intersection of the equidistant paths is always approached.

Programming

G450	;Transition circle
G451	;Point of intersection

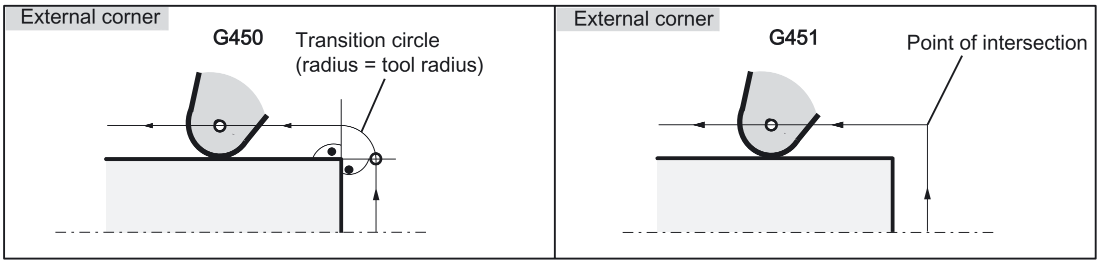


Figure 9-51 Corner behavior at an external corner

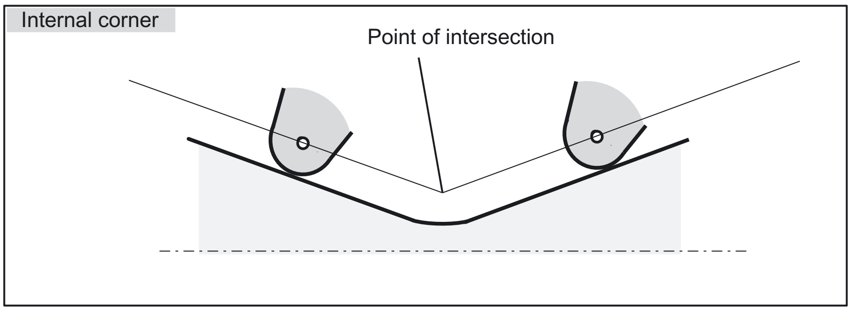


Figure 9-52 Corner behavior at an internal corner

Transition circle G450

The tool center point travels around the workpiece external corner in an arc with the tool radius. In view of the data, for example, as far as the feedrate value is concerned, the transition circle belongs to the next block containing traversing movements.

Point of intersection G451

For a G451 intersection of the equidistant paths, the point (intersection) that results from the center point paths of the tool (circle or straight line) is approached.

9.6.6 Tool radius compensation OFF: G40

Functionality

The compensation mode (G41/G42) is deselected with G40. G40 is also the switch-on position at the beginning of the program.
The tool ends the **block before G40** in the normal end position (compensation vector vertical to the tangent in the end point); independently of the start angle.
If G40 is active, the reference point is the tool tip. The tool tip then travels to the programmed point upon deselection.

Always select the end point of the G40 block such that collision-free traversing is guaranteed!

Programming

```
G40 X... Z...                    ; Tool radius compensation OFF
```

Remark: The compensation mode can only be deselected with linear interpolation (G0, G1).

Program both axes. If you only specify one axis, the second axis is automatically completed with the last programmed value.

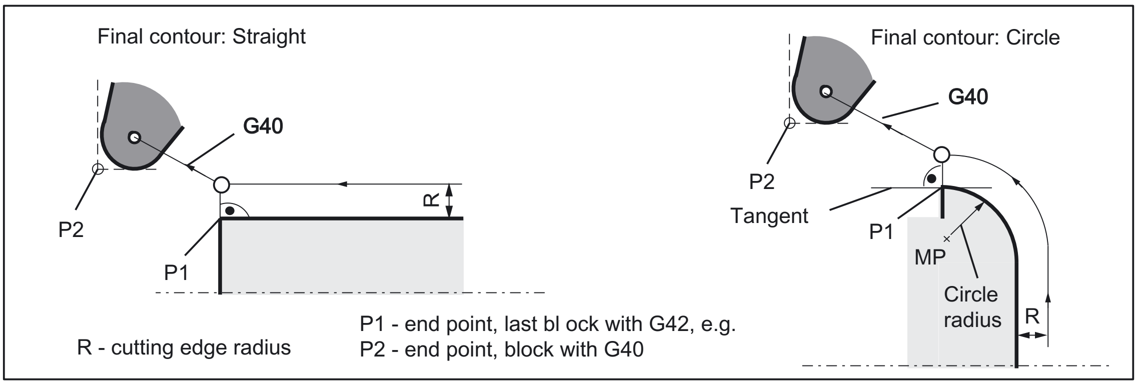


Figure 9-53 Ending the tool radius compensation with G40, with the example of G42, cutting edge position =3

Programming example

```
...
N100 X... Z...            ;Last block on the contour, circle or straight line, P1
N110 G40 G1 X... Z...     ;Switch off tool radius compensation,P2
```

9.6.7 Special cases of the tool radius compensation

Change of the compensation direction

The G41 ⇄ G42 compensation direction can be changed without writing G40 in between. The last block that uses the old compensation direction will end at the normal end position of the compensation vector in the end point. The new compensation direction is executed as a compensation start (default setting at starting point).

Repetition of G41, G41 or G42, G42

The same compensation can again be programmed without writing G40 in between. The last block before the new compensation call will end at the normal positon of the compensation vector in the end point. The new compensation is carried out as a compensation start (behavior as described for change in compensation direction).

Changing the offset number D

The offset number D can be changed in the compensation mode. A modified tool radius is active with effect from the block in which the new D number is programmed. Its complete modification is only achieved at the end of the block. In other words: The modification is traversed continuously over the entire block, also for circular interpolation.

Cancellation of compensation by M2

If the offset mode is canceled with M2 (program end) without writing the command G40, the last block with coordinates ends in the normal offset vector setting. No compensating movement is executed. The program ends with this tool position.

Critical machining cases

When programming, pay special attention to cases where the contour path for inner corners is smaller than the tool radius; and smaller than the diameter for two successive inner corners.

Such cases should be avoided.

Also check over multiple blocks that the contour contains no "bottlenecks".

When carrying out a test/dry run, use the largest tool radius you are offered.

Acute contour angles

If very sharp outside corners occur in the contour with active G451 intersection, the control system automatically switches to transition circle. This avoids long idle motions.

9.6.8 Example of tool radius compensation (turning)

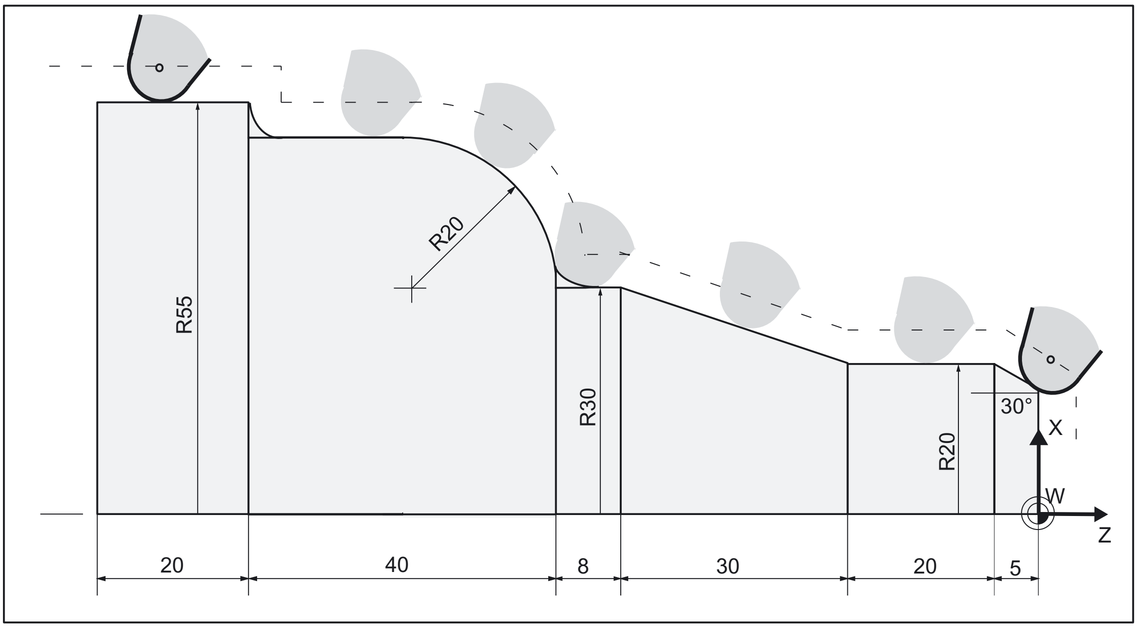


Figure 9-54 Example of tool radius compensation, cutting edge radius shown magnified

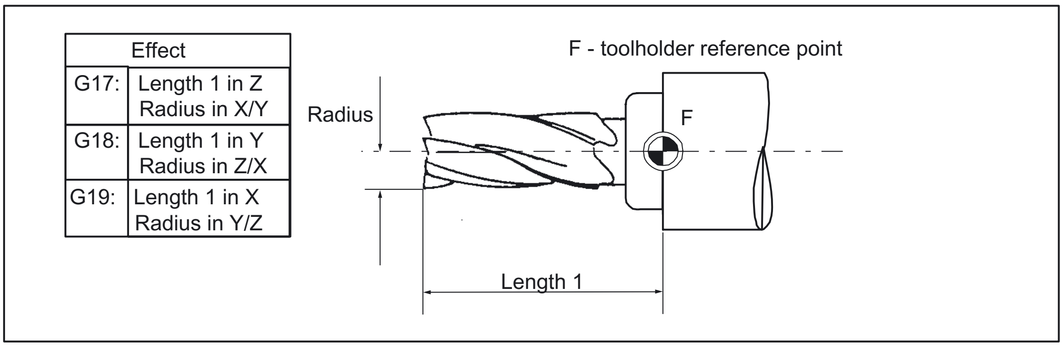
Programming example

```
N1                                         ; Contour cut
N2 T1                                      ; Tool 1 with offset D1
N10 DIAMOF F... S... M..                   ; Radius dimension, technological values
N15 G54 G0 G90 X100 Z15
N20 X0 Z6
N30 G1 G42 G451 X0 Z0                      ; Start compensation mode
N40 G91 X20 CHF=(5* 1.1223 )               ; Insert chamfer, 30 degrees
N50 Z-25
N60 X10 Z-30
N70 Z-8
N80 G3 X20 Z-20 CR=20
N90 G1 Z-20
N95 X5
N100 Z-25
N110 G40 G0 G90 X100                       ; Terminate compensation mode
N120 M2
```

9.6.9 Use of milling cutters

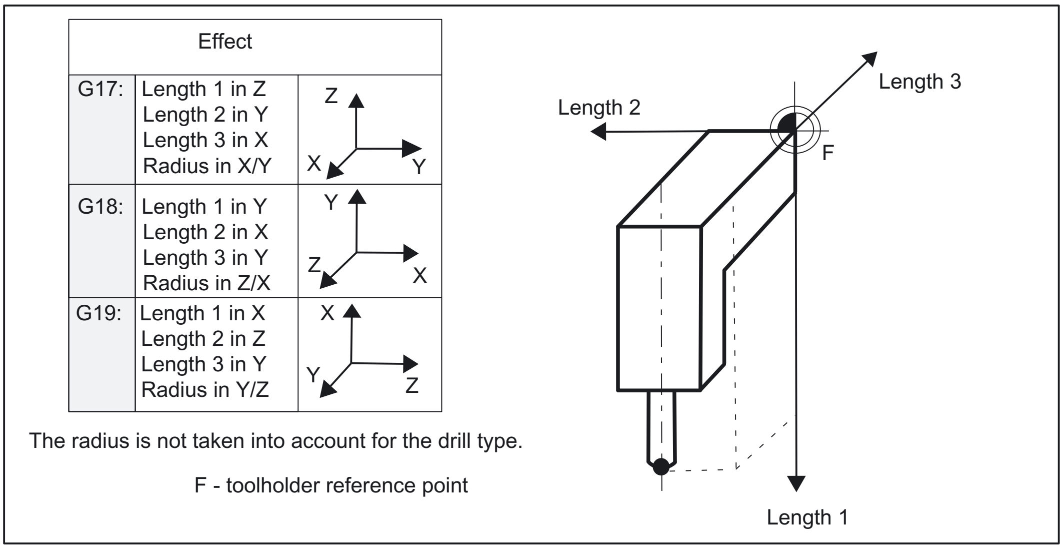
Function

The kinematic transformation functions TRANSMIT and TRACYL are associated with the use of milling cutters on turning machines.
The tool compensations for milling cutters act differently than for tools.



	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

Figure 9-55 Effect of the compensations for milling cutter tool type



	Effect	
G17:	Length 1 in Z Length 2 in Y Length 3 in X Radius in X/Y	Z X Y
G18:	Length 1 in Y Length 2 in X Length 3 in Y Radius in Z/X	Y Z X
G19:	Length 1 in X Length 2 in Z Length 3 in Y Radius in Y/Z	X Y Z

The radius is not taken into account for the drill type.

F - toolholder reference point

Figure 9-56 Effect of the tool length compensations, three-dimensional (special case)

Milling cutter radius compensation G41, G42

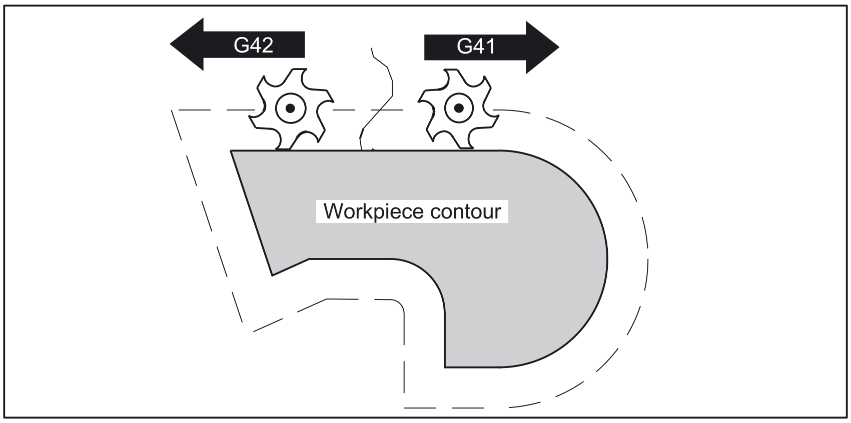


Figure 9-57 Milling cutter radius compensation to the right-left of the contour

Starting the compensation

The tool approaches the contour on a straight line and is positioned perpendicular to the path tangent in the starting point of the contour.
Select the starting point thus that collision-free travel is ensured!

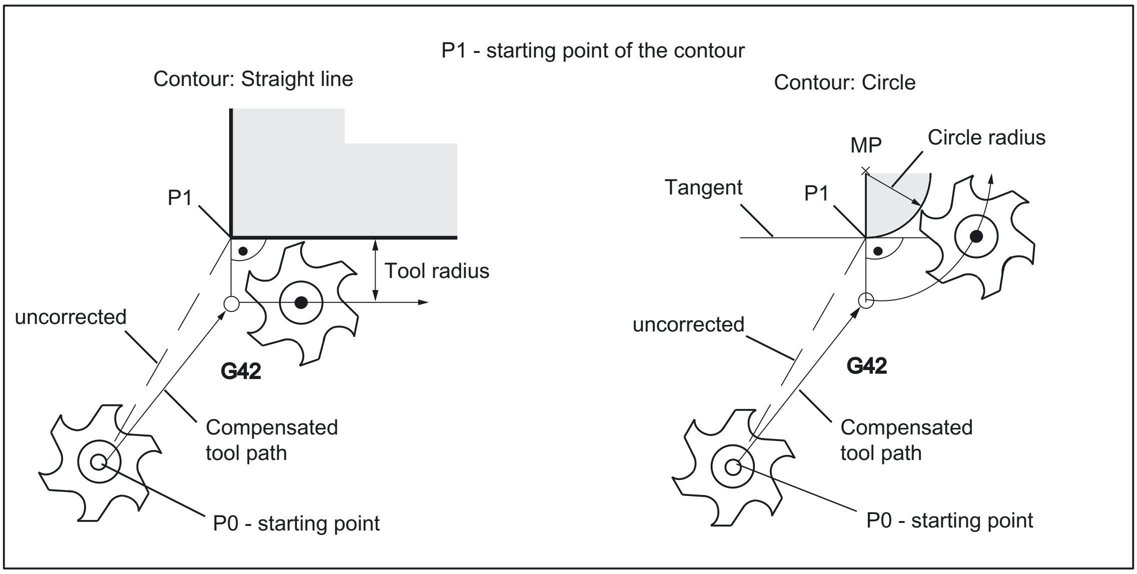


Figure 9-58 Start of cutter radius compensation with G42 as example

Information

The milling tool radius compensation acts in the same way as the radius compensation for a turning tool.

Reference

SINUMERIK 802D sl "Milling Programming and Operating Manual"

9.6.10 Special handling of tool compensation (turning)

With the SINUMERIK 802Dsl plus and 802Dsl pro, the following special actions are available for tool offset.

Influence of setting data

With the use of the following setting data, the operator / programmer can influence the calculation of the **length compensation** of the tool used:

- SD 42940: TOOL_LENGTH_CONST

  (Assignment of tool-length components to geometry axes)
- SD 42950: TOOL_LENGTH_TYPE

  (Assignment of the tool-length components independent of tool type)

Note

The modified setting data will become effective with the next cutting edge selection.

Examples

With SD 42950: TOOL_LENGTH_TYPE =2
a milling tool used is taken into account in length compensation as a turning tool:

- G17: Length 1 in Y axis, length 2 in X axis
- G18: Length 1 in X axis, length 2 in Z axis
- G19: Length 1 in Z axis, length 2 in Y axis

With SD 42940: TOOL_LENGTH_CONST =18
the length assignment is performed in all planes G17 to G19 as for G18:

- Length 1 in X axis, length 2 in Z axis

Setting data in the program

In addition to setting of setting data via operator input, these can also be written in the program.

Programming example

```
N10 $MC_TOOL_LENGTH_TYPE=2
N20 $MC_TOOL_LENGTH_CONST=18
```

Reference

SINUMERIK 802D sl Function Manual for Turning, Milling, Nibbling; Special Treatments for Tool Compensations

9.7 Miscellaneous function M

Functionality

The miscellaneous function M initiates switching operations, such as "Coolant ON/OFF" and other functions.

Various M functions have already been assigned a fixed functionality by the CNC manufacturer. The functions not yet assigned fixed functions are reserved for free use of the machine manufacturer.

Note

An overview of the M miscellaneous functions used and reserved in the control system can be found in section "Overview of instructions".

Programming

```
M...                    ;Max. 5 M functions per block
```

Effect

Activation in blocks with axis movements:

If the functions **M0, M1, M2** are contained in a block with traversing movements of the axes, these M functions become effective **after the traversing movements.**

The functions M3, M4, M5 are output to the internal interface (PLC) before the traversing movements. The axis movements only begin once the controlled spindle has ramped up for M3, M4. For M5, however, the spindle standstill is not waited for. The axis movements already begin before the spindle stops (default setting).

The remaining M functions are output to the PLC with the traversing movements.

If you would like to program an M function directly before or after an axis movement, insert a separate block with this M function.

Note

The M function interrupts the G64 continuous path mode and generates exact stop:

Programming example

```
N10 S...
N20 X... M3                     ;M function in the block with axis movement, spindle
                                accelerates before the X axis movement
N180 M78 M67 M10 M12 M37        ;Max. 5 M functions in the block
```

Note

In addition to M and H functions, T, D, and S functions can also be transferred to the PLC (programmable logic controller). In all, a maximum of 10 such function outputs are possible in a block.

9.8 H function

Functionality

With H functions, floating point data (REAL data type - as with arithmetic parameters, see Section "Arithmetic Parameters R") can be transferred from the program to the PLC.

The meaning of the values for a given H function is defined by the machine manufacturer.

Programming

```
H0=... to H9999=...                ;Max. 3 H functions per block
```

Programming example

```
N10 H1=1.987 H2=978.123 H3=4      ;3 H functions in block
N20 G0 X71.3 H99=-8978.234        ;With axis movements in block
N30 H5                            ;Corresponds to H0=5.0
```

Note

In addition to M and H functions, T, D, and S functions can also be transferred to the PLC (programmable logic controller). In all, a maximum of 10 function outputs of this type are possible in a part program block.

9.9 Arithmetic parameters, LUD and PLC variables

9.9.1 Arithmetic parameter R

Functionality

The arithmetic parameters are used if an NC program is not only to be valid for values assigned once, or if you must calculate values. The required values can be set or calculated by the control system during program execution.

Another possibility consists of setting the arithmetic parameter values by operator inputs. If values have been assigned to the arithmetic parameters, they can be assigned to other variable-setting NC addresses in the program.

Programming

R0=... to R299=...	;Assign values to the arithmetic parameters
R[R0]=...	;Indirect programming: Assign a value to the arithmetic parameter R, whose number can be found, e.g. in R0
X=R0	;Assign arithmetic parameters to the NC addresses, e.g. for the X axis

Value assignments

You can assign values in the following range to the R parameters:

±(0.000 0001 ... 9999 9999)
(8 decimal places, arithmetic sign, and decimal point)

The decimal point can be omitted for integer values. A plus sign can always be omitted.

Example:

R0=3.5678 R1=-37.3 R2=2 R3=-7 R4=-45678.123

Use the **exponential notation** to assign an extended range of numbers:

± (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)

Range of values for 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 incl. assignments of arithmetic expressions.

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, arithmetic expressions and arithmetic parameters can be assigned to all addresses; **Exception: addresses N, G, and L**.

When assigning, write the " = " sign after the address character. It is also possible to have an assignment with a minus sign.

A separate block is required for assignments to axis addresses (traversing instructions).

Example:

```
N10 G0 X=R2           ;Assignment to X axis
```

Arithmetic operations/arithmetic functions

When operators/arithmetic functions are used, it is imperative to use conventional mathematical notation. Machining priorities are set using round brackets. Otherwise, multiplication and division take precedence over addition and subtraction.

Degrees are used for the trigonometrical functions.

Permitted arithmetic functions: see Section "List of instructions"

Programming example: Calculating with R parameters

```
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 and division take 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:
N70 R1= -R1                        ;The new R1 is the negative old R1
```

Programming example: Assign R parameters to the axes

```
N10 G1 G91 X=R1 Z=R2 F300          ;Separate blocks (traversing blocks)
N20 Z=R3
N30 X=-R4
N40 Z= SIN(25.3)-R5                ;With arithmetic operations
```

```
...
```

Programming example: Indirect programming

```
N10 R1=5                              ;Assigning R1 directly value 5 (integer)
...
N100 R[R1]=27.123                     ;Indirectly assign R5 the value 27.123
```

9.9.2 Local User Data (LUD)

Functionality

The operator/programmer (user) can define his/her own variable in the program from various data types (LUD = Local User Data). These variables are only available in the program in which they were defined. The definition takes place immediately at the start of the program and can also be associated with a value assignment at the same time. Otherwise the starting value is zero.

The name of a variable can be defined by the programmer. The naming is subject to the following rules:

- A maximum of 32 characters can be used.
- It is imperative to use letters for the first two characters; the remaining characters can be either letters, underscore or digits.
- Do not use a name already used in the control system (NC addresses, keywords, names of programs, subroutines, etc.).

Programming / data types

```
DEF BOOL varname1                     ;Boolean typ, values: TRUE (=1), FALSE (=0)
DEF CHAR varname2                     ;Char type, 1 ASCII code character: "a", "b", ...
                                      ;Numerical code value: 0 ... 255
DEF INT varname3                      ;Integer type, integer values, 32 bit value range:
                                      ;-2 147 483 648 through +2 147 483 647 (decimal)
DEF REAL varname4                     ;Real type, natural number (like arithmetic parameter
                                      R),
                                      ;Value range: ±(0.000 0001 ... 9999 9999)
                                      ;(8 decimal places, arithmetic sign and decimal point)
                                      or
                                      ;Exponential notation: ± (10 to power of -300 ... 10 to
                                      power of +300)
DEF STRING[string length]             ; STRING type, [string length]: Maximum number of
varname41                             characters
```

Each data type requires its own program line. However, several variables of the same type can be defined in one line.

Example:

```
DEF INT PVAR1, PVAR2, PVAR3=12, PVAR4      ;4 type INT variables
```

Example for STRING type with assignment:

```
DEF STRING[12] PVAR="Hello"                ; Define variable PVAR with a maximum of
                                           12 characters and assign string "Hello"
```

Fields

In addition to the individual variables, one or two-dimensional fields of variables of these data types can also be defined:

```
DEF INT PVAR5[n]                ;One-dimensional field, type INT, n: integer
DEF INT PVAR6[n,m]              ;Two-dimensional field, type INT, n, m: integer
```

Example:

```
DEF INT PVAR7[3]                ;Field with 3 elements of the type INT
```

Within the program, the individual field elements can be reached via the field index and can be treated like individual variables. The field index runs from 0 to a small number of the elements.

Example:

```
N10 PVAR7[2]=24                 ;The third field element (with index 2) is assigned
                                the value 24.
```

Value assignment for field with SET instruction:

```
N20 PVAR5[2]=SET(1,2,3)         ;After the 3rd field element, different values are
                                assigned.
```

Value assignment for field with REP instruction:

```
N20 PVAR7[4]=REP(2)             ;After field element [4] - all are assigned the same
                                value, here 2.
```

9.9.3 Reading and writing PLC variables

Functionality

To allow rapid data exchange between NC and PLC, a special data area exists in the PLC user interface with a length of 512 bytes. In this area, PLC data are compatible in data type and position offset. In the NC program, these compatible PLC variables can be read or written.

To this end, special system variables are provided:

$A_DBB[n]	;Data byte (8-bit value)
$A_DBW[n]	;Data word (16-bit value)
$A_DBD[n]	;Data double-word (32-bit value)
$A_DBR[n]	;REAL data (32-bit value)

"n" stands here for the position offset (start of data area to start of variable) in bytes

Programming example

```
R1=$A_DBR[5]             ;Reading a REAL value, offset 5 (starts at byte 5 of range)
```

Note

The reading of variables generates a preprocessing stop (internal STOPRE).

NOTICE

Writing of PLC tags is generally limited to a maximum of three tags (elements).

Where PLC tags are to be written in rapid succession, one element will be required per write operation.

If more write operations are to be executed than there are elements available, then block transfer will be required (a preprocessing stop may need to be triggered).

Example:

```
$A_DBB[1]=1 $A_DBB[2]=2 $A_DBB[3]=3
STOPRE
$A_DBB[4]=4
```

9.10 Program jumps

9.10.1 Jump destination for program jumps

Functionality

A **label** or a **block number** serve to mark blocks as jump destinations for program jumps. Program jumps can be used to branch to the program sequence.

Labels can be freely selected, but must contain a minimum of 2 and a maximum of 8 letters or numbers of which the **first two characters must be letters** or underscore characters.

Labels that are in the block that serves as the jump destination are **ended by a colon**. They are always at the start of a block. If a block number is also present, the label is located **after the block number**.

Labels must be unique within a program.

Programming example

```
N10 LABEL1: G1 X20              ;LABEL1 is the label, jump destination
...
TR789: G0 X10 Z20               ;TR789 is the label, jump destination
                                - No block number existing
N100 ...                        ;Block number can be jump target
...
```

9.10.2 Unconditional program jumps

Functionality

NC programs process their blocks in the sequence in which they were arranged when they were written.

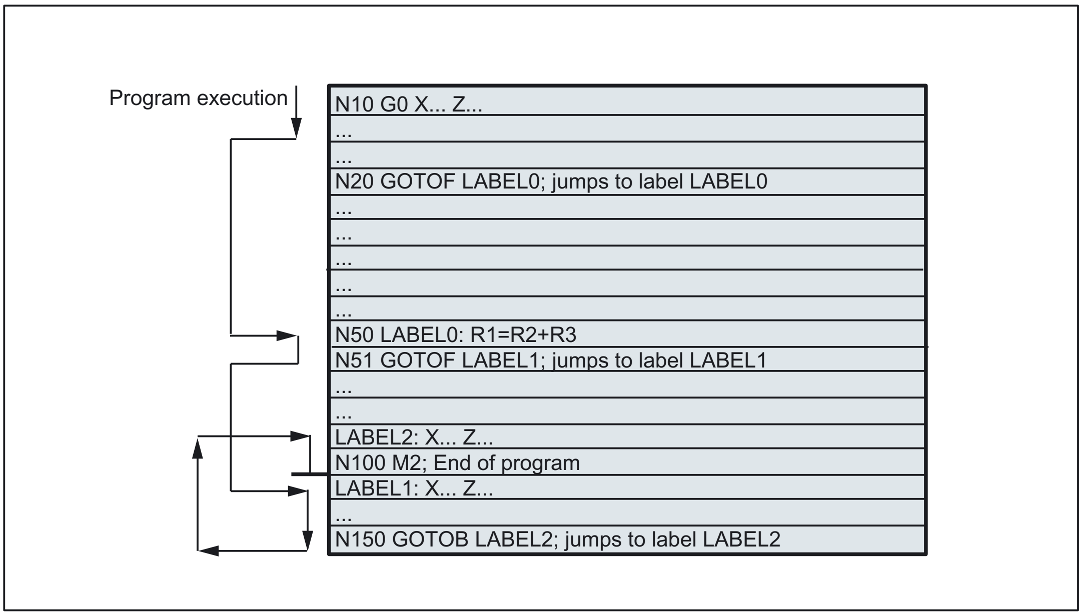
The processing sequence can be changed by introducing program jumps.

The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

The unconditional jump instruction requires a separate block.

Programming

GOTOF label	;Jump forward (in the direction of the last block of the program)
GOTOB label	;Jump backwards (in the direction of the first block of the program)
Label	;Selected string for the label (jump label) or block number



```
N10 G0 X... Z...
...
...
N20 GOTOF LABEL0; jumps to label LABEL0
...
...
...
...
...
N50 LABEL0: R1=R2+R3
N51 GOTOF LABEL1; jumps to label LABEL1
...
...
LABEL2: X... Z...
N100 M2; End of program
LABEL1: X... Z...
...
N150 GOTOB LABEL2; jumps to label LABEL2
```

Figure 9-59 Unconditional jumps using an example

9.10.3 Conditional program jumps

Functionality

Jump conditions are formulated after the **IF instruction**. If the jump condition (**value not zero**) is satisfied, the jump takes place.

The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

Conditional jump instructions require a separate block. Several conditional jump instructions can be located in the same block.

By using conditional program jumps, you can also considerably shorten the program, if necessary.

Programming

IF condition GOTOF label	;Jump forward
IF condition GOTOB label	;Jump backwards
GOTOF	;Jump direction forward (in the direction of the last block of the program)
GOTOB	;Jump direction backwards (in the direction of the first block of the program)
Label	;Selected string for the label (jump label) or block number
IF	;Introduction of the jump condition
Condition	;Arithmetic parameter, arithmetic expression for formulating the condition

Comparison operations

Operators	Meaning
= =	Equal to
< >	Not equal to
>	greater than
<	less than
> =	greater than or equal to
< =	less than or equal to

The comparison operations support formulating of a jump condition. Arithmetic expressions can also be compared.

The result of comparison operations is "satisfied" or "not satisfied." "Not satisfied" sets the value to zero.

Programming example for comparison operators

```
R1>1                        ;R1 greater than 1
1 < R1                      ;1 less than R1
R1<R2+R3                    ;R1 less than R2 plus R3
R6>=SIN( R7*R7)             ; R6 greater than or equal to SIN (R7) squared
```

Programming example

```
N10 IF R1 GOTOF LABEL1                ;If R1 is not null then go to the block
                                      having LABEL1
...
N90 LABEL1: ...
N100 IF R1>1 GOTOF LABEL2             ;If R1 is greater than 1 then go to the
                                      block having LABEL2
...
N150 LABEL2: ...
...
N800 LABEL3: ...
...
N1000 IF R45==R7+1 GOTOB LABEL3       ;If R45 is equal to R7 plus 1 then go to the
                                      block having LABEL3
...
Several conditional jumps in the
block:
N10 MA1: ...
...
N20 IF R1==1 GOTOB MA1 IF R1==2 GOTOF MA2 ...
...
N50 MA2: ...
```

Note

The jump is executed for the first fulfilled condition.

9.10.4 Program example for jumps

Task

Approaching points on a circle segment:

Existing conditions:

Start angle: 30° in R1

Circle radius: 32 mm in R2

Position spacing: 10° in R3

Number of points: 11 in R4

Position of circle center in Z: 50 mm in R5

Position of circle center in X: 20 mm in R6

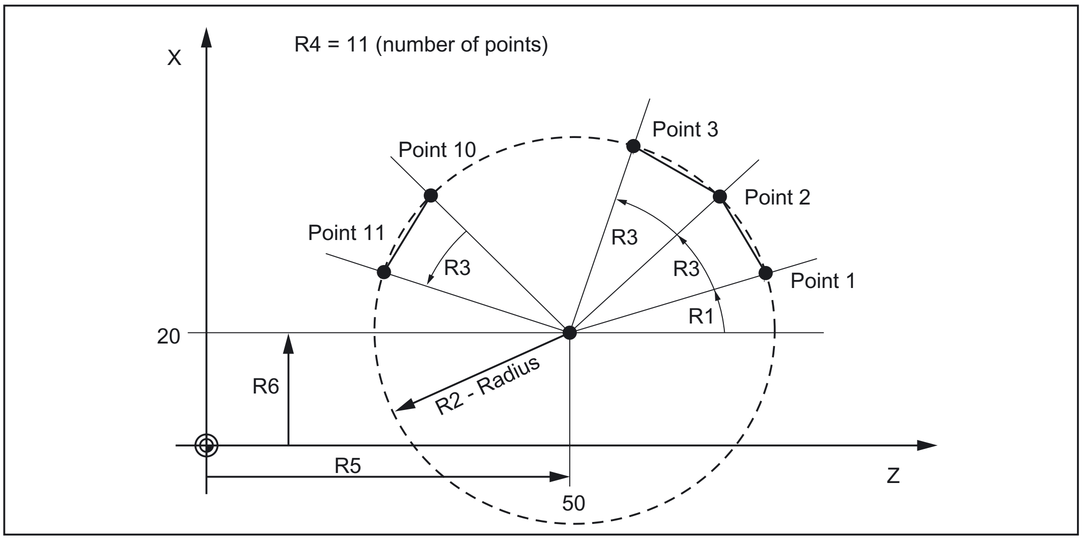


Figure 9-60 Linear approach of points on a circle segment

Programming example

```
N10 R1=30 R2=32 R3=10 R4=11 R5=50 R6=20    ;Assignment of initial values
N20 MA1: G0 Z=R2*COS (R1)+R5               ;Calculation and assignment to axis
X=R2*SIN(R1)+R6                            addresses
N30 R1=R1+R3 R4= R4-1
N40 IF R4 > 0 GOTOB MA1
N50 M2
```

Explanation

In block N10, the starting conditions are assigned to the corresponding arithmetic parameters. The calculation of the coordinates in X and Z and the processing takes place in N20.

In block N30, R1 is incremented by the clearance angle R3, and R4 is decremented by 1.

If R4 > 0, N20 is executed again; otherwise, N50 with End of program.

9.11 Subroutine technique

9.11.1 General information

Usage

Basically, there is no difference between a main program and a subroutine.

Frequently recurring machining sequences are stored in subroutines, e.g certain contour shapes. These subroutines are called at the appropriate locations in the main program and then executed.

One form of a subroutine is the **machining cycle**. Machining cycles contain universally valid machining scenarios. By assigning values via included transfer parameters, you can adapt the subroutine to your specific application.

Layout

The structure of a subroutine is identical to that of a main program (see Section "Program structure"). Like main programs, subroutines contain **M2 - end of program** in the last block of the program sequence. This means a return to the program level where the subroutine was called from.

End of program

The end instruction **RET** can also be used instead of the M2 program end in the subroutine.

RET must be programmed in a separate block.

The RET instruction is used when G64 continuous-path mode is not to be interrupted by a return. With M2, G64 is interrupted and exact stop is initiated.

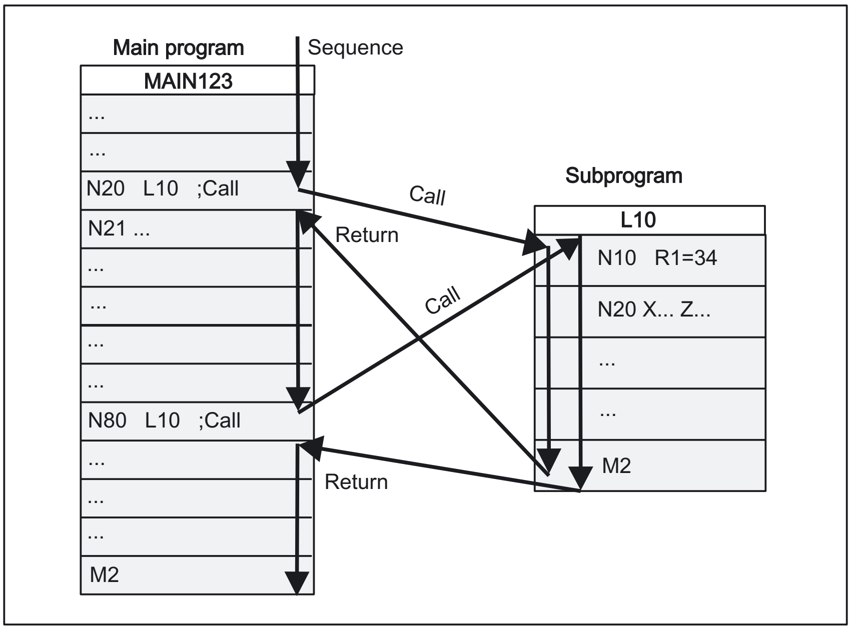


Figure 9-61 Example of a sequence when a subroutine is called in a two-channel manner.

Subroutine name

The subprogram is given a unique name allowing it to be selected from several subroutines. When you create the program, the program name may be freely selected provided the following conventions are observed:

The same rules apply as for the names of main programs.

Example: BUCHSE7

It is also possible to use the address word **L...** in subroutines. The value can have 7 decimal places (integers only).

Please observe: With address L, leading zeros are meaningful for differentiation.

Example: L128 ist nicht L0128 oder L00128 !
Dies sind 3 verschiedene Unterprogramme.

Note: The subroutine name LL6 is reserved for tool change.

Subroutine call

Subroutines are called in a program (main or subprogram) with their names. To do this, a separate block is required.
Example:

```
N10 L785                    ; Subprogram call L785
N20 SHAFT7                  ; Subprogram call SHAFT7
```

Program repetition P...

If a subroutine is to be executed several times in succession, write the number of times it is to be executed in the block of the call after the subroutine name under the **address P**. A maximum of **9,999 cycles** are possible (P1 ... P9999).
Example:

```
N10 L785 P3                 ; Subprogram call L785, 3 cycles
```

Nesting depth

Subroutines can also be called from a subroutine, not only from a main program. In total, up to **8 program levels** are available for this type of nested call, including the main program level.

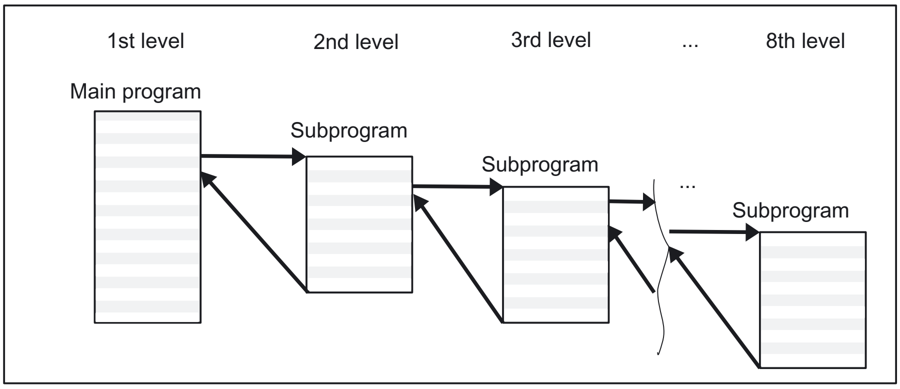


Figure 9-62 Execution with 8 program levels

Information

Modal G functions can be changed in the subroutine, e.g. G90 -> G91. When returning to the calling program, ensure that all modal functions are set the way you need them to be.

Please make sure that the values of your arithmetic parameters used in upper program levels are not inadvertently changed in lower program levels.

When working with SIEMENS cycles, up to 7 program levels are needed.

9.11.2 Calling machining cycles (turning)

Functionality

Cycle are technology subroutines that realize a certain machining process, Adaptation to the particular problem is performaed directly via supply parameters/values when calling the respective cycle.

Programming example

```
N10 CYCLE83(110, 90, ...)          ;Call of cycle 83, transfer values directly,
                                   separate block
…
N40 RTP=100 RFP= 95.5 ...          ;Set transfer parameters for cycle 82
N50 CYCLE82(RTP, RFP, ...)         ;Call of cycle 82, separate block
```

9.11.3 Execute external subroutine (EXTCALL)

Function

With SINUMERIK 802D sl pro, it is possible to reload and execute programs with the `EXTCALL` command via the following external data carriers:

- Customer CompactFlash card (drive D)
- USB FlashDrive (drive G)
- Ethernet to the PG/PC (from drive H)

Machine data

The following machine data is used for the `EXTCALL` command:

- MD10132 $MN_MMC_CMD_TIMEOUT

  Monitoring time for the command in part program
- MD18362 $MN_MM_EXT_PROG_NUM

  Number of program levels that can be processed simultaneously from external
- SD42700 $SC_EXT_PROGRAM_PATH

  Program path for external subroutine call

NOTICE
When using SD42700 $SC_EXT_PROGRAM_PATH, all subprograms called with EXCALL are searched under this path. No drives can be specified in the EXTCALL call.

Programming with path specification in SD42700 EXT_PROGRAM_PATH

`EXTCALL` ("<program name>")

Parameter

EXTCALL	; Keyword for subroutine call
<program name>	; Constant/variable of STRING type
Example:	
EXTCALL ("RECTANGULAR POCKET")	

Programming without path specification in SD42700 EXT_PROGRAM_PATH

`EXTCALL` ("<path\program name>")

Parameter

EXTCALL	; Keyword for subroutine call
<Path\program name>	; Constant/variable of STRING type
Example:	
EXTCALL ("D:\EXTERNE_UP\RECHTECKTASCHE")	

Note

External subroutines must not contain jump statements such as `GOTOF`, `GOTOB`, `CASE`, `FOR`, `LOOP`, `WHILE`, or `REPEAT`.

`IF-ELSE-ENDIF` constructions are possible.

Subroutine calls and nested `EXTCALL` calls may be used.

RESET, POWER ON

RESET and POWER ON cause external subroutine calls to be interrupted and the associated load memory to be erased.

Examples

1. Processing of external customer CompactFlash Card or USB FlashDrive

System: SINUMERIK 802D sl pro

The "Main.mpf" main program is stored in NC memory and is selected for execution:

```
N010 PROC MAIN
N020 ...
N030 EXTCALL ("D:\EXTERNE_UP\BOHRUNG")
N040 ...
N050 M30
```

The "BOHRUNG.SPF" subprogram to be reloaded is located on the customer CompactFlash Card.

```
N010 PROC MAIN
N020 ...
N030 EXTCALL ("G:\EXTERNE_UP\BOHRUNG ")
N040 ...
N050 M30
```

The "BOHRUNG.SPF" subprogram to be reloaded is located on the USB-Flash Drive.

```
N010 PROC BOHRUNG
N020 G1 F1000
N030 X= ... Z= ...
N040 ...
...
...
N999999 M17
```

External program memory

The external program memory can be stored on the following data media:

- Customer CompactFlash card (drive D)
- USB FlashDrive (drive G)
- Using Ethernet to the PG/PC (see "Connect and separate network drive")

Note

Execution from external source via V24 interface

With SINUMERIK 802D sl pro, the "Execution from external source" softkey can be used to transfer external programs across the RS-232 interface onto the NC.

Adjustable reload memory (FIFO buffer)

A reload memory is required in the NCK in order to run a program in "Execution from external source" mode (main program or subroutine). The size of the reload memory is preset to 30 Kbytes and, like all other memory-related machine data, can only be changed to match requirements by the machine manufacturer.

One reload memory must be set for each program (main program or subroutine) to run concurrently in "Execution from external source" mode.

Machine manufacturer

Please contact the machine manufacturer if the size and number of reload memories is to be extended.

9.12 Timers and workpiece counters

9.12.1 Runtime timer

Functionality

The timers are prepared as system variables ($A...) that can be used for monitoring the technological processes in the program or only in the display.

These timers are read-only. There are timers that are always active. Others can be deactivated via machine data.

Timers - always active

- $AN_SETUP_TIME

  Time since the last control powerup with default values (in minutes)

  It is automatically reset in the case of a "Control power-up with default values".
- $AN_POWERON_TIME

  Time since the last control powerup (in minutes)

  It is reset to zero automatically with each power-up of the control system.

Timers that can be deactivated

The following timers are activated via machine data (default setting).

The start is timer-specific. Each active run-time measurement is automatically interrupted in the stopped program state or for feedrate-override-zero.

The behavior of the activated timers for active dry run feedrate and program testing can be specified using machine data.

- **$AC_OPERATING_TIME**

  Total execution time in seconds of NC programs in the automatic mode

  In the AUTOMATIC mode, the runtimes of all programs between NC START and end of program / RESET are summed up. The timer is zeroed with each power-up of the control system.

- **$AC_CYCLE_TIME**

  Runtime of the selected NC program (in seconds)

  The runtime between NC Start and End of program / Reset is measured in the selected NC program. The timer is reset with the start of a new NC program.

- **$AC_CUTTING_TIME**

  Tool action time (in seconds)

  The runtime of the path axes is measured in all NC programs between NC START and end of program / RESET without rapid traverse active and with the tool active (default setting).

  The measurement is interrupted when a dwell time is active.

  The timer is automatically set to zero with each power-up of the control system.

Programming example

```
N10 IF $AC_CUTTING_TIME>=R10 GOTOF WZZEIT            ; Tool operation time limit
                                                      value?
...
N80 WZZEIT:
N90 MSG("Tool action time: Limit value reached")
N100 M0
```

Display

The contents of the active system variables are visible on the screen under
<OFFSET PARAM> -> "Setting data" ">" "Times/counters":

Total run time = $AC_OPERATING_TIME

Program run time = $AC_CYCLE_TIME

Feedrate run time = $AC_CUTTING_TIME

Time since cold restart = $AN_SETUP_TIME

Time since warm restart= $AN_POWERON_TIME

"Program run time" is also visible in the AUTOMATIC mode in the "Position" operating area in the information line.

9.12.2 Workpiece counter

Functionality

The "Workpiece counter" function provides counters for counting workpieces.

These counters exist as system variables with write and read access from the program or via operator input (observe the protection level for writing!).

Machine data can be used to control counter activation, counter reset timing and the counting algorithm.

Counters

- $AC_REQUIRED_PARTS

  Number of workpieces required (workpiece setpoint)

  In this counter you can define the number of workpieces at which the actual workpiece counter $AC_ACTUAL_PARTS is reset to zero.

  The generation of the display alarm 21800 "Workpiece setpoint reached" can be activated via machine data.
- $AC_TOTAL_PARTS

  Total number of workpieces produced (total actual)

  The counter specifies the total number of all workpieces produced since the start time.

  The counter is automatically set to zero upon every booting of the control system.
- $AC_ACTUAL_PARTS

  Number of actual workpieces (actual)

  This counter registers the number of all workpieces produced since the starting time. When the workpiece setpoint is reached ($AC_REQUIRED_PARTS, value greater than zero), the counter is automatically zeroed.
- $AC_SPECIAL_PARTS

  Number of workpieces specified by the user

  This counter allows users to make a workpiece counting in accordance with their own definition. Alarm output can be defined for the case of identity with $AC_REQUIRED_PARTS (workpiece target). Users must reset the counter themselves.

Programming example

```
N10 IF $AC_TOTAL_PARTS==R15 GOTOF SIST              ; Count reached?
...
N80 SIST:
N90 MSG("Workpiece setpoint reached")
N100 M0
```

Display

The contents of the active system variables are visible on the screen under <OFFSET PARAM> -> "Setting data" ">" "Times/counters":

Total parts= $AC_TOTAL_PARTS

Required parts= $AC_REQUIRED_PARTS

Number of parts =$AC_ACTUAL_PARTS, $AC_SPECIAL_PARTS not available for display

"Number of parts" is also visible in the AUTOMATIC mode in the "Position" operating area in the information line.

9.13 Language commands for tool monitoring

9.13.1 Tool monitoring overview

Functionality

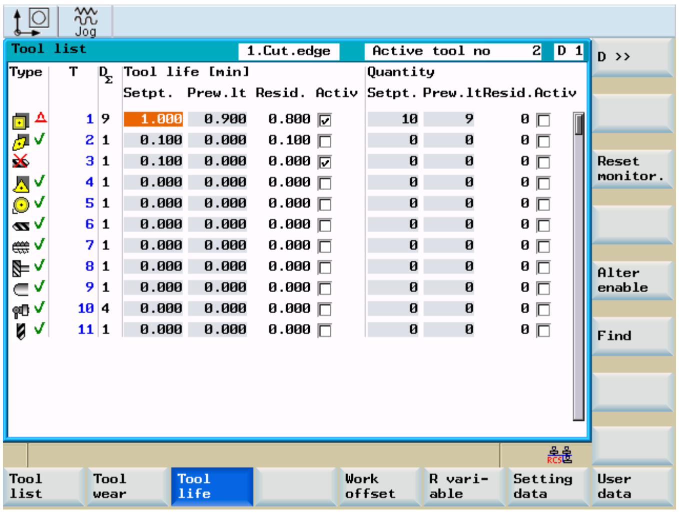
This function is available for SINUMERIK 802D sl plus and 802D sl pro.

The tool monitoring is activated via machine data.

OFFSET PARAM

Tool monitoring

Monitoring is carried out in the <OFFSET PARAM> > "Tool monitoring" operating area.



Tool list | 1.Cut.edge | Active tool no 2 | D 1

Type	T	DΣ	Tool life [min]				Quantity			
			Setpt.	Prew.lt	Resid.	Activ	Setpt.	Prew.lt	Resid.	Activ
△	1	9	1.000	0.900	0.800	☑	10	9	0	☐
✓	2	1	0.100	0.000	0.100	☐	0	0	0	☐
✗	3	1	0.100	0.000	0.000	☑	0	0	0	☐
✓	4	1	0.000	0.000	0.000	☐	0	0	0	☐
✓	5	1	0.000	0.000	0.000	☐	0	0	0	☐
✓	6	1	0.000	0.000	0.000	☐	0	0	0	☐
✓	7	1	0.000	0.000	0.000	☐	0	0	0	☐
✓	8	1	0.000	0.000	0.000	☐	0	0	0	☐
✓	9	1	0.000	0.000	0.000	☐	0	0	0	☐
✓	10	4	0.000	0.000	0.000	☐	0	0	0	☐
✓	11	1	0.000	0.000	0.000	☐	0	0	0	☐

Figure 9-63 Tool monitoring

You can monitor tool wear by observing tool life and/or the number of workpieces. If a tool reaches its wear limit, an advance warning is automatically output and the tool is blocked so that it can no longer be used for machining.

Note

For machines, tools and technical equipment, the lifetime is the time in which these can be used up until the next maintenance, cleaning or similar operation must be carried out; i.e. the time in which the machine, system or the tool can operate without any interruption.

You can specify the following data in the tool monitoring:

- Tool life, specified as a setpoint, and an advance warning limit for tool monitoring. The time remaining before the tool is blocked is calculated and displayed.
- Number of workpieces, specified as a setpoint, and advance warning limit for tool monitoring. The number of workpieces remaining before the tool is blocked is calculated and displayed.
- Tool monitoring can be activated for tool life or number of workpieces.
  - Monitoring of the **tool life**

    By activating service life monitoring, the service life during the action time of the tool (G1, G2, G3) is monitored.
  - Monitoring of the **workpiece count**

    By activating workpiece count monitoring, the workpiece count is monitored via the program command SETPIECE( ) at the end of the part program.

The above-mentioned types of monitoring can be activated for one tool (T) simultaneously.

The control / data input for the tool monitoring is preferably made using done by the operator. In addition, various functions can be programmed.

Monitoring counter

Monitoring counters exist for each monitoring type. The monitoring counters count from a set value >0 down to zero. When a counter has decremented to a value of ≤0, the limit value is reached. A corresponding alarm message is issued.

System variable for type and condition of the monitoring

- **$TC_TP8[t]**

  ; Status of the tool with the number t:

  - Bit 0

    =1: Tool is **active**

    =0: Tool is not active
  - Bit 1

    =1: Tool is **enabled**

    =0: Not released
  - Bit 2

    =1: Tool is **disabled**

    =0: Not blocked
  - Bit 3: Reserved
  - Bit 4

    =1: **Prewarning limit reached**

    =0: not reached
- **$TC_TP9[t]**

  ; Type of monitoring function for the tool with number t :

  - = 0: No monitoring
  - = 1: (Tool) life-monitored tool
  - = 2: Quantity-monitored tool

These system variables can be read/written in the NC program.

System variables for tool monitoring data

Table 9- 6 Tool monitoring data

Name	Description	Data type	Default
$TC_MOP1[t,d]	Warning limit for tool life in minutes	REAL	0.0
$TC_MOP2[t,d]	Residual tool life in minutes	REAL	0.0
$TC_MOP3[t,d]	Warning limit for count	INT	0
$TC_MOP4[t,d]	Residual unit quantity	INT	0
...	...		
$TC_MOP11[t,d]	Tool life setpoint	REAL	0.0
$TC_MOP13[t,d]	Unit quantity setpoint	INT	0
t for tool number T, d for D number			

System variables for active tool

The following can be read in the NC program via system variables:

- $P_TOOLNO - number of the active tool T
- $P_TOOL - active D number of the active tool

9.13.2 Tool life monitoring

Tool life monitoring is done for the tool cutting edge that is currently in use (active cutting edge D of the active tool T).

As soon as the path axes traverse (G1, G2, G3, ... but not for G0), the residual tool life ($TC_MOP2[t,d]) of this tool cutting edge is updated. If the residual tool life of a tool's cutting edge runs below the value of "Prewarning limit for tool life" ($TC_MOP1[t,d]), it is reported via an interface signal to the PLC.

If the residual service life = 0, an NCK alarm is output. The tool changes to the "disabled" condition and cannot be programmed again until this condition changes. The operator must intervene: The operator must change the tool or ensure that he/she has an operational tool for machining.

$A_MONIFACT system variable

The **$A_MONIFACT** system variable (REAL data type) allows the monitoring clock to be run slower or faster. This factor can be set before using the tool, in order to take the different kinds of wear into consideration according to the workpiece material used, for example.

After control system power-up, Reset / End of program, the $A_MONIFACT factor has the value 1.0. Real time is in effect.

Examples for accounting:

$A_MONIFACT=1: 1 minute real time = 1 minute tool life which is decremented

$A_MONIFACT=0.1: 1 minute real time = 0.1 minute tool life which is decremented

$A_MONIFACT=5: 1 minute real time = 5 minute tool life which is decremented

Setpoint update with RESETMON( )

The RESETMON(state, t, d, mon) function sets the actual value to the setpoint:

- For all cutting edges or only for a specific cutting edge of a specific tool
- For all monitoring types or only for a specific monitoring type.

Transfer parameters:

- INT

  state: Status of the command execution:

  = 0: Successful execution

  = -1: The cutting edge with the specified D number d does not exist.

  = -2: The tool with the specified T number t does not exist.

  = -3: The specified tool t does not have a defined monitoring function.

  = -4: The monitoring function is not activated, i.e. the command is not executed.
- INT

  t: Internal T number:

  = 0: for all tools

  > 0: for this tool
- INT

  d: *Optional:*D number of the tool with the number t:

  > 0: for this D number

  without d/= 0: all cutting of the tool t
- INT

  mon: *Optional:* bit-encoded parameters for the monitoring type (values like $TC_TP9):

  = 1: Tool life

  = 2: Quantity

  without monitoring or = 0: All actual values of the monitoring functions active for tool t are set to their setpoints.

Note

RESETMON

- RESETMON( ) has no effect when "Program test" is active.
- The variable for the state feedback must be defined at the beginning of the program using a DEF statement: DEF INT state

  You can also define a different name for the variable (instead of state, with a maximum of 15 characters, beginning with 2 letters). The variable is only available in the program in which it was defined.

  This also applies to the mon. monitoring type variable. As far as a specification is required at all, it can be directly transferred as a number (1 or 2).

9.13.3 Workpiece count monitoring

Function

The workpiece count of the active cutting edge of the active tool is monitored.
The workpiece count monitoring records all the tool cutting edges that are used to produce a workpiece. If the count is changed by new parameters, the monitoring data are adapted to all of the tool cutting edges that became active since the last unit count.

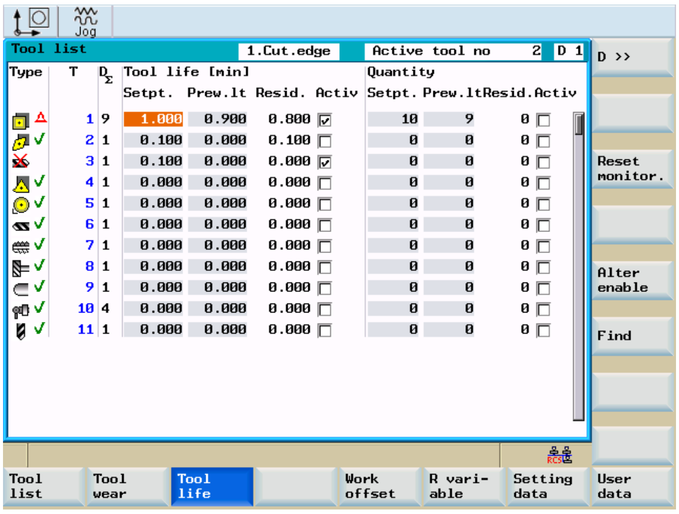
Updating the workpiece count via HMI operation

OFFSET PARAM

Tool monitoring

In the <OFFSET PARAM> > "Tool monitoring" operating area, the "workpiece count" is entered as the "setpoint" and "prewarning limit" for tool monitoring.

When executing the SETPIECE ( ) language command, the remaining number of workpiece to the last call before blocking the tool is calculated and displayed.



Type	T	D Σ	Tool life [min]				Quantity			
			Setpt.	Prew.lt	Resid.	Activ	Setpt.	Prew.lt	Resid.	Activ
△	1	9	1.000	0.900	0.800	☑	10	9	0	□
✓	2	1	0.100	0.000	0.100	□	0	0	0	□
	3	1	0.100	0.000	0.000	☑	0	0	0	□
✓	4	1	0.000	0.000	0.000	□	0	0	0	□
✓	5	1	0.000	0.000	0.000	□	0	0	0	□
✓	6	1	0.000	0.000	0.000	□	0	0	0	□
✓	7	1	0.000	0.000	0.000	□	0	0	0	□
✓	8	1	0.000	0.000	0.000	□	0	0	0	□
✓	9	1	0.000	0.000	0.000	□	0	0	0	□
✓	10	4	0.000	0.000	0.000	□	0	0	0	□
✓	11	1	0.000	0.000	0.000	□	0	0	0	□

Figure 9-64 Tool monitoring

SETPIECE - decrementing the workpiece counter

With the **SETPIECE**function, the user can update the workpiece count monitoring data of the tools that were involved in the machining process.

Each tool that has been loaded since the last activation of SETPIECE is recorded.

As a rule, the function is used for programming at the end of the NC part program. The workpiece count of all tools that were involved in the workpiece count monitoring is decremented by a specified amount.

Note

The SETPIECE( ) command is not active during the block search.

Direct writing of $TC_MOP4[t,d] is recommended only in simple cases. A subsequent block with the STOPRE command is required.

The SETPIECE ( ) command is also active for the tool or edge selected prior to program start. Change the tool in the "MDA" mode and the SETPIECE ( ) command will be active for the tools after program start.

Programming

SETPIECE(n, s)	; n : = 0... 32000 Number of workpieces that have been produced since the last execution of the SETPIECE function. The counter status for the remaining part quantity ($TC_MOP4[t,d]) is reduced by this value. ; s : = 1 or 2 spindles 1 or 2 (tool holder), only required if 2 spindles are present

Programming example

```
N10 G0 X100
N20 ...
N30 T1                        ;Tool change with T command
N50 D1
...                           ;Machining with T1, D1
N90 SETPIECE(2)               ;$TC_MOP4[1,1 ] (T1,D1) is decremented by 2
N100 T2
N110 D2
...                           ;Machining with T2, D2
N200 SETPIECE(1)              ;$TC_MOP4[2,2 ] (T2,D2) is decremented by 1
...
N300 M2
```

Examples of SETPIECE with tool change command M06

The tools involved in a workpiece (program) should be decremented by a value of 1.

```
T1                             ; T1 is preselected (with regard to main spindle)
M06                            ; T1 is changed
D1                             ; D1 is activated
T2                             ; T2 is preselected
.                              ; Machining program
.
M06                            ; T2 is changed
D1                             ; D1 from T2 is activated
T3                             ; T3 is preselected
.                              ; Machining program
.
M06
T0                             ; Preparation for clearing the spindle
.
.
M06                            ; Clearing the spindle
SETPIECE(1)                    ; SETPIECE on all tools
M2
```

The counter is to be decremented for each tool

In this example, tools T1, T2 and T3 are to machine a program.

All three tools are monitored for workpiece count. The aim is to decrement tool T1 by the value 1, T2 by the value 2 and not to decrement T3.

```
N500 T1
N600 M06                       ; Tool change
N700 D1                        ; By selecting the compensation, the new tool is
                               loaded into the SETPIECE memory
N900 T2                        ; Preparing the next tool
.                              ; Machining commands
.
N1000 SETPIECE (1)             ; SETPIECE acts on T1, SETPIECE memory is cleared
N1100 M06                      ; Tool change
N1200 D1
N1400 T3                       ; Preparing the next tool
.                              ; Machining commands
.
N1500 SETPIECE (2)             ; Only active on T2
N1600 M06                      ; Tool change
N1700 D1
.                              ; Machining commands
```

```
.
N1800 SETPIECE (0)            ; Only active on T3, not decremented
N1900 T0
N2000 M06
N2100 D0
N2300 M2
```

Setpoint update

The setpoint is updated via the HMI.

However, the setpoint can also be updated via the RESETMON (state, t, d, mon) function.

During the setpoint update, the residual workpiece counter ($TC_MOP4[t,d]) is internally set to the workpiece setpoint ($TC_MOP13[t,d]).

Example:

```
DEF INT state                         ;Define the variable for status feedback at
                                      the beginning of the program
...
N100 RESETMON(state,12,1,2)           Setpoint update of the workpiece counter for
                                      T12, D1, setpoint 2
...
```

Programming example

```
DEF INT state                         ;Define the variable for RESETMON() status
                                      feedback
…
G0 X...                               ;Traverse freely
T7                                    ;New tool; if necessary, use M6 for loading
$TC_MOP3[$P_TOOLNO,$P_TOOL]=100       ;100 pieces pre-warning limit
$TC_MOP4[$P_TOOLNO,$P_TOOL]=700       ;Residual unit quantity
$TC_MOP13[$P_TOOLNO,$P_TOOL]=700      -Setpoint for workpiece count
;Activation after setting:
$TC_TP9[$P_TOOLNO,$P_TOOL]=2          ;Activation of workpiece count monitoring;
                                      active tool
STOPRE
ANF:
MACHINING                             ;Subroutine for workpiece machining
SETPIECE(1)                           ,Update counter
M0                                    ;Next workpiece; press NC START to continue
IF ($TC_MOP4[$P_TOOLNO,$P_TOOL]]>1) GOTOB ANF
MSG("Tool T7 worn - Please replace")
M0                                    ;After tool change, press NC START to
                                      continue
RESETMON(state,7,1,2)                 ;Setpoint update of workpiece counter
```

```
IF (state<>0) GOTOF ALARM
GOTOB START
ALARM:                                  ;Display errors:
MSG("error RESETMON: " <<state)
M0
M2
```

9.14 Milling on turning machines

9.14.1 Milling of the front face - TRANSMIT

This function is available for SINUMERIK 802D sl plus and 802D sl pro.

Functionality

- The kinematic transformation function TRANSMIT allows milling / drilling on the front face of turned parts in the turning clamp.
- A Cartesian coordinate system is used to program these machining operations.
- The control transforms the programmed traversing movements of the Cartesian coordinate system into traversing movements of the real machine axes. The main spindle functions here as the machine rotary axis.
- TRANSMIT must be configured via special machine data elements. A tool center offset relative to the turning center is permitted and is also configured via these machine data elements.
- In addition to the tool length compensation, it is also possible to work with the tool radius compensation (G41, G42).
- The velocity control makes allowance for the limits defined for the rotations.

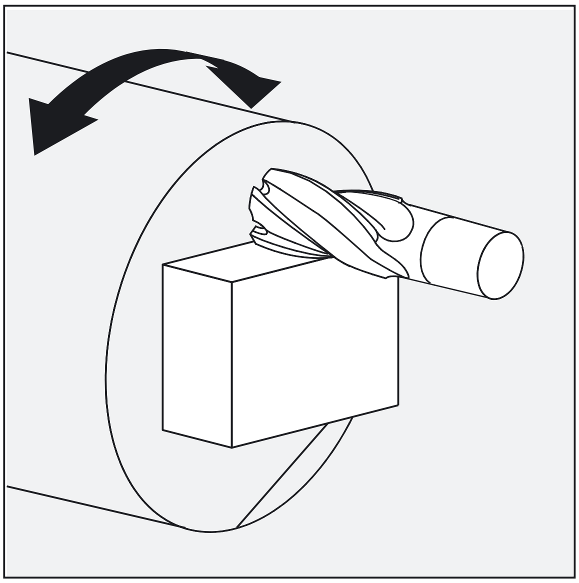


Figure 9-65 Milling of the front face

Programming

```
TRANSMIT                    ;Switch on TRANSMIT (separate block)
With TRAFOOF                ;Switch off (separate block)
```

TRAFOOF deactivates any active transformation function.

Programming example

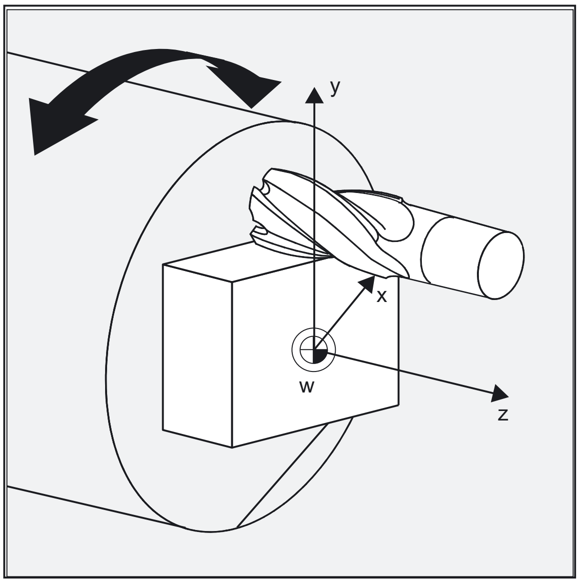


Figure 9-66 Cartesian coordinate system X, Y, Z with origin at the turning center when programming TRANSMIT

```
;Mill square, off-center and rotated
N10 T1 F400 G94 G54          ;Milling tool, feedrate, type of feedrate
N20 G0 X50 Z60 SPOS=0        ; Approach the starting position
N25 SETMS(2)                 ; Master spindle is now the milling spindle
N30 TRANSMIT                 ; Activate TRANSMIT function
N35 G55 G17                  ;Work offset, activate X/Y plane
N40 ROT RPL=-45              ;Programmable rotation in X/Y plane
N50 ATRANS X-2 Y3            ;Programmable translation
N55 S600 M3                  ; Switch on milling spindle
N60 G1 X12 Y-10 G41          ;Switch on tool radius compensation
N65 Z-5                      ;Feed milling tool
N70 X-10
N80 Y10
N90 X10
N100 Y-12
N110 G0 Z40                  ; Raise milling tool
```

```
N120 X15 Y-15 G40          ;Switch off tool radius compensation
N130 TRANS                 ;Switch off programmable offset and rotation
N140 M5                    ;Switch off milling spindle
N150 TRAFOOF               ; Switch off TRANSMIT
N160 SETMS                 ; Master spindle is now the main spindle again
N170 G54 G18 G0 X50 Z60    ; Approach the starting position
SPOS=0
N200 M2
```

Information

The turning center with X0/Y0 is designated as the pole. Workpiece machining operations close to the pole are not recommended since these may require sharp feedrate reductions to prevent overloading of the rotary axis. Avoid selecting TRANSMIT when the tool is positioned exactly on the pole. Ensure that the path of the tool center point does not travel through the X0/Y0 pole.

9.14.2 Milling of the peripheral surface - TRACYL

This function is available for SINUMERIK 802D sl plus and 802D sl pro.

Functionality

- The kinematic transformation function TRACYL is used for milling machining of the peripheral surface of cylindrical objects and allows the production of grooves at any position.
- The path of the grooves is programmed in the **plane** peripheral surface, which was logically developed for a specific machining cylinder diameter.

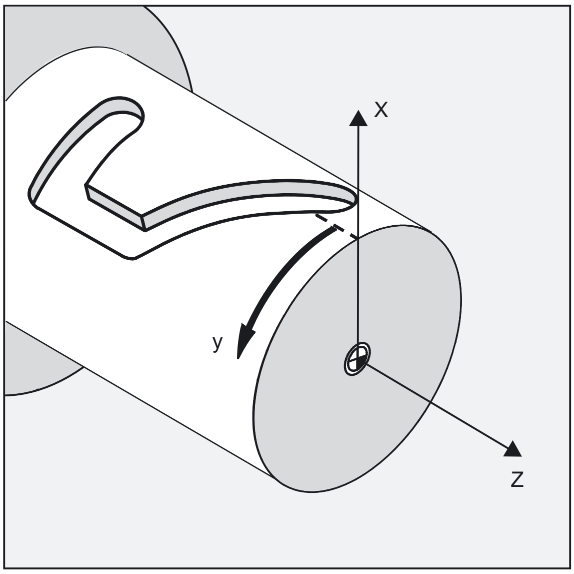


Figure 9-67 Cartesian coordinate system X, Y, Z when programming TRACYL

- The control transforms the programmed traversing movements in the Cartesian coordinate X, Y, Z system into traversing movements of the real machine axes. The main spindle functions here as the machine rotary axis.
- TRACYL must be configured using special machine data. The rotary axis position at which the value Y=0 is also defined here.
- If the machine has a real machine Y axis (YM), an expanded TRACYL variant can also be configured. This allows grooves with groove side compensation to be produced: The groove side and the groove base are perpendicular to each other - even if the milling tool diameter is smaller than the groove width. This is otherwise only possible with exact fitting milling cutters.

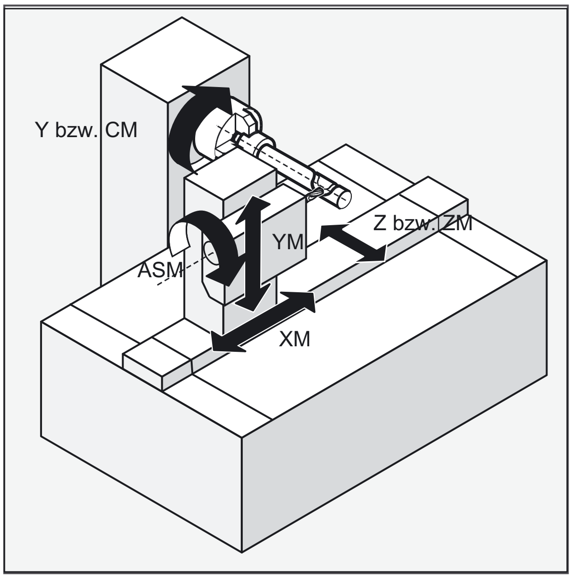


Figure 9-68 Special machine kinematics with additional machine Y axis (YM)

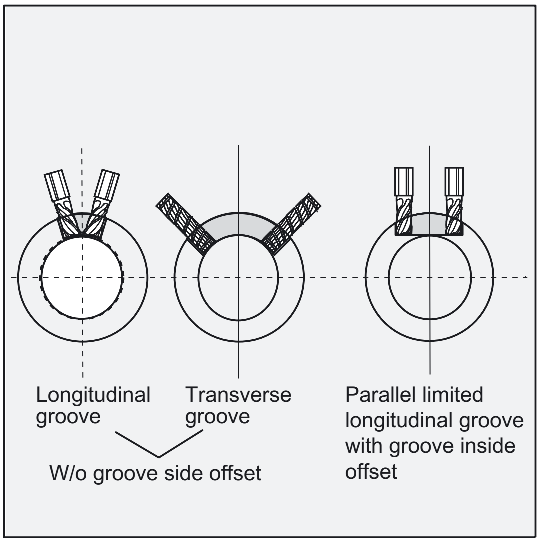

Longitudinal groove
Transverse groove
W/o groove side offset
Parallel limited longitudinal groove with groove inside offset

Figure 9-69 Various grooves in cross section

Programming

```
TRACYL(d)              ; Switch on TRACYL (separate block)
   TRAFOOF             ; Switch off (separate block)
                       ; d - Machining diameter of the cylinder in mm
```

TRAFOOF deactivates any active transformation function.

OFFN address

Distance from the groove side wall to the programmed path
The groove center line is generally programmed. OFFN defines the (half) groove width for activated milling cutter radius compensation (G41, G42).
Programming: OFFN=... ; distance in mm

Note:
Set OFFN=0 once the groove has been completed. OFFN is also used outside of TRACYL - for offset programming in combination with G41, G42.

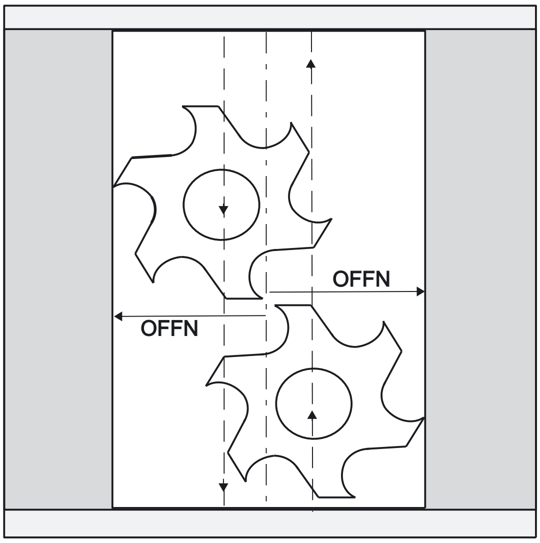


Figure 9-70 Use of OFFN for the groove width

Programming notes

In order to mill with TRACYL, the groove center line is programmed in the part program with the coordinates and the (half) groove width is programmed with OFFN.
OFFN only becomes effective with the tool radius compensation selected. Furthermore, OFFN must be >= tool radius to avoid damage to the opposite groove side.
Normally, a part program for groove milling includes the following steps:

1. Select the tool
2. Select TRACYL
3. Select the appropriate work offset
4. Positioning
5. Program OFFN
6. Select TRC

7. Approach block (travel in TRC and approach groove side)
8. Program the groove path through the groove center line
9. Deselect TRC
10. Retraction block (travel in TRC and retract from groove side)
11. Positioning
12. Delete OFFN
13.TRAFOOF (deselect TRACYL)
14. Select the original work offset again
(also see following programming example)

Information

- Guide grooves:
  By using a tool diameter that corresponds exactly to the groove width, it is possible to produce an exact groove. The tool radiuis compensation is not enabled in this case. With TRACYL, grooves can also be produced, whose tool diameter is smaller than the groove width. Here, it makes sense to use tool radius compensation (G41, G42) and OFFN.
  In order to avoid accuracy problems, the tool diameter should only slightly be smaller than the groove width.
- For TRACYL with groove side compensation, the axis (YM) used for the compensation should be located at the turning center. Thus, the groove is created centered on the programmed groove center line.
- Anwahl der Werkzeugradiuskorrektur (WRK) :
  Die WRK wirkt zur programmierten Nutmittenlinie. This produces the groove side. G42 is input so that the tool traverses to the left of the groove side (to the right of the groove center line). Accordingly, G41 is to be written to the right of the groove side (to the left of the groove center line).
  As an alternative to exchanging G41<->G42, you can input the groove width with a minus sign in OFFN.
- Since, even without TRACYL, OFFN is included when TRC is active, OFFN should be reset to zero after TRAFOOF. OFFN acts differently with TRACYL than it does without TRACYL.
- It is possible to change OFFN within a part program. This allows the actual groove center line to be offset from the center.

Reference

SINUMERIK 802D sl Function Manual for Turning, Milling, Nibbling; Kinematic Transformations

Programming example

Making a hook-shaped groove

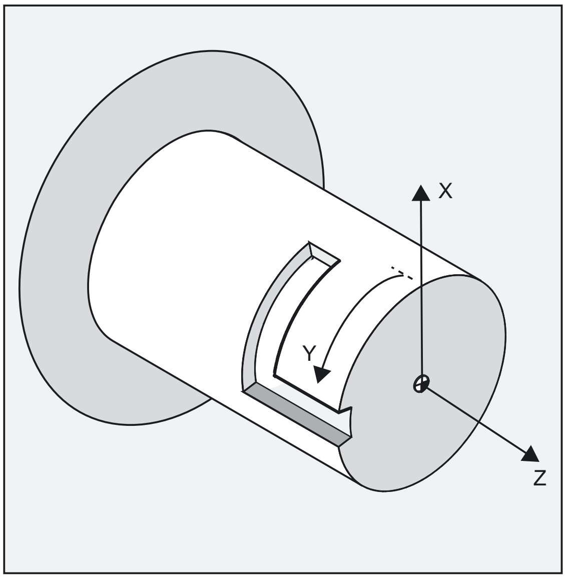


Figure 9-71 Producing a groove (example)

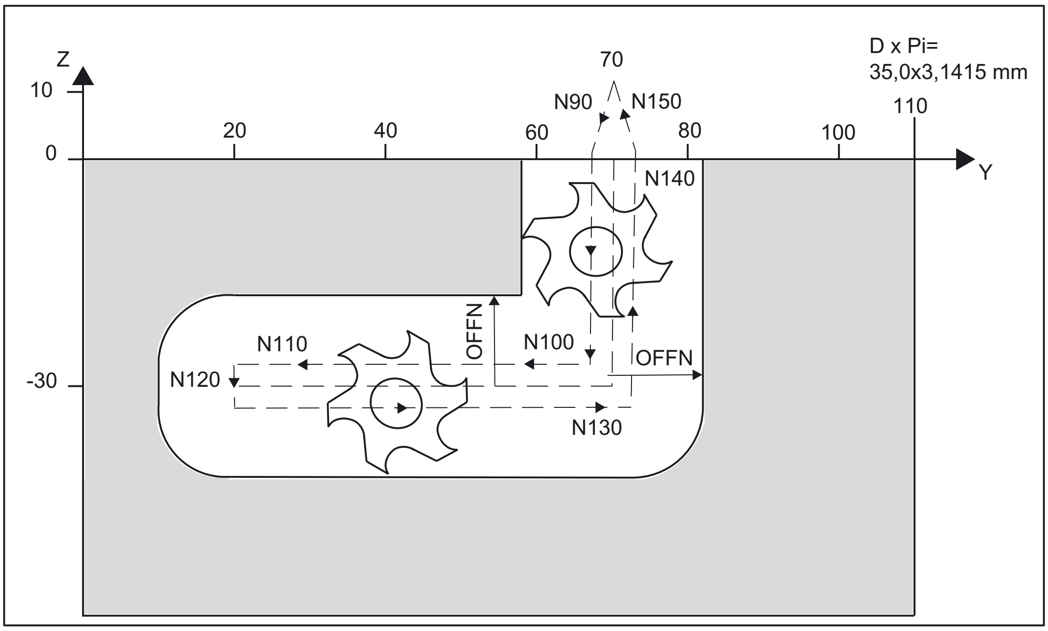


Figure 9-72 Programming the groove, values at the groove base

;Machining diameter of the cylinder at the groove base: 35.0 mm
; desired total groove width: 24.8 mm, the milling cutter in use has a radius of: 10.123 mm

```
N10 T1 F400 G94 G54         ;Milling tool, feedrate, type of feedrate, work offset
                            compensation
N30 G0 X25 Z50 SPOS=200     ; Approach the starting position
N35 SETMS(2)                ; Master spindle is now the milling spindle
N40 TRACYL (35.0)           ; Switch on TRACYL, machining diameter 35.0 mm
N50 G55 G19                 ;WO compensation, plane selection: Y/Z plane
N60 S800 M3                 ; Switch on milling spindle
N70 G0 Y70 Z10              ;Starting position Y / Z
N80 G1 X17.5                ;Feed milling tool to groove base
N70 OFFN=12.4               ;Distance from groove side to groove center line 12.4 mm
N90 G1 Y70 Z1 G42           ; Activate TRC, approach groove side
N100 Z-30                   ; Groove cut parallel to cylinder axis
N110 Y20                    ; Groove cut parallel to circumference
N120 G42 G1 Y20 Z-30        ;Start TRC again, approach the other groove side
                            ;Distance from groove side to groove center line remains
                            12.4 mm
N130 Y70 F600               ; Groove cut parallel to circumference
N140 Z1                     ; Groove cut parallel to cylinder axis
N150 Y70 Z10 G40            ; Switch off TRC
N160 G0 X25                 ; Raise milling tool
N170 M5 OFFN=0              ; Switch off milling spindle, delete groove side
                            distance
N180 TRAFOOF                ; Switch off TRACYL
N190 SETMS                  ; Master spindle is now the main spindle again
N200 G54 G18 G0 X25 Z50     ; Approach the starting position
SPOS=200
N210 M2
```

Cycles 10

10.1 Overview of cycles

Cycles are generally applicable technology subroutines that can be used to carry out a specific machining process, such as tapping. These cycles are adapted to individual tasks by parameter assignment.

References

The cycles described here correspond to the SINUMERIK 840D sl cycles. See also SINUMERIK 840D sl Programming Instructions Cycles.

Drilling cycles and turning cycles

The following standard cycles can be carried out using the SINUMERIK 802D sl control system:

- Drilling cycles

  CYCLE81: Drilling, centering

  CYCLE82: Drilling, counterboring

  CYCLE83: Deep-hole drilling

  CYCLE84: Rigid tapping

  CYCLE840: Tapping with compensating chuck

  CYCLE85: Reaming 1 (boring out 1)

  CYCLE86: Boring (boring out 2)

  CYCLE87: Drilling with stop 1 (boring out 3)

  CYCLE88: Drilling with stop 2 (boring out 4)

  CYCLE89: Reaming 2 (boring out 5)

With SINUMERIK 840D, the boring cycles CYCLE85 ... CYCLE89 are called boring 1 ... boring 5, but are nevertheless identical in their function.

- Hole pattern cycles

  HOLES1: Row of holes

  HOLES2: Circle of holes

- Turning cycles

  CYCLE93: Recess

  CYCLE94: Undercut (DIN form E and F)

  CYCLE95: Stock removal with relief cutting

  CYCLE96: Thread undercut

  CYCLE97: Thread cutting

  CYCLE98: Thread chain

The cycles are delivered with the Toolbox and must be loaded via the RS232 interface to the part program memory, if required.

Cycle auxiliary subroutines

The cycles package includes the following auxiliary subroutines:

- cyclest.spf
- steigung.spf and
- meldung.spf

These must always be loaded in the control.

10.2 Programming cycles

A standard cycle is defined as a subroutine with name and parameter list.

Call and return conditions

The G functions effective prior to the cycle call and the programmable offsets remain active beyond the cycle.

The machining plane G17 for drilling cycles or G18 for turning cycles is defined before calling the cycle.

With drilling cycles, the drilling operation is carried out in the axis standing vertically to the current plane.

Messages output during execution of a cycle

During various cycles, messages that refer to the state of machining are displayed on the screen of the control system during program execution.

These message do not interrupt the program execution and continue to be displayed on the screen until the next message appears.

The message texts and their meaning are listed together with the cycle to which they refer.

A summary of all relevant messages is to be found in section 9.4.

Block display during execution of a cycle

The cycle call is displayed in the current block display for the duration of the cycle.

Cycle call and parameter list

The defining parameters for the cycles can be transferred via the parameter list when the cycle is called.

Note

Cycle calls must always be programmed in a separate block.

Basic instructions with regard to the assignment of standard cycle parameters

The Programming Manual describes the parameter list of every cycle with the

- order and the
- type.

It is imperative to observe the order of the defining parameters.

Each defining parameter of a cycle has a certain data type. The parameter being used must be specified when the cycle is called. The following can be transferred in the parameter list:

- R parameters (only numerical values)
- Constants

If R parameters are used in the parameter list, they must first be assigned values in the calling program. Proceed as follows to call the cycles:

- with an incomplete parameter list or

  or

- by leaving out parameters.

If transfer parameters are omitted at the end of the parameter list, the parameter list must be prematurely ended with ")". If any parameters are to be omitted within the list, a comma "..., ,..." must be written as a placeholder.

No plausibility checks are made for parameter values with a limited range of values unless an error response has been specifically described for a cycle.

If when calling the cycle the parameter list contains more entries than parameters are defined in the cycle, the general NC alarm 12340 "Too many parameters" is displayed and the cycle is not executed.

Note

Axis-specific and channel-specific machine data of the spindle must be configured.

Cycle call

The individual methods for writing a cycle are shown in the programming examples provided for the individual cycles.

Simulation of cycles

Programs with cycle calls can be tested first in simulation.

During simulation, the traversing movements of the cycle are visualized on the screen.

10.3 Graphical cycle support in the program editor

The program editor in the control system provides you with programming support to add cycle calls to the program and to enter parameters.

Function

The cycle support consists of three components:

1. Cycle selection
2. Input screens for parameter assignment
3. Help display per cycle.

Summary of required files

The following files constitute the basis for cycle support:

- sc.com
- cov.com

Note

These files are loaded during the commissioning of the control system and must always remain loaded.

Operating the cycle support

To add a cycle call to the program, carry out the following steps one after the other:

- You can branch to selection bars for the individual cycles from the horizontal softkey bar using the existing softkeys "Drilling" and "Turning".
- The cycle selection is carried out using the vertical softkey bar until the appropriate input screenform with the help display appears on the screen.
- The values can be entered either directly (numerical values) or indirectly (R parameters, e.g. R27, or expressions consisting of R parameters, e.g. R27+10).

  If numerical values are entered, a check is carried out to see whether the value is within the admissible range.
- Some parameters that may have only a few values are selected using the toggle key.
- For drilling cycles, it is also possible to call a cycle modally using the vertical "Modal Call" softkey.

  The modal call is selected via "Deselect modal" from the drilling cycles list box.
- Press "OK" to complete your input (or "Abort" in case of error).

Recompiling

Recompiling of program codes serves to make modifications to an existing program using the cycle support.

Position the cursor on the line to be modified and select the "Recompile" softkey.

This will reopen the input screen from which the program piece has been created, and you can modify and accept the values.

10.4 Drilling cycles

10.4.1 General information

Drilling cycles are motional sequences specified according to DIN 66025 for drilling, boring, tapping, etc.

They are called in the form of a subroutine with a defined name and a parameter list.

They all follow a different technological procedure and are therefore parameterized differently.

The drilling cycles can be modally effective, i.e. they are executed at the end of each block which contains motion commands (see section "Overview of instructions" or "Graphical cycle support in the program editor").

There are two types of parameters:

- Geometrical parameters and
- Machining parameters

The geometrical parameters are identical with all drilling cycles. They define the reference and retraction planes, the safety clearance and the absolute or relative final drilling depth. Geometrical parameters are assigned once during the first drilling cycle CYCLE82.

The machining parameters have a different meaning and effect in the individual cycles. They are therefore programmed in each cycle separately.

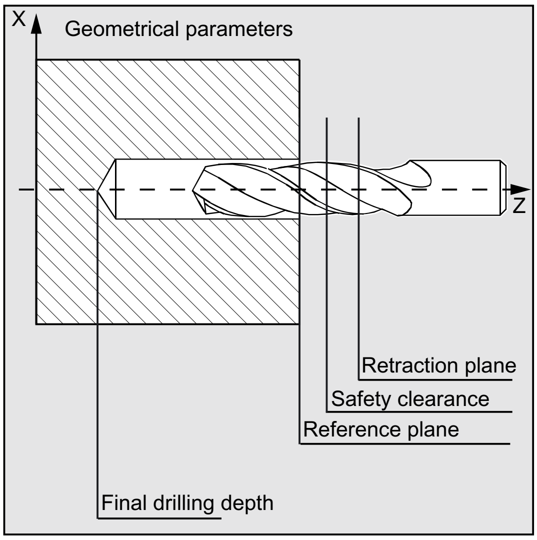


Figure 10-1 Geometrical parameters

10.4.2 Requirements

Call and return conditions

Drilling cycles are programmed independently of the actual axis names. The drilling position must be approached in the higher-level program before the cycle is called.

The required values for feedrate, spindle speed and direction of spindle rotation must be programmed in the part program if there are no defining parameters in the drilling cycle.

The G functions and the current data record active before the cycle was called remain active beyond the cycle.

Plane definition

In the case of drilling cycles, it is generally assumed that the current workpiece coordinate system in which the machining operation is to be performed is to be defined by selecting plane G17 and activating a programmable offset. The drilling axis is always the axis of this coordinate system which stands vertically to the current plane.

A tool length compensation must be selected before the cycle is called. Its effect is always perpendicular to the selected plane and remains active even after the end of the cycle.

In turning, the drilling axis is thus the Z axis. Drilling is performed to the end face of the workpiece.

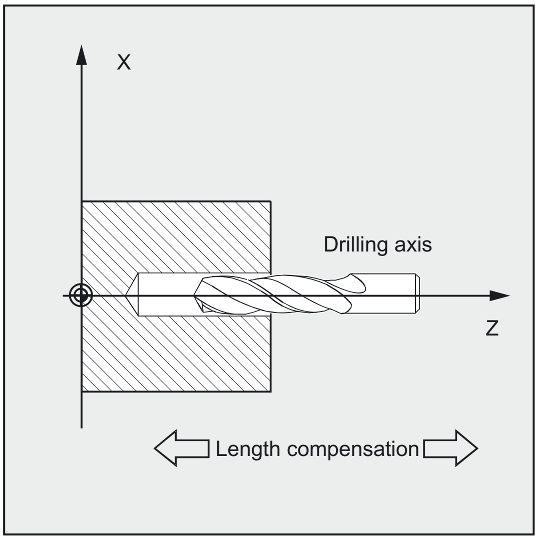


Figure 10-2 Drilling axis when turning

Dwell time programming

The parameters for dwell times in the drilling cycles are always assigned to the F word and must therefore be assigned with values in seconds. Any deviations from this procedure must be expressly stated.

Special features when using drilling cycles on a turning machine

Simple turning machines without driven tools can apply drilling cycles only for drilling on the end face (with Z-axis) in the turning center. These drilling cycles must always be called in the G17 plane.

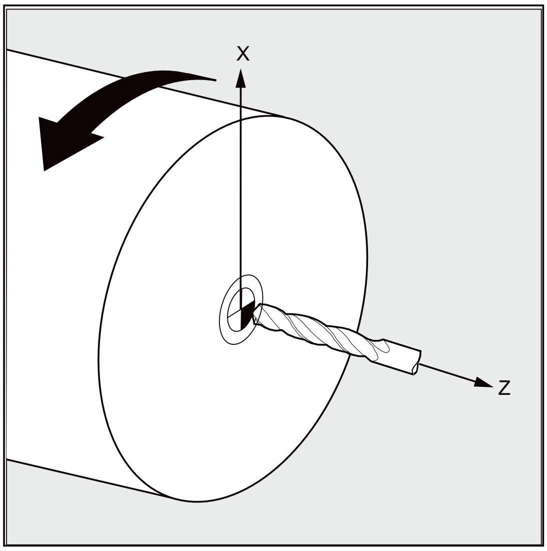


Figure 10-3 Drilling on turning center without a driven tool

Turning machines with driven tools can also drill off-center on the end face or on the peripheral surface if the machine setup permits this.

The following must be observed when drilling off-center on the end face:

- Working plane is G17 - Z is the resulting tool axis.
- The spindle of the driven tool must be declared to the master spindle (SETMS command).
- The drilling position can be programmed either with X and the C-axis or, if TRANSMIT is active, with X and Y.

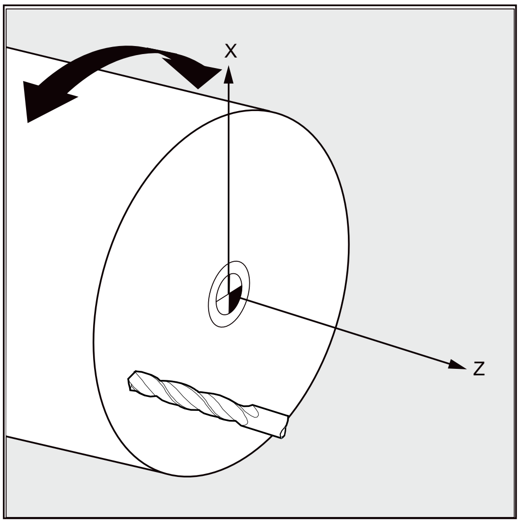


Figure 10-4 Drilling on end face with a driven tool

The following must be observed when drilling on the peripheral surface:

- Working plane is G19 - X is the resulting tool axis.
- The spindle of the driven tool must be declared to the master spindle (SETMS command).
- The drilling position can be programmed either with Z and the C-axis or, if TRACYL is active, with X and Z.

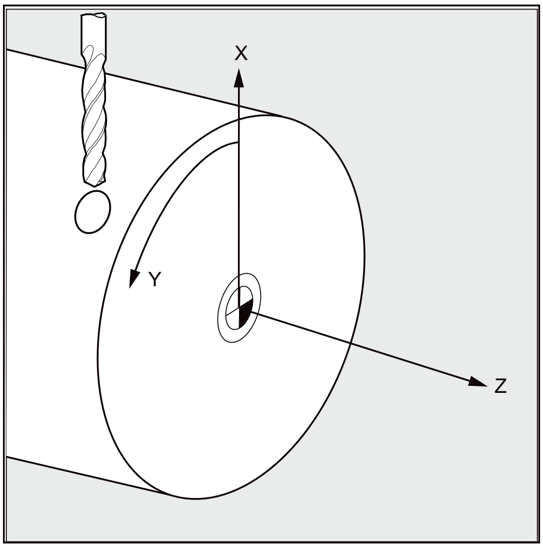


Figure 10-5 Drilling on peripheral surface with a driven tool

10.4.3 Drilling, centering - CYCLE81

Programming

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

Table 10- 1 Parameters for CYCLE81

Parameter	Data type	Significance
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 the reference plane (enter without sign)

Function

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

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

Approach of the reference plane brought forward by the safety clearance by using G0

- Traversing to the final drilling depth at the feedrate programmed in the calling program (G1)
- Retraction to the retraction plane with G0

Explanation of parameters: RFP and RTP (reference plane and retraction plane)

Normally, reference plane (RFP) and return plane (RTP) have different values. The cycle assumes that the retraction plane precedes the reference plane. This means that the distance from the retraction plane to the final drilling depth is larger than the distance from the reference plane to the final drilling depth.

SDIS (safety clearance)

The safety clearance (SDIS) acts with reference to the reference plane. This 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 (final drilling depth)

The final drilling depth can be specified either absolute (DP) or relative (DPR) to the reference plane.

With relative specification, the cycle will calculate the resulting depth automatically using the positions of reference and retraction planes.

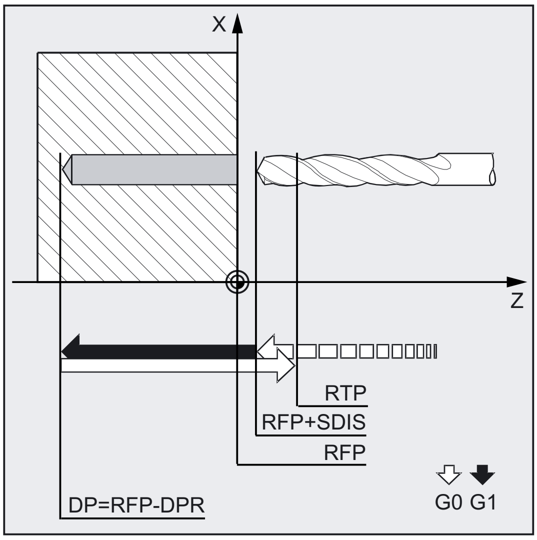


Figure 10-6 Parameters for CYCLE81

Note

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

If the values for reference and retraction planes are identical, a relative depth specification is not permitted. The error message 61101 "Reference plane defined incorrectly" is output and the cycle is not executed. This error message is also output if the retraction plane is located after the reference plane, i.e. its distance to the final drilling depth is smaller.

Programming example: Drilling_centering

This program produces one hole on the end face using the CYCLE81 drilling cycle. The drilling axis is always the Z axis.

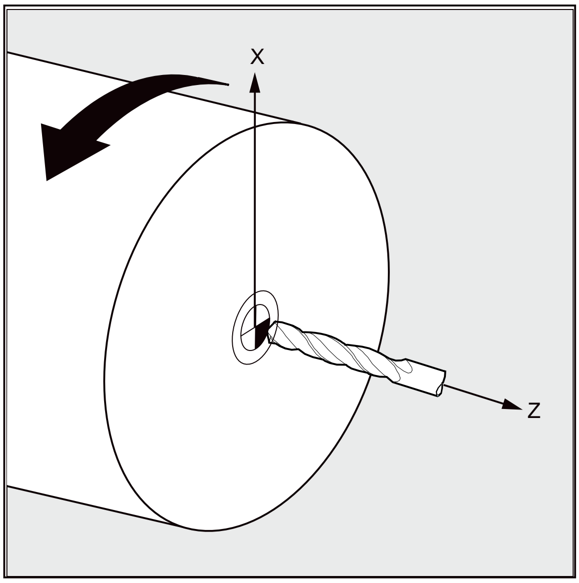


Figure 10-7 Drilling_centering

```
N10 G0 G17 G90 F200 S300 M3       ; Specification of technology values
N20 T3 D1                         ; Load tool
N30 M6
N40 Z10                           ; Approach retraction plane
N50 X0                            ; Approach drilling position
N60 CYCLE81(10, 0, 2, --35,)      ; Cycle call
N70 G0 Z100                       ;Retraction in Z
N100 M2                           ; End of program
```

10.4.4 Drilling, counterboring - CYCLE82:

Programming

CYCLE82 (RTP, RFP, SDIS, DP, DPR, DTB)

parameters

Table 10- 2 Parameters for CYCLE82

Parameter	Data type	Significance
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 the reference plane (enter without sign)
DTB	REAL	Dwell time at final drilling depth (chip breakage)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. A dwell time can be allowed to elapse when the final drilling depth has been reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the final drilling depth with the feedrate (G1) programmed prior to the cycle call
- Dwell time at final drilling depth
- Retraction to the retraction plane with G0

Explanation of the parameters

For parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

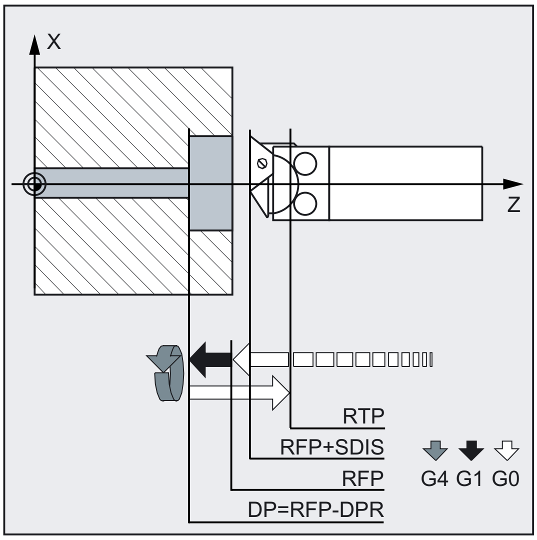


Figure 10-8 Parameters for CYCLE82

DTB (dwell time)

The dwell time to the final drilling depth (chip breakage) is programmed under DTB in seconds.

Note

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

If the values for reference and retraction planes are identical, a relative depth specification is not permitted. The error message 61101 "Reference plane defined incorrectly" is output and the cycle is not executed. This error message is also output if the retraction plane is located after the reference plane, i.e. its distance to the final drilling depth is smaller.

Programming example: Boring_counterboring

The program machines a single hole of a depth of 20 mm at position X0 with cycle CYCLE82.

The dwell time programmed is 3 s, the safety clearance in the drilling axis Z is 2.4 mm.

```
N10 G0 G90 G54 F2 S300 M3          ; Specification of technology values
N20 D1 T6 Z50                      ; Approach retraction plane
N30 G17 X0                         ; Approach drilling position
N40 CYCLE82 (3, 1.1, 2.4, -20, ,   ; Cycle call with absolute final drilling depth
3)                                 and safety clearance
N50 M2                             ; End of program
```

10.4.5 Deep-hole drilling - CYCLE83

Programming

CYCLE83(RTP, RFP, SDIS, DP, DPR, FDEP, FDPR, DAM, DTB, DTS, FRF, VARI)

parameters

Table 10- 3 Parameters for CYCLE83

Parameter	Data type	Description
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 the reference plane (enter without sign)
FDEP	REAL	First drilling depth (absolute)
FDPR	REAL	First drilling depth relative to the reference plane (enter without sign)
DAM	REAL	Amount of degression (enter without sign)
DTB	REAL	Dwell time at final drilling depth (chip breaking)
DTS	REAL	Dwell time at starting point and for swarf removal
FRF	REAL	Feedrate factor for the first drilling depth (enter without sign) Range of values: 0.001 ... 1
VARI	INT	Machining type: Chip breakage=0, Chip removal=1

Function

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

Deep hole drilling is performed with a depth infeed of a maximum definable depth executed several times, increasing gradually until the final drilling depth is reached.

The drill can either be retracted to the reference plane + safety clearance after every infeed depth for swarf removal or retracted in each case by 1 mm for chip breaking.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence:

Deep-hole drilling with chip removal (VARI=1):

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the first drilling depth with G1, the feedrate for which is derived from the feedrate defined with the program call which is subject to parameter FRF (feedrate factor)
- Dwell time at final drilling depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance for swarf removal by using G0
- Dwell time at the starting point (parameter DTS)
- Approach of the drilling depth last reached, reduced by anticipation distance by using G0
- Traversing to the next drilling depth with G1 (sequence of motions is continued until the final drilling depth is reached)
- Retraction to the retraction plane with G0

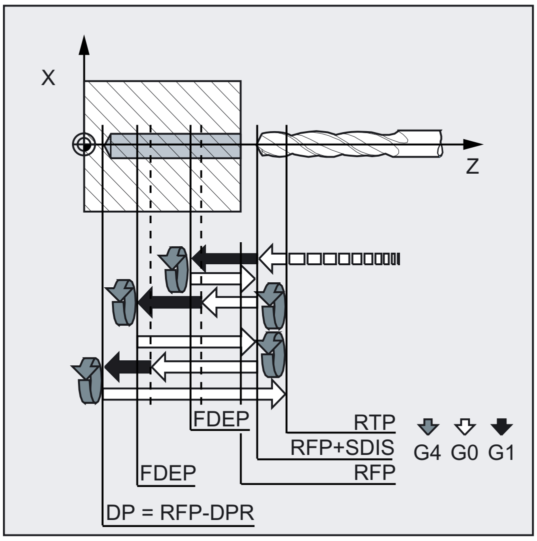


Figure 10-9 Deep-hole drilling with swarf removal

Deep-hole drilling with chip breakage (VARI=0):

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the first drilling depth with G1, the feedrate for which is derived from the feedrate defined with the program call which is subject to parameter FRF (feedrate factor)
- Dwell time at final drilling depth (parameter DTB)
- Retraction by 1 mm from the current drilling depth with G1 and the feedrate programmed in the calling program (for chip breaking)

- Traversing to the next drilling depth with G1 and the programmed feedrate (sequence of motions is continued until the final drilling depth is reached)
- Retraction to the retraction plane with G0

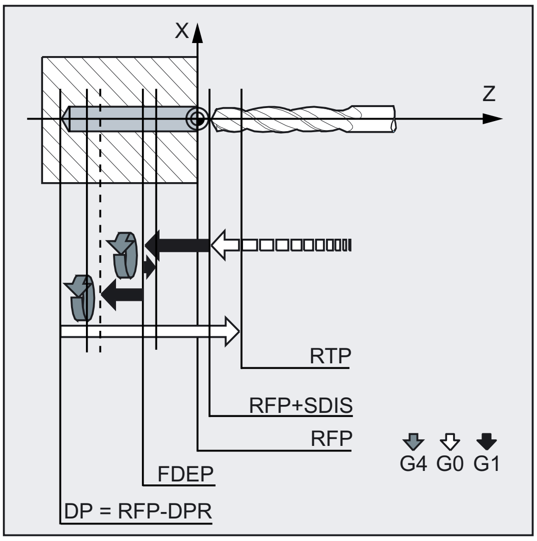


Figure 10-10 Deep-hole drilling with chip breaking

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

Interrelation of the parameters DP (or DPR), FDEP (or FDPR) and DMA

The intermediate drilling depth are calculated in the cycle on the basis of final drilling depth, first drilling depth and amount of degression as follows:

- In the first step, the depth parameterized with the first drilling depth is traversed as long as it does not exceed the total drilling depth.
- From the second drilling depth on, the drilling stroke is obtained by subtracting the amount of degression from the stroke of the last drilling depth, provided that the latter is greater than the programmed amount of degression.
- The next drilling strokes correspond to the amount of degression, as long as the remaining depth is greater than twice the amount of degression.
- The last two drilling strokes are divided and traversed equally and are therefore always greater than half of the amount of degression.
- If the value for the first drilling depth is incompatible with the total depth, the error message 61107 "First drilling depth defined incorrectly" is output and the cycle is not executed.

The FDPR parameter has the same effect in the cycle as the DPR parameter. If the values for the reference and retraction planes are identical, the first drilling depth can be defined as a relative value.

If the first drilling depth is programmed larger than the final drilling depth, the final drilling depth is never exceeded. The cycle will reduce the first drilling depth automatically as far as the final drilling depth is reached when drilling only once, and will therefore drill only once.

DTB (dwell time)

The dwell time to the final drilling depth (chip breakage) is programmed under DTB in seconds.

DTS (dwell time)

The dwell time at the starting point is only performed if VARI=1 (chip removal).

FRF (feedrate factor)

With this parameter, you can enter a reduction factor for the active feedrate which only applies to the approach to the first drilling depth in the cycle.

VARI (machining type)

If parameter VARI=0 is set, the drill retracts 1 mm after reaching each drilling depth for chip breaking. If VARI=1 (for chip removal), the drill traverses in each case to the reference plane shifted by the amount of the safety clearance.

Note

The anticipation distance is calculated internally in the cycle as follows:

- If the drilling depth is 30 mm, the value of the anticipation distance is always 0.6 mm.
- For larger drilling depths, the formula drilling depth /50 is used (maximum value 7 mm).

Programming example: Deep-hole drilling

This program executes the cycle CYCLE83 at the position X0. The first drill hole is drilled with a dwell time zero and machining type chip breaking. The final drilling depth and the first drilling depth are entered as absolute values. The drilling axis is the Z axis.

```
N10 G0 G54 G90 F5 S500 M4                        ;Specification of technology values
N20 D1 T6 Z50                                    ; Approach retraction plane
N30 G17 X0                                       ; Approach drilling position
N40 CYCLE83(3.3, 0, 0, -80, 0, -10, 0, 0, 0,     ;Call of cycle, depth parameters
0, 1, 0)                                         with absolute values
N50 M2                                           ; End of program
```

10.4.6 Rigid tapping - CYCLE84

Programming

CYCLE84(RTP, RFP, SDIS, DP, DPR, DTB, SDAC, MPIT, PIT, POSS, SST, SST1)

parameters

Table 10- 4 Parameters for CYCLE84

Parameter	Data type	Significance
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 the reference plane (enter without sign)
DTB	REAL	Dwell time at thread depth (chip breakage)
SDAC	INT	Direction of rotation after end of cycle Values: 3, 4 or 5 (for M3, M4 or M5)
MPIT	REAL	Thread lead as a thread size (signed): Range of values 3 (for M3) to 48 (for M48); the sign determines the direction of rotation in the thread
PST	REAL	Thread lead as a value (signed) Range of values: 0.001 ... 2000.000 mm); the sign determines the direction of rotation in the thread
POSS	REAL	Spindle position for oriented spindle stop in the cycle (in degrees)
SST	REAL	Speed for tapping
SST1	REAL	Speed for retraction

Function

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

CYCLE84 can be used to make tapped holes without compensating chuck.

Note

CYCLE84 can be used if the spindle to be used for the boring operation is technically able to be operated in the position-controlled spindle operation.

For tapping with compensating chuck, a separate cycle CYCLE840 is provided.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Oriented spindle stop (value in parameter POSS) and switching the spindle to axis mode
- Tapping to final drilling depth and speed SST
- Dwell time at thread depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance, speed SST1 and direction reversal
- Retraction to the retraction plane with G0; spindle mode is reinitiated by reprogramming the spindle speed active before the cycle was called and the direction of rotation programmed under SDAC

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

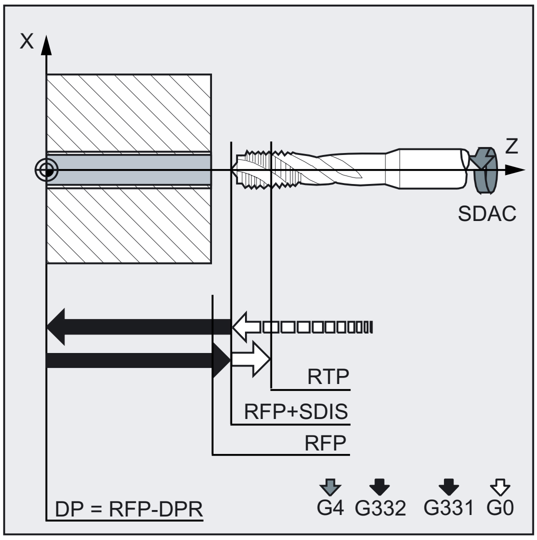


Figure 10-11 Parameters for CYCLE84

DTB (dwell time)

The dwell time must be programmed in seconds. When tapping blind holes, it is recommended that you omit the dwell time.

SDAC (direction of rotation after end of cycle)

The direction of the spindle rotation after end of cycle must be programmed under SDAC. For tapping, the direction is changed automatically by the cycle.

MPIT and PIT (thread lead as a thread size and as a value)

The value for the thread lead can be defined either as the thread size (for metric threads between M3 and M48 only) or as a value (distance from one thread turn to the next as a numerical value). The parameter not required in each case is omitted in the call or is assigned the value zero.

RH or LH threads are defined by the sign of the lead parameters:

- Positive value → right (same as M3)
- Negative value → left (same as M4)

If the two lead parameters have conflicting values, alarm 61001 "Thread lead wrong" is generated by the cycle and cycle execution is aborted.

POSS (spindle position)

In the cycle, the spindle is oriented and stopped, and brought into position control before tapping with the SPOS command.

The spindle position for this spindle stop is programmed under POSS.

SST (speed)

Parameter SST contains the spindle speed for the tapping block.

SST1 (retraction speed)

The speed for retraction from the tapped hole is programmed under SST1 with G332. If this parameter is assigned the value zero, retraction is carried out at the speed programmed under SST.

Note

The direction of rotation when tapping in the cycle is always reversed automatically.

Programming example: Rigid tapping

Rigid tapping is carried out at position X0; the drilling axis is the Z axis. No dwell time is programmed; the depth is programmed as a relative value. The parameters for the direction of rotation and for the lead must be assigned values. A metric thread M5 is tapped.

```
N10 G0 G90 G54 T6 D1                          ; Specification of technology values
N20 G17 X0 Z40                                ; Approach drilling position
N30 CYCLE84(4, 0, 2, , 30, , 3, 5, , 90,      ; Cycle call; parameter PIT has been
200, 500)                                     omitted; no value is entered for the
                                              absolute depth or the dwell time;
                                              spindle stop at 90 degrees; speed for
                                              tapping is 200, speed for retraction is
                                              500
N40 M2                                        ; End of program
```

10.4.7 Tapping with compensating chuck - CYCLE840

Programming

CYCLE840(RTP, RFP, SDIS, DP, DPR, DTB, SDR, SDAC, ENC, MPIT, PIT, AXN)

parameters

Table 10- 5 Parameters for CYCLE840

Parameter	Data type	Significance
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 the reference plane (enter without sign)
DTB	REAL	Dwell time at thread depth (chip breakage)
SDR	INT	Direction of rotation for retraction Values: 0 (automatic direction reversal), 3 or 4 (for M3 or M4)
SDAC	INT	Direction of rotation after end of cycle Values: 3, 4 or 5 (for M3, M4 or M5)
ENC	INT	Tapping with/without encoder Values: 0 = with encoder, 1 = without encoder
MPIT	REAL	Thread lead as a thread size (signed): Range of values 3 (for M3) to 48 (for M48)
PST	REAL	Thread lead as a value (signed) Range of values: 0.001 ... 2000.000 mm
AXN	INT	Tool axis
		Values: 1 = 1. axis of the plane 2 = 2. axis of the plane otherwise, 3rd axis of the plane

Function

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

Use this cycle to perform tapping with compensating chuck:

- without encoder and
- with encoder.

Sequence of operations: Tapping with compensating chuck without encoder

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Tapping to the final drilling depth
- Dwell time at tapping depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance
- Retraction to the retraction plane with G0

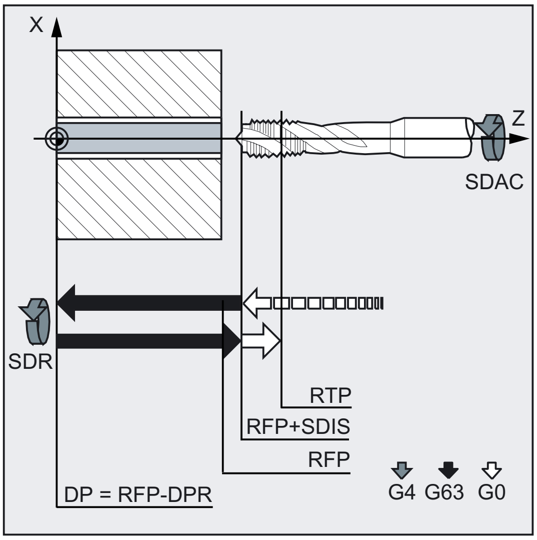


Figure 10-12 Motion sequence without encoder

Sequence of operations: Tapping with compensating chuck with encoder

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Tapping to the final drilling depth
- Dwell time at thread depth (parameter DTB)

- Retraction to the reference plane brought forward by the safety clearance
- Retraction to the retraction plane with G0

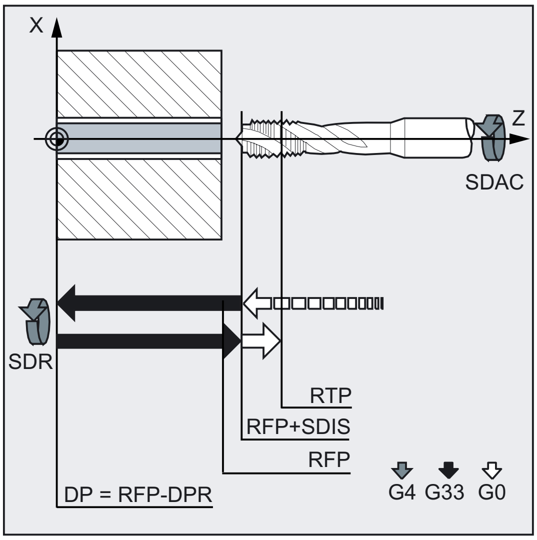


Figure 10-13 Motion sequence with encoder.

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

DTB (dwell time)

The dwell time must be programmed in seconds. It is only effective in tapping without encoder.

SDR (direction of rotation for retraction)

SDR=0 must be set if the spindle direction is to reverse automatically.

If the machine data is defined such that no encoder is set (in this case, machine data MD30200 $MA_NUM_ENCS is 0), the parameter must be assigned the value 3 or 4 for the direction of rotation; otherwise, alarm 61202 "No spindle direction programmed" is output and the cycle is aborted.

SDAC (direction of rotation)

Because the cycle can also be called modally (see section "Graphical cycle support in the program editor"), it requires a direction of rotation for tapping further threaded holes. This is programmed in parameter SDAC and corresponds to the direction of rotation programmed before the first call in the higher-level program. If SDR=0, the value assigned to SDAC has no meaning in the cycle and can be omitted in the parameterization.

ENC (tapping)

If tapping is to be performed without encoder although an encoder exists, parameter ENC must be assigned value 1.

If, however, no encoder is installed and the parameter is assigned the value 0, it is ignored in the cycle.

MPIT and PIT (thread lead as a thread size and as a value)

The parameter for the lead is only relevant if tapping is performed with encoder. The cycle calculates the feedrate from the spindle speed and the lead.

The value for the thread lead can be defined either as the thread size (for metric threads between M3 and M48 only) or as a value (distance from one thread turn to the next as a numerical value). The parameter not required in each case is omitted in the call or is assigned the value zero.

If the two lead parameters have conflicting values, alarm 61001 "Thread lead wrong" is generated by the cycle and cycle execution is interrupted.

Note

Depending on the settings in machine data MD30200 $MA_NUM_ENCS, the cycle selects whether tapping is to be performed with or without encoder.

The direction of rotation for the spindle must be programmed with M3 or M4.

In thread blocks with G63, the values of the feedrate override switch and spindle speed override switch are frozen to 100%.

A longer compensating chuck is usually required for tapping without encoder.

AXN (tool axis)

The following figure presents the options for the drilling axes to be selected.

With G18:

- AXN=1 ;Corresponds to Z
- AXN=2 ;Corresponds to X
- AXN=3 ;Corresponds to Y (if Y axis is present)

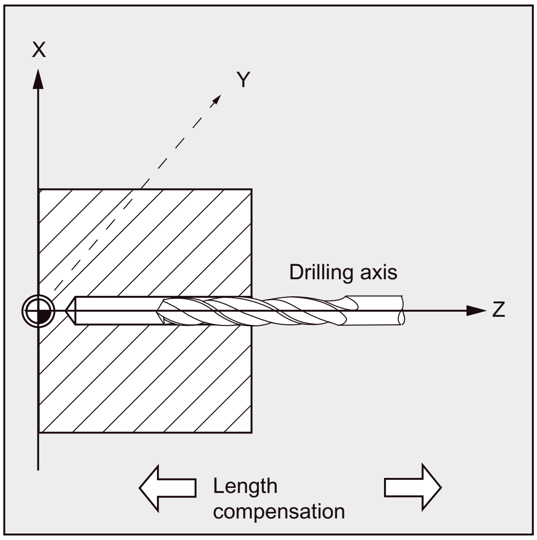


Figure 10-14 AXN (tool axis)

Using AXN (number of the drilling axis) to program the drilling axis enables the drilling axis to be directly programmed.

AXN=1	1. axis of the plane
AXN=2	2. axis of the plane
AXN=3	3. axis of the plane

For example, to machine a center hole (in Z) in the G18 plane, you program:

G18

AXN=1

Programming example: Tapping without encoder

Tapping is carried out without encoder at position X0; the drilling axis is the Z axis. The parameters SDR and SDAC for the direction of rotation must be assigned; parameter ENC is assigned the value 1, the value for the depth is the absolute value. Lead parameter PIT can be omitted. A compensating chuck is used in machining.

```
N10 G90 G0 G54 D1 T6 S500 M3                 ; Specification of technology
                                             values
N20 G17 X0 Z60                               ; Approach drilling position
N30 G1 F200                                  ; Setting the path feedrate
N40 CYCLE840(3, 0, , -15, 0, 1, 4, 3, 1, , ,3)    ; Cycle call; dwell time 1 s;
                                             direction of rotation for
                                             retraction M4; direction of
                                             rotation after cycle M3; no
                                             safety clearance
                                             The MPIT and PIT parameters have
                                             been omitted.
N50 M2                                       ; End of program
```

Programming example: Tapping with encoder

This program is used for tapping with encoder at position X0. The drilling axis is the Z axis. The lead parameter must be defined, automatic reversal of the direction of rotation is programmed. A compensating chuck is used in machining.

```
N10 G90 G0 G54 D1 T6 S500 M3                 ; Specification of technology
                                             values
N20 G17 X0 Z60                               ; Approach drilling position
N30 G1 F200                                  ; Setting the path feedrate
N40 CYCLE840(3, 0, , -15, 0, 0, , ,0, 3.5, ,3)    ; Cycle call without safety
                                             clearance
N50 M2                                       ; End of program
```

10.4.8 Reaming1 (boring 1) – CYCLE85

Programming

CYCLE85 (RTP, RFP, SDIS, DP, DPR, DTB, FFR, RFF)

parameters

Table 10- 6 Parameters for CYCLE85

Parameter	Data type	Significance
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 the reference plane (enter without sign)
DTB	REAL	Dwell time at final drilling depth (chip breakage)
FFR	REAL	Feedrate
RFF	REAL	Retraction feedrate

Function

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

The inward and outward movement is performed at the feedrate assigned to FFR and RFF respectively.

This cycle can be used for reaming of bore holes.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

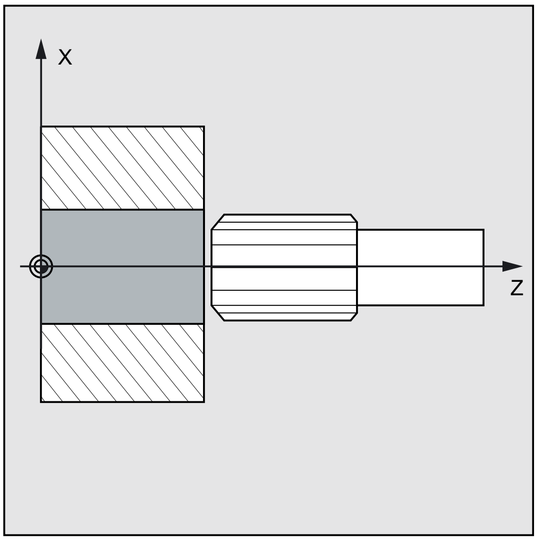


Figure 10-15 Drill position

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the final drilling depth with G1 and at the feedrate programmed under the parameter FFR
- Dwell time at final drilling depth
- Retraction to the reference plane brought forward by the safety clearance with G1 and the retraction feedrate defined under the parameter RFF
- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

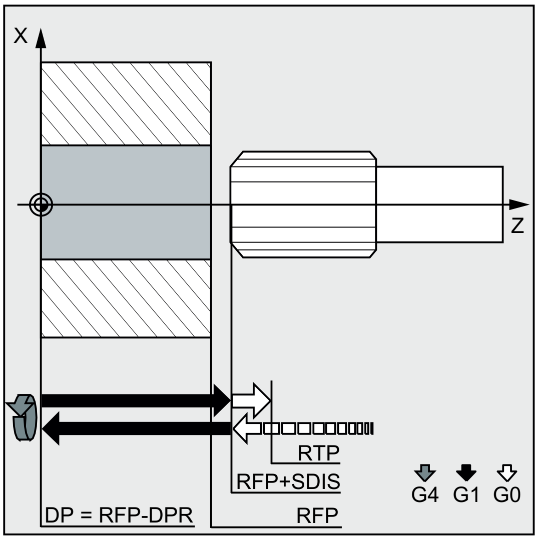


Figure 10-16 Parameters for CYCLE85

DTB (dwell time)

The dwell time to the final drilling depth is programmed under DTB in seconds.

FFR (feedrate)

The feedrate value programmed under FFR is active in drilling.

RFF (retraction feedrate)

The feedrate value programmed under RFF is active when retracting from the hole to the reference plane + safety clearance.

Programming example: First boring pass

CYCLE85 is called at Z70 X0. The drilling axis is the Z axis. The value for the final drilling depth in the cycle call is programmed as a relative value; no dwell time is programmed. The workpiece upper edge is at Z0.

```
N10 G90 G0 S300 M3
N20 T3 G17 G54 Z70 X0                    ; Approach drilling position
N30 CYCLE85(10, 2, 2, , 25, , 300, 450)  ; Cycle call, no dwell time
                                         programmed
N40 M2                                   ; End of program
```

10.4.9 Boring (boring 2) – CYCLE86

Programming

CYCLE86 (RTP, RFP, SDIS, DP, DPR, DTB, SDIR, RPA, RPO, RPAP, POSS)

parameters

Table 10- 7 Parameters for CYCLE86

Parameter	Data type	Significance
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 the reference plane (enter without sign)
DTB	REAL	Dwell time at final drilling depth (chip breakage)
SDIR	INT	Direction of rotation Values: 3 (for M3), 4 (for M4)
RPA	REAL	Retraction path along the 1st axis of the plane (incremental, enter with sign)
RPO	REAL	Retraction path along the 2nd axis of the plane (incremental, enter with sign)
RPAP	REAL	Retraction path along the boring axis (incremental, enter with sign)
POSS	REAL	Spindle position for oriented spindle stop in the cycle (in degrees)

Function

The cycle supports the boring of holes with a boring bar.

The tool drills at the programmed spindle speed and feedrate velocity up to the entered drilling depth.

With boring 2, oriented spindle stop is activated once the drilling depth has been reached. Then, the programmed retraction positions are approached in rapid traverse and, from there, the retraction plane.

CYCLE86 can be applied on a turning machine only with TRANSMIT in the G17 plane and with a driven tool (see section "End face milling - TRANSMIT").

Z is the tool axis in this case. The drilling positions are programmed internally in the cycle in the X-Y plane.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time to final drilling depth
- Oriented spindle stop at the spindle position programmed under POSS
- Traverse retraction path in up to three axes with G0
- Retraction in the boring axis to the reference plane brought forward by the safety clearance by using G0
- Retraction to the retraction plane with G0 (initial drilling position in both axes of the plane)

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

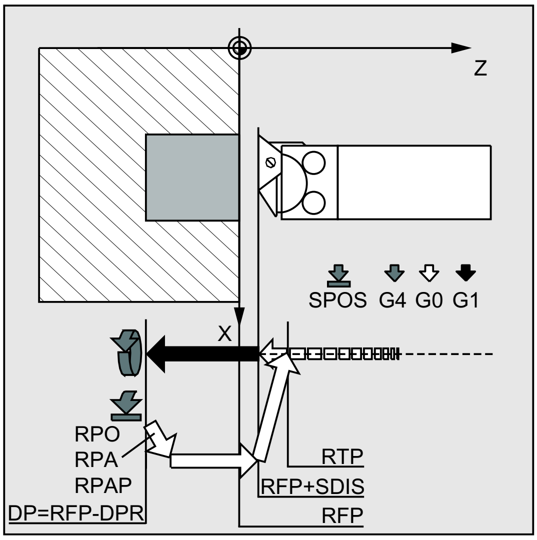


Figure 10-17 Parameters for CYCLE86

DTB (dwell time)

The dwell time to the final drilling depth (chip breakage) is programmed under DTB in seconds.

SDIR (direction of rotation)

With this parameter, you determine the direction of rotation with which boring is performed in the cycle. If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is generated and the cycle is not executed.

RPA (retraction path along the 1st axis)

Use this parameter to define a retraction motion along the 1st axis (abscissa), which is performed after reaching the final drilling depth and oriented spindle stop.

RPO (retraction path along the 2nd axis)

Use this parameter to define a retraction motion along the 2nd axis (ordinate), which is performed after reaching the final drilling depth and oriented spindle stop.

RPAP (retraction path along the boring axis)

You use this parameter to define a retraction movement along the drilling axis, which is executed after the final drilling axis has been reached and oriented spindle stop has been performed.

POSS (spindle position)

Use POSS to program the spindle position for the oriented spindle stop in degrees, which is performed after the final drilling depth has been reached.

Note

It is possible to stop the active spindle with orientation. The angular value is programmed using a transfer parameter.

Cycle CYCLE86 can be used if the spindle to be used for the boring operation is technically able to go into position-controlled spindle operation.

Programming example: Second boring pass

CYCLE86 will be used to drill on the end face in the X-Y plane at position X20 Y20. The drilling axis is the Z axis. The final drilling depth is programmed as an absolute value; no safety clearance is specified. The dwell time at the final drilling depth is 2 sec. The top edge of the workpiece is positioned at Z10. In the cycle, the spindle is to rotate with M3 and to stop at 45 degrees.

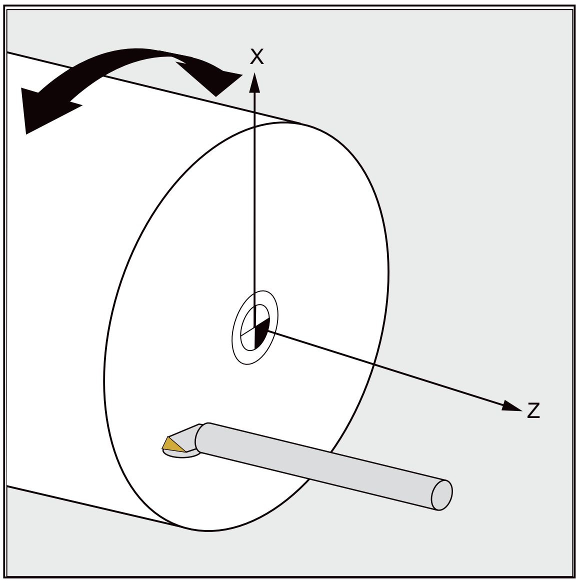


Figure 10-18 Second boring pass

```
N10 G0 G90 X0 Z100 SPOS=0                         ;Approach start position
N15 SETMS(2)                                      ; Master spindle is now the
                                                  milling spindle
N20 TRANSMIT                                      ; Activate TRANSMIT function
N35 T10 D1                                        ; Load tool
N40 M6
N50 G17 G0 G90 X20 Y20                            ;Drilling position
N60 S800 M3 F500
N70 CYCLE86(112, 110, , 77, 0, 2, 3, -1, -1, 1,   ; Cycle call with absolute
45)                                               drilling depth
N80 G0 Z100
N90 TRAFOOF                                       ; Switch off TRANSMIT
N95 SETMS                                         ;Master spindle is now the main
                                                  spindle again
N200 M2                                           ; End of program
```

10.4.10 Boring with stop 1 (boring pass 3) – CYCLE87

Programming

CYCLE87 (RTP, RFP, SDIS, DP, DPR, SDIR)

parameters

Table 10- 8 Parameters for CYCLE87

Parameter	Data type	Significance
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 the reference plane (enter without sign)
SDIR	INT	Direction of rotation Values: 3 (for M3), 4 (for M4)

Function

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

During boring 3, a spindle stop without orientation M5 is generated after reaching the final drilling depth, followed by a programmed stop M0. Pressing the NC START key continues the retraction movement at rapid traverse until the retraction plane is reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Spindle stop with M5
- Press NC START
- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

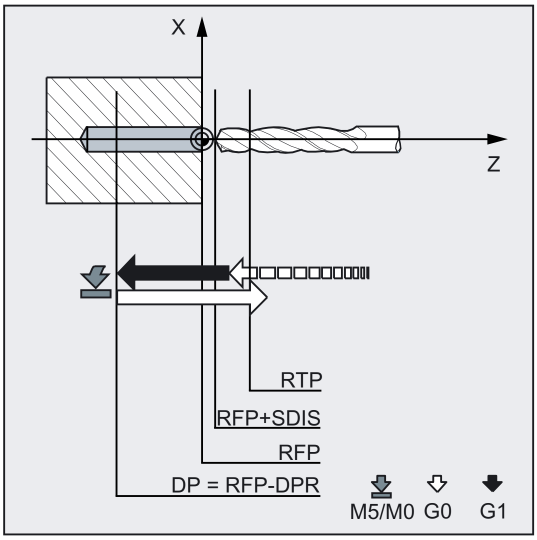


Figure 10-19 Parameters for CYCLE87

SDIR (direction of rotation)

This parameter determines the direction of rotation with which the drilling operation is carried out in the cycle.

If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is generated and the cycle is aborted.

Programming example: Third boring

CYCLE87 is called at position X0 in the XY plane. The drilling axis is the Z axis. The final drilling depth is specified as an absolute value. The safety clearance is 2 mm.

```
N20 G0 G17 G90 F200 S300 X0                         ;Specification of technology
                                                    values and drilling position
N30 D3 T3 Z13                                       ; Approach retraction plane
N50 CYCLE87 (13, 10, 2, -7, , 3)                    ;Cycle call with programmed
                                                    direction of rotation of spindle
                                                    M3
N60 M2                                              ; End of program
```

10.4.11 Drilling with stop 2 (boring 4) - CYCLE88

Programming

CYCLE88 (RTP, RFP, SDIS, DP, DPR, DTB, SDIR)

parameters

Table 10- 9 CYCLE88 parameters

Parameter	Data type	Significance
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 the reference plane (enter without sign)
DTB	REAL	Dwell time at final drilling depth (chip breakage)
SDIR	INT	Direction of rotation Values: 3 (for M3), 4 (for M4)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. During boring pass 4, a dwell time, a spindle stop without orientation M5 and a programmed stop M0 are generated when the final drilling depth is reached. Pressing the NC START key traverses the outward movement at rapid traverse until the retraction plane is reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time at final drilling depth
- Spindle and program stop with M5 M0. After program stop, press the NC START key.
- Retraction to the retraction plane with G0

Explanation of the parameters

For parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

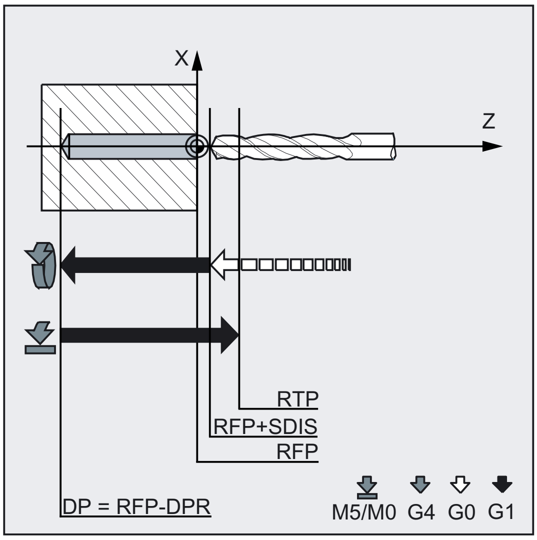


Figure 10-20 CYCLE88 parameters

DTB (dwell time)

The dwell time to the final drilling depth (chip breakage) is programmed under DTB in seconds.

SDIR (direction of rotation)

The programmed direction of rotation is active for the distance to be traversed to the final drilling depth.

If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is generated and the cycle is aborted.

Programming example: Fourth boring pass

CYCLE88 is called at X0. The drilling axis is the Z axis. The safety clearance is programmed with 3 mm; the final drilling depth is specified relative to the reference plane. M4 is active in the cycle.

```
N10 G17 G54 G90 F1 S450 M3 T1              ; Specification of technology
                                           values
N20 G0 X0 Z10                              ;Approach drilling position
N30 CYCLE88 (5, 2, 3, , 72, 3, 4)          ; Cycle call with programmed
                                           direction of rotation of spindle
                                           M4
N40 M2                                     ; End of program
```

10.4.12 Reaming 2 (boring 5) – CYCLE89

Programming

CYCLE89 (RTP, RFP, SDIS, DP, DPR, DTB)

parameters

Table 10- 10 Parameter CYCLE89

parameters	Data type	Significance
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 the reference plane (enter without sign)
DTB	REAL	Dwell time at final drilling depth (chip breakage)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. When the final drilling depth is reached, a dwell time can be programmed.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time to final drilling depth
- Retraction up to the reference plane brought forward by the safety clearance using G1 and the same feedrate value
- Retraction to the retraction plane with G0

Explanation of the parameters

For parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

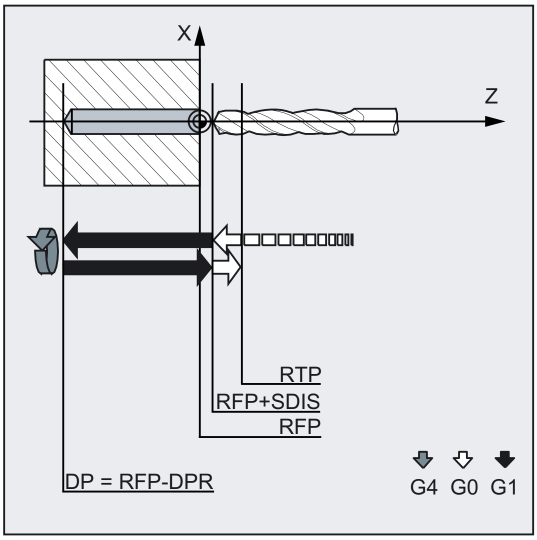


Figure 10-21 Parameter CYCLE89

DTB (dwell time)

The dwell time to the final drilling depth (chip breakage) is programmed under DTB in seconds.

Programming example: Fifth boring

At X0, the CYCLE89 drilling cycle is called with a safety clearance of 5 mm and specification of the final drilling depth as an absolute value. The drilling axis is the Z axis.

```
N10 G90 G17 F100 S450 M4                          ; Specification of technology
                                                  values
N20 G0 X0 Z107                                    ;Approach drilling position
N30 CYCLE89(107, 102, 5, 72, ,3)                  ; Cycle call
N40 M2                                            ; End of program
```

10.4.13 Row of holes - HOLES1

Programming

HOLES1 (SPCA, SPCO, STA1, FDIS, DBH, NUM)

parameters

Table 10- 11 HOLES1 parameters

Parameter	Data type	Significance
SPCA	REAL	1. axis of the plane (abscissa) of a reference point on the straight line (absolute)
SPCO	REAL	2. axis of the plane (ordinate) of this reference point (absolute)
STA1	REAL	Angle to the 1st axis of the plane (abscissa) Range of values: -180<STA1<=180 degrees
FDIS	REAL	Distance from the first hole to the reference point (enter without sign)
DBH	REAL	Distance between the holes (enter without sign)
NUM	INT	Number of drill holes

Function

This cycle can be used to produce a row of holes, i.e. a number of drill holes arranged along a straight line, or a grid of holes. The type of drill hole is determined by the drilling cycle that has already been called modally.

On a turning machine, cycle CYCLE86 can only be used with TRANSMIT in the G17 plane and with a driven tool (see section "End face milling - TRANSMIT").

Z is the tool axis in this case. The drilling positions are programmed internally in the cycle in the X-Y plane.

Sequence

To avoid unnecessary travel, the cycle calculates whether the row of holes is machined starting from the first hole or the last hole from the actual position of the plane axes and the geometry of the row of holes. The drilling positions are then approached one after the other at rapid traverse.

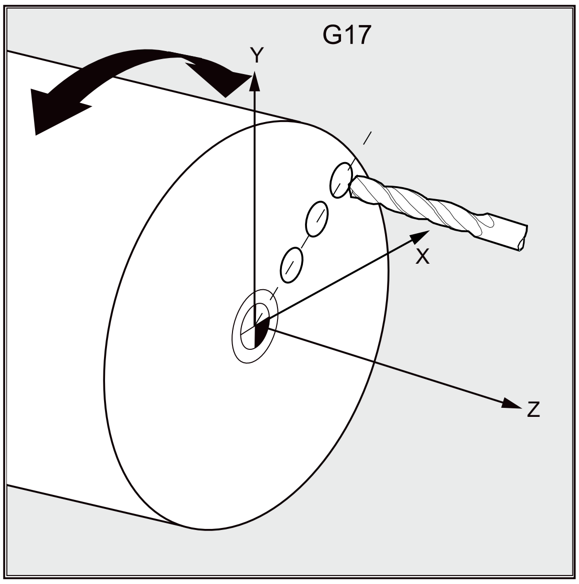


Figure 10-22 Sequence

Explanation of the parameters

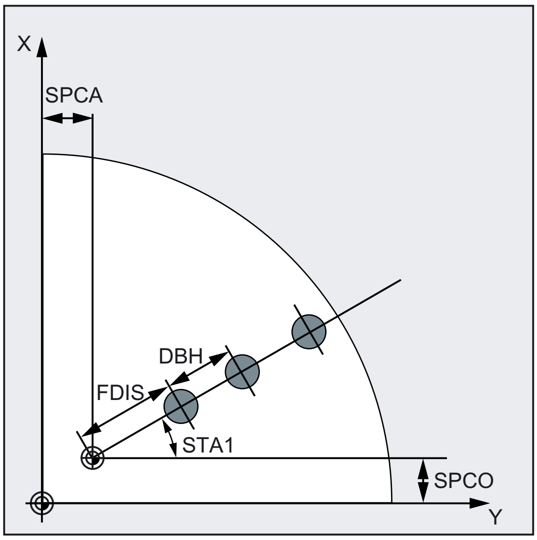


Figure 10-23 HOLES1 parameters

SPCA and SPCO (reference point on the 1st axis of the plane and of the 2nd axis of the plane)

One point along the straight line of the row of holes is defined as the reference point for determining the spacing between the holes. The distance to the first hole FDIS is defined from this point.

STA1 (angle)

The straight line can be arranged in any position in the plane. It is specified both by the point defined by SPCA and SPCO and by the angle contained by the straight line and the 1st axis of the workpiece coordinate system that is active when the cycle is called. The angle is entered under STA1 in degrees.

FDIS and DBH (distance)

The distance of the first hole to the reference point defined under SPCA and SPCO is specified with FDIS. The parameter DBH contains the distance between any two holes.

NUM (number)

The NUM parameter is used to define the number of drill holes.

Programming example: Row of holes

You can use this program to machine a series of holes from 4 tapped holes on the end face of a turned part. The holes are situated at a 45 degree angle relative to the X axis, and the reference point is located on the turning center. The first hole has a distance of 15 mm, and the distance between holes is 10 mm.

The geometry of the row of holes is described by the cycle HOLES1. First, drilling is carried out using CYCLE82, and then tapping is performed using CYCLE84 (tapping without compensating chuck). The holes are 22 mm in depth (difference between reference plane and final drilling depth).

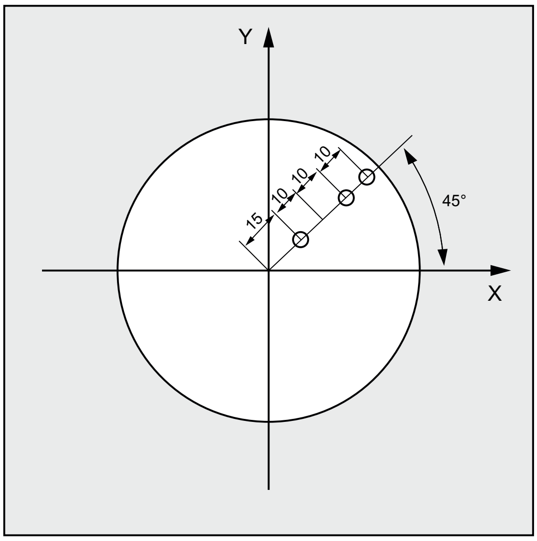


Figure 10-24 Programming example: Row of holes

```
N10 G0 G90 X0 Z10 SPOS=0                            ; Approach starting position
N15 SETMS(2)                                        ; Master spindle is now the
                                                    milling spindle
N20 TRANSMIT                                        ; Activate TRANSMIT function
N25 G17 G90 X0 Y0
N30 F30 S500 M3                                     ; Specification of technology
                                                    values
N35 T10 D1                                          ; Change drilling machine
N40 M6
N45 MCALL CYCLE82(10, 0, 2, --22, 0, 1)             ; Modal call of drilling cycle
N50 HOLES1(0, 0, 45, 15, 10, 4)                     ;Call row-of-holes cycle
N55 MCALL                                           ; Deselect modal call
N60 T11 D1
N65 M6                                              ;Change tap
N70 G90 G0 X0 Z10 Y0                                ; Approach starting position
N75 MCALL CYCLE84(10, 0, 2, --22, 0, , 3, , 4.2,    ; Modal call of the tapping cycle
,300,)
N80 HOLES1(0, 0, 45, 15, 10, 4)                     ;Call row-of-holes cycle again
N85 MCALL                                           ; Deselect modal call
N90 TRAFOOF                                         ; Switch off TRANSMIT
N95 SETMS                                           ; Master spindle is now the main
                                                    spindle again
N100 M2                                             ; End of program
```

Programming example: Grid of holes

This program can be used to machine a grid of holes consisting of 3 rows with 5 holes each situated on the end face of a turned part and separated by a distance of 10 mm. The starting point of the grid of holes is X-20 Y-10.

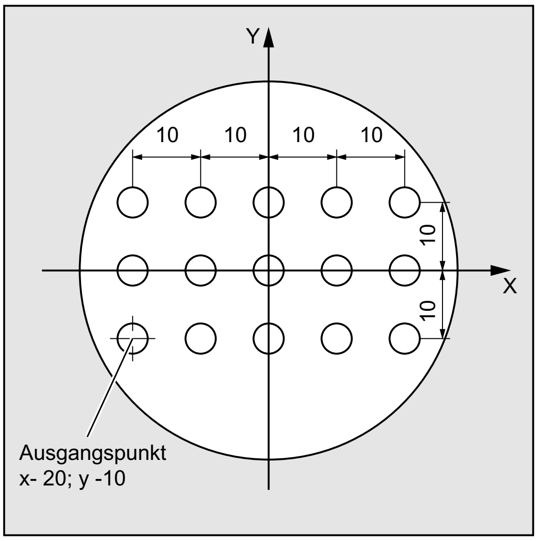


Figure 10-25 Programming example: Grid of holes

```
N10 G0 G90 X0 Z10 SPOS=0                          ; Approach starting position
N15 SETMS(2)                                      ; Master spindle is now the
                                                  milling spindle
N20 TRANSMIT                                      ; Activate TRANSMIT function
N25 G17 G90 X-20 Y-10
N30 F30 S500 M3                                   ; Specification of technology
                                                  values
N35 T10 D1                                        ; Change drilling machine
N40 M6
N45 MCALL CYCLE82(10, 0, 2, --22, 0, 1)           ; Modal call of drilling cycle
N50 HOLES1(--20, --10, 0, 0, 10, 5)               ;Call cycle for 1st row
N60 HOLES1(--20, 0, 0, 0, 10, 5)                  ;Call cycle for 2nd row
N70 HOLES1(--20, 10, 0, 0, 10, 5)                 ;Call cycle for 3rd row
N80 MCALL                                         ; Deselect modal call
N90 TRAFOOF                                       ; Switch off TRANSMIT
N95 SETMS                                         ; Master spindle is now the main
                                                  spindle again
N100 M2                                           ; End of program
```

10.4.14 Circle of holes - HOLES2

Programming

HOLES2 (CPA, CPO, RAD, STA1, INDA, NUM)

parameters

Table 10- 12 HOLES2 parameters

Parameter	Data type	Significance
CPA	REAL	Center point of circle of holes (absolute), 1st axis of the plane
CPO	REAL	Center point of circle of holes (absolute), 2nd axis of the plane
RAD	REAL	Radius of circle of holes (enter without sign)
STA1	REAL	Starting angle Range of values: -180<STA1<=180 degrees
INDA	REAL	Incrementing angle
NUM	INT	Number of drill holes

Function

Use this cycle to machine a circle of holes. The machining plane must be defined before the cycle is called.

The type of drill hole is determined by the drilling cycle that has already been called modally.

The hole pattern cycle HOLES2 can be applied on a turning machine only with TRANSMIT in the G17 plane and with a driven tool (see section "End face milling - TRANSMIT").

Z is the tool axis in this case. The drilling positions are programmed internally in the cycle in the X-Y plane.

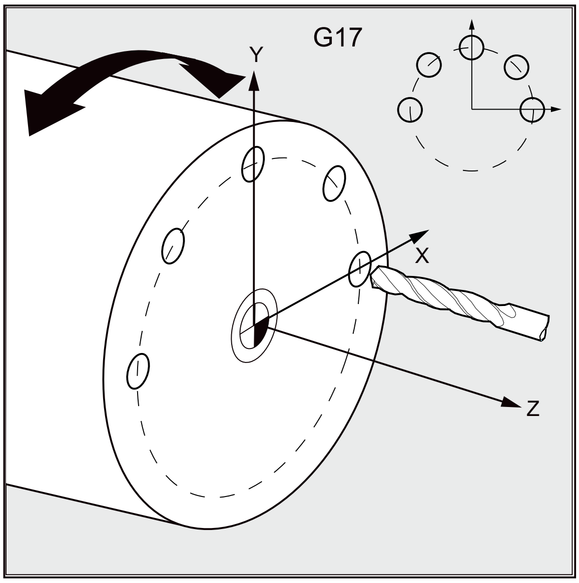


Figure 10-26 HOLES2

Sequence

In the cycle, the drilling positions are approached one after the other in the plane with G0.

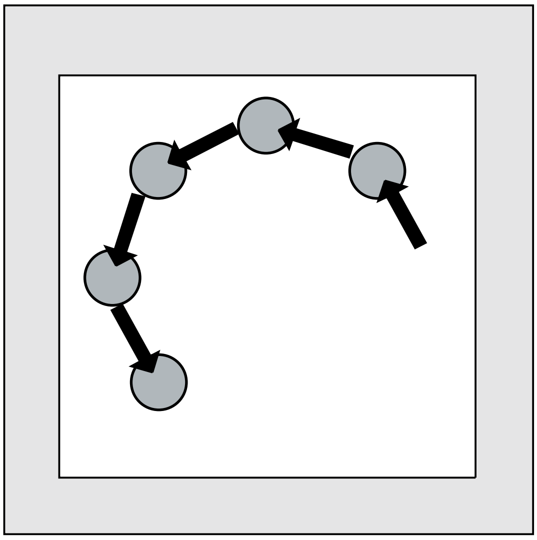


Figure 10-27 Sequence

Explanation of the parameters

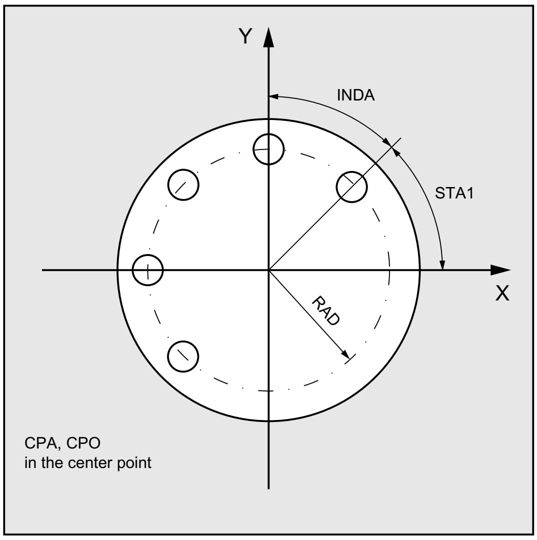


Figure 10-28 HOLES2 parameters

CPA, CPO and RAD (center point position and radius)

The position of the circle of holes in the machining plane is defined via center point (parameters CPA and CPO) and radius (parameter RAD). Only positive values are permitted for the radius.

STA1 and INDA (starting and incremental angle)

These parameters define the arrangement of the holes on the circle of holes.

Parameter STA1 defines the angle of rotation between the positive direction of the 1st axis (abscissa) in the workpiece coordinate system active before the cycle was called and the first hole. Parameter INDA contains the angle of rotation from one hole to the next.

If the INDA parameter is assigned the value zero, the indexing angle is calculated internally from the number of holes which are positioned equally in a circle.

NUM (number)

The NUM parameter defines the number of holes.

Programming example: Circle of holes

The program uses CYCLE82 to produce four holes on the end face of a turned part.

The final drilling depth of 30 mm is specified relative to the reference plane. The safety clearance in drilling axis Z is 2 mm. The circle has a radius of 42 mm. The starting angle is 33 degrees.

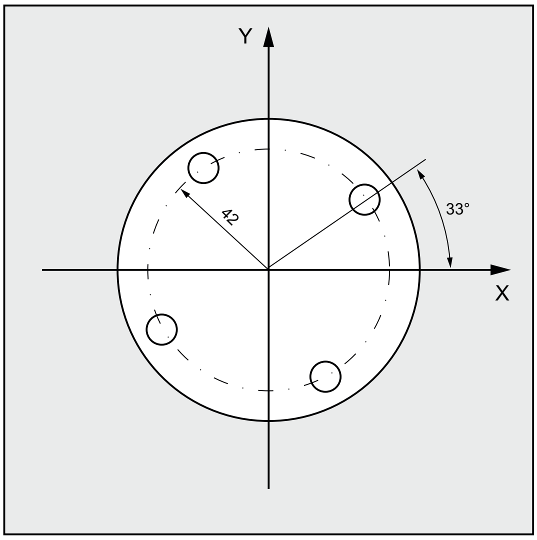


Figure 10-29 Example: Circle of holes

```
N10 G0 G90 X0 Z10 SPOS=0                 ; Approach starting position
N15 SETMS(2)                             ; Master spindle is now the
                                         milling spindle
N20 TRANSMIT                             ; Activate TRANSMIT function
N25 G17 G90 X-20 Y-10
N30 F30 S500 M3                          ; Specification of technology
                                         values
N35 T10 D1                               ; Change drilling machine
N40 M6
N45 MCALL CYCLE82(10, 0, 2, 0, 30, 1)    ; Modal call of drilling cycle
N50 HOLES2(0, 0, 42, 33, 0, 4)           ;Call circle-of-holes cycle
N85 MCALL                                ; Deselect modal call
N90 TRAFOOF                              ; Switch off TRANSMIT
N95 SETMS                                ; Master spindle is now the main
                                         spindle again
N60 M2                                   ; End of program
```

10.5 Turning cycles

10.5.1 Requirements

The turning cycles are part of the configuration file setup_T.cnf which is loaded into the user memory of the control system.

Call and return conditions

The G functions effective prior to the cycle call remain active beyond the cycle.

Plane definition

The machining plane must be defined prior to the cycle call. With turning, it is usually the G18 (ZX plane). The two axes of the current plane in turning are hereinafter referred to as the longitudinal axis (first axis of this plane) and transverse axis (second axis of this plane).

In the turning cycles, with diameter programming active, the second axis is taken into account as the transverse axis in all cases (see Programming Manual).

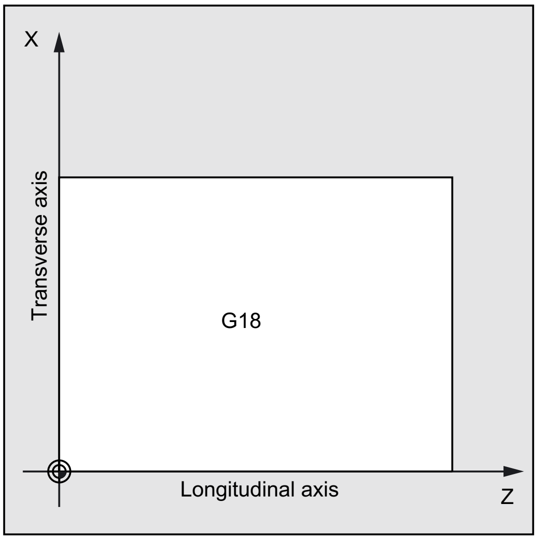


Figure 10-30 Plane definition

Contour monitoring relative to the clearance angle of the tool

Certain turning cycles in which traversing motions with relief cutting are generated monitor the clearance angle of the active tool for a possible contour violation. This angle is entered in the tool compensation as a value (in the D offset under the parameter DP24). A value between 1 and 90 degrees (0=no monitoring) without sign must be specified for the angle.

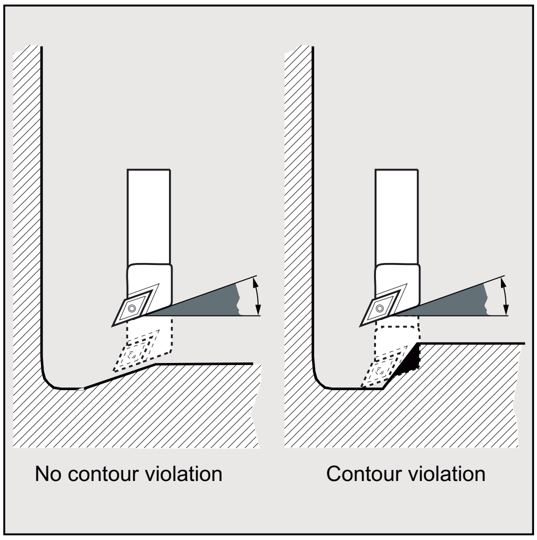


Figure 10-31 Longitudinal contour monitoring

When entering the tool clearance angle, note that this depends on the machining type 'longitudinal' or 'face'. If you want to use one tool for longitudinal and face machining, two tool compensations must be used in the case of different tool clearance angles.

The cycle will check whether or not the programmed contour can be machined using the selected tool.

If the machining is not possible using this tool, then

- the cycle will abort and an error message is output (in stock removal) or
- the contour is continued to be machined and a message is output (with undercut cycles). In this case, the contour is determined by the cutting edge geometry.

If the tool clearance angle is specified with zero in the tool compensation, this monitoring will not be performed. For details on the reactions, please refer to the individual cycles.

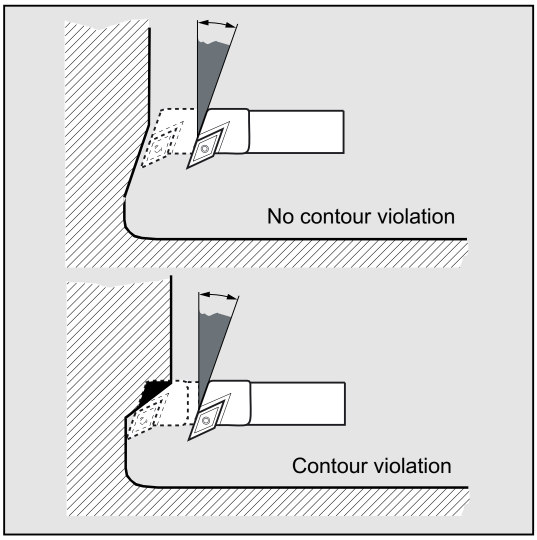


Figure 10-32 Planar contour monitoring

10.5.2 Groove - CYCLE93

Programming

CYCLE93(SPD, SPL, WIDG, DIAG, STA1, ANG1, ANG2, RCO1, RCO2, RCI1, RCI2, FAL1, FAL2, IDEP, DTB, VARI, VRT)

parameters

Table 10- 13 Parameters for CYCLE93

Parameter	Data type	Significance
SPD	REAL	Starting point in the transverse axis
SPL	REAL	Starting point in the longitudinal axis
WIDG	REAL	Groove width (enter without sign)
DIAG	REAL	Groove depth (enter without sign)
STA1	REAL	Angle between contour and longitudinal axis Range of values: 0<=STA1<=180 degrees
ANG1	REAL	Flank angle 1: on the side of the groove determined by the starting point (enter without sign) Range of values: 0<=ANG1<89.999 degrees
ANG2	REAL	Flank angle 2: on the other side (enter without sign) Range of values: 0<=ANG2<89.999
RCO1	REAL	Radius/chamfer 1, externally: on the side determined by the starting point
RCO2	REAL	Radius/chamfer 2, externally
RCI1	REAL	Radius/chamfer 1, internally: on the starting point side
RCI2	REAL	Radius/chamfer 2, internally
FAL1	REAL	Finishing allowance at the recess base
FAL2	REAL	Finishing allowance at the flanks
IDEP	REAL	Infeed depth (enter without sign)
DTB	REAL	Dwell time at recess base
VARI	INT	Machining type Range of values: 1...8 and 11...18
VRT	REAL	Variable retraction distance from contour, incremental (enter without sign)

Function

The grooving cycle can be used to carry out symmetrical and asymmetrical grooves for longitudinal and face machining at any straight contour elements. External and internal grooves can be produced.

Sequence

The infeed in the depth (towards the groove base) and in the width (from groove to groove) are calculated in the cycle internally and distributed equally with the maximum possible value.

When grooving at oblique faces, the tool will traverse from one groove to the next on the shortest path, i.e. parallel to the cone at which the groove is machined. During this process, a safety clearance to the contour is calculated internally in the cycle.

1. Step

Paraxial roughing down to the base of the groove in single infeed steps.

After each infeed, the tool is retracted for chip breaking.

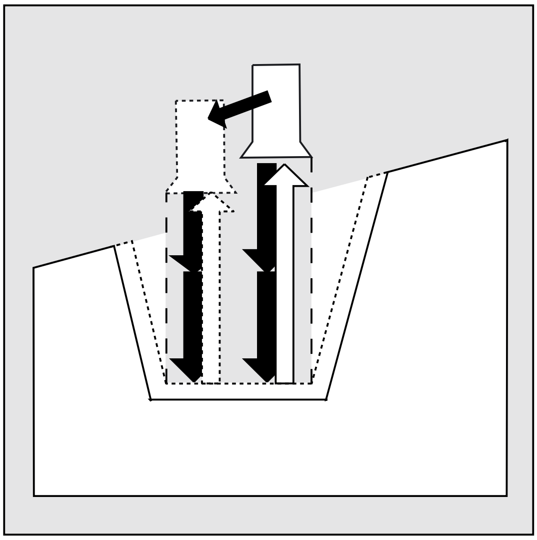


Figure 10-33 1. Step

2. Step

The groove is machined vertically to the infeed direction in one or several steps whereby each step, in turn, is divided according to the infeed depth. From the second cut along the groove width onwards, the tool will retract by 1 mm before each retraction.

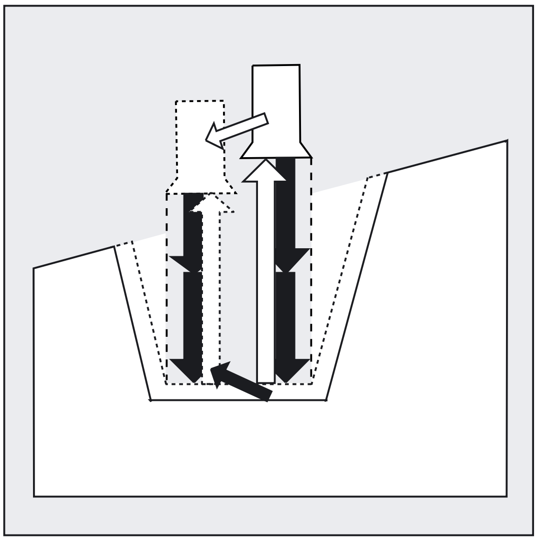


Figure 10-34 2. Step

3. Step

Machining of the flanks in one step if angles are programmed under ANG1 or ANG2. Infeed along the groove width is carried out in several steps if the flank width is larger.

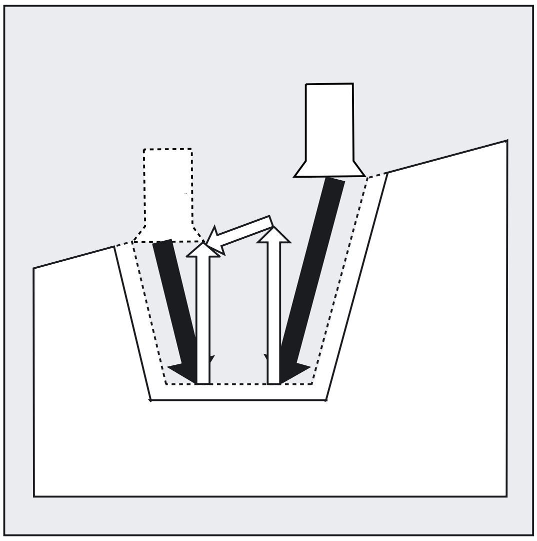


Figure 10-35 3. Step

4. Step

Stock removal of the finishing allowance parallel to the contour from the edge to the groove center. During this operation, the tool radius compensation is selected and deselected by the cycle automatically.

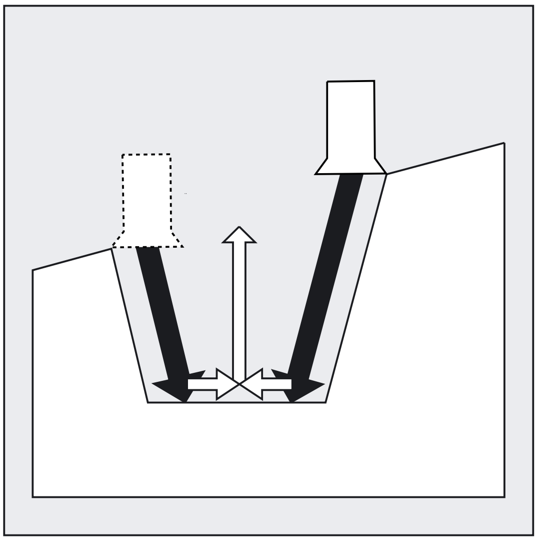


Figure 10-36 4. Step

Explanation of parameters: SPD and SPL (starting point)

These coordinates can be used to define the starting point of a groove starting from which the form is calculated in the cycle. The cycle determines its own starting point. For an external groove, movement begins in the direction of the longitudinal axis, for an internal groove in the direction of the facing axis.

Grooves at bent contour elements can be realized differently. Depending on the form and radius of the bend, either a paraxial straight line can be laid over the maximum of the bend or a tangential oblique line can be created in a point of the edge points of the groove.

Radii and chamfers at the groove edge make sense with bent contours only if the appropriate edge point is on the straight line specified for the cycle.

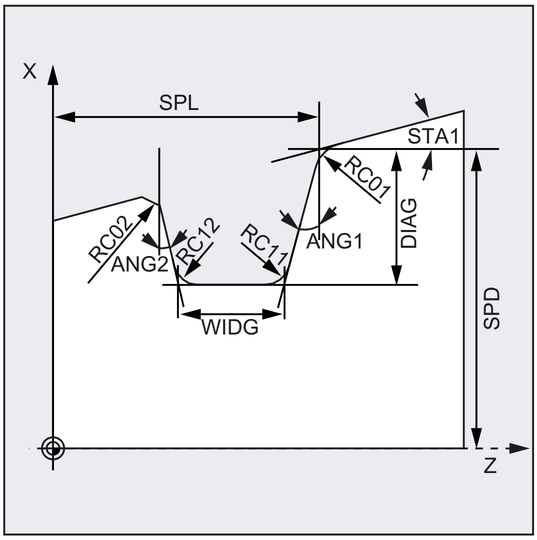


Figure 10-37 Parameter CYCLE93 longitudinal

WIDG and DIAG (groove width and groove depth)

The parameters groove width (WIDG) and groove depth (DIAG) are used to define the form of the groove. In its calculation, the cycle always assumes the point programmed under SPD and SPL.

If the groove width is larger than that of the active tool, the width is removed in several steps. When doing so, the whole width is distributed by the cycle equally. The maximum infeed is 95% of the tool width after deduction of the cutting edge radii. This provides a cutting overlap.

If the programmed groove width is smaller than the real tool width, the error message 61602 "Tool width defined incorrectly" and machining is aborted. The alarm will also appear if a cutting edge width equal to zero is detected in the cycle.

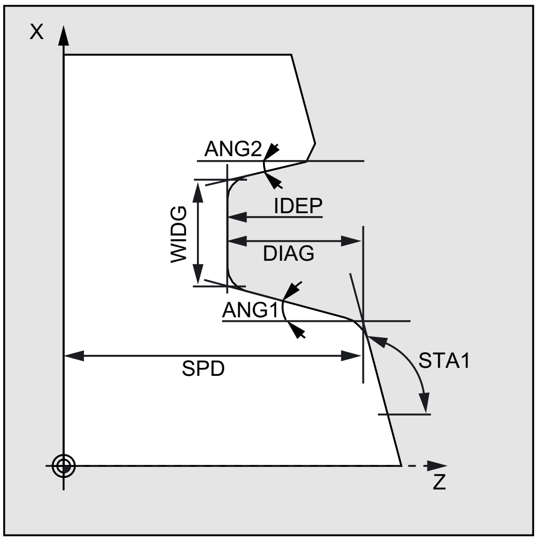


Figure 10-38 Parameter CYCLE93 planar

STA1 (angle)

Use the STA1 parameter to program the angle of the oblique line at which the groove is to be machined. The angle can assume values between 0 and 180 degrees and always refers to the longitudinal axis.

ANG1 and ANG2 (flank angle)

Asymmetric grooves can be described by flank angles specified separately. The angles can assume values between 0 and 89.999 degrees.

RCO1, RCO2 and RCI1, RCI2 (radius/chamfer)

The form of the groove can be modified by entering radii/chamfers at the margin or at the base. **It is imperative to enter the radii with positive sign, and the chamfers with negative sign.**

How the programmed chamfers are taken into account is specified in dependence of the tens digit of the VARI parameter.

- With VARI<10 (tens=0) Chamfers with CHF=...
- With VARI>10 chamfers programmed with CHR

(CHF/CHR, see section "List of instructions")

FAL1 and FAL2 (finishing allowance)

It is possible to program separate finishing allowances for groove base and flanks. During roughing, stock removal is carried out up to these finishing allowances. The same tool is then used to machine a contour-parallel cut along the final contour.

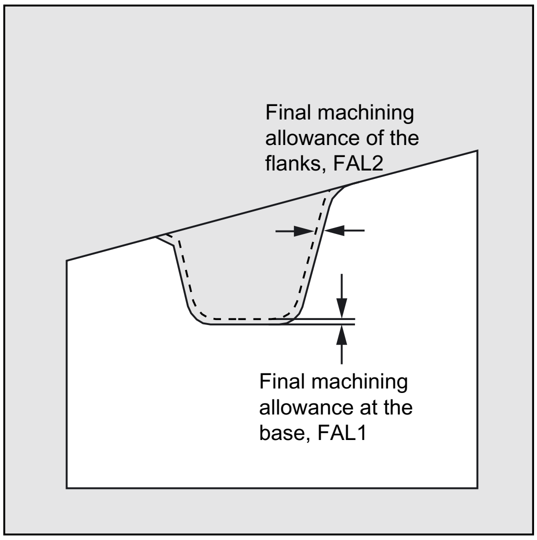

Final machining allowance of the flanks, FAL2
Final machining allowance at the base, FAL1

Figure 10-39 Finishing allowance

IDEP (infeed depth)

You can divide the paraxial grooving into several depth infeeds by programming an infeed depth. After each infeed, the tool is retracted by 1 mm for chip breaking.

The IDEP parameter must be programmed in all cases.

DTB (dwell time)

The dwell time at the groove base should be selected such that at least one spindle revolution is carried out. It is programmed in seconds.

VARI (machining type)

The machining type of the groove is defined with the units digit of the VARI parameter. It can assume the values indicated in the illustration.

The tens digit of parameter VARI determines how the chamfers are taken into account.

VARI 1...8: Chamfers are calculated as CHF

VARI 11...18: Chamfers are calculated as CHR

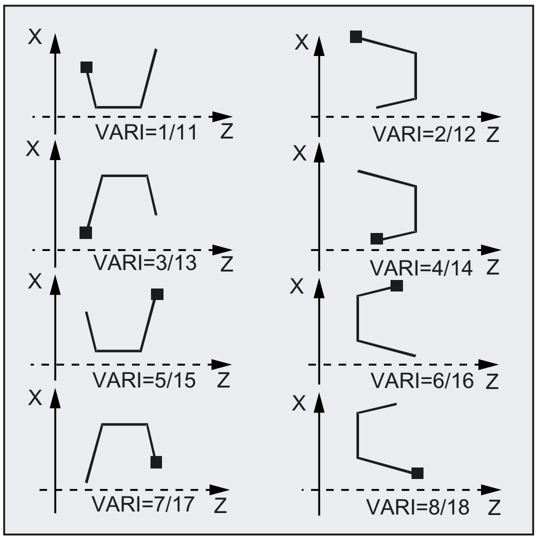


Figure 10-40 Machining methods

If the parameter has a different value, the cycle will abort with alarm 61002 "Machining type defined incorrectly".

The cycle carries out a contour monitoring such that a reasonable groove contour results. This is not the case if the radii/chamfers come into contact or intersect at the groove base or if you try to carry out a face grooving operation at a contour segment located parallel to the longitudinal axis. In such cases, the cycle will abort with alarm 61603 "Groove form defined incorrectly".

VRT (variable retraction path)

The retraction path can be programmed in the _VRT parameter on the basis of the outside or inside diameter of the groove.

For VRT=0 (parameter not programmed), the tool is retracted by 1 mm. The retraction path is always measured according to the programmed system of units, inch or metric.

The same retraction path is also used for chip breaking after each depth infeed into the groove.

Note

Before calling the grooving cycle, a double-edged tool must be enabled. The offset values for the two cutting edges must be stored in two successive D numbers of the tool whereby the first of which must be activated prior to the first cycle call. The cycle itself defines for which machining step it will use which of the two tool compensation values and will also enable them automatically. After completion of the cycle, the tool compensation number programmed prior to the cycle call is active again. If no D number is programmed for a tool compensation when the cycle is called, the execution of the cycle is aborted with the alarm 61000 "No tool compensation active".

Programming example: Plunge-cutting

This program is used to produce a groove externally at an oblique line in the longitudinal direction.

The starting point is on the right-hand side at X35 Z60.

The cycle will use the tool compensations D1 and D2 of tool T5. The cutting tool must be defined accordingly.

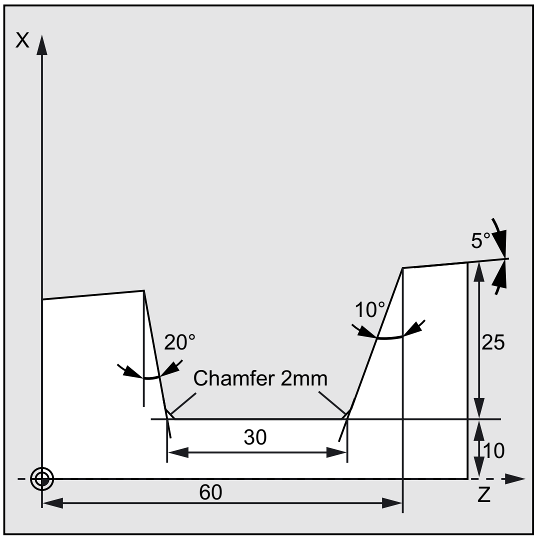


Figure 10-41 Programming example: Plunge-cutting

```
N10 G0 G90 Z65 X50 T5 D1 S400 M3                 ;Starting point before the
                                                 beginning of the cycle
N20 G95 F0.2                                     ; Specification of technology
                                                 values
N30 CYCLE93(35, 60, 30, 25, 5, 10, 20, 0, 0, -2, ; Cycle call
-2, 1, 1, 10, 1, 5,0.2)                          Retraction distance of 0.2 mm
                                                 programmed
N40 G0 G90 X50 Z65                               ;Next position
N50 M02                                          ; End of program
```

10.5.3 Undercut (forms E and F to DIN) - CYCLE94

Programming

CYCLE94(SPD, SPL, FORM, VARI)

Parameter

Table 10- 14 Parameters for CYCLE94

Parameter	Data type	Significance
SPD	REAL	Starting point in the transversal axis (enter without sign)
SPL	REAL	Starting point of the tool compensation in the longitudinal axis (enter without sign)
FORM	CHAR	Definition of the form Values: E (for form E), F (for form F)
VARI	INT	Specification of undercut position Values: 0 (according to tool cutting edge position), 1...4 (define position)

Function

This cycle can be used to perform undercuts to DIN509 of forms E and F with standard requirements at a finished diameter of >3 mm.

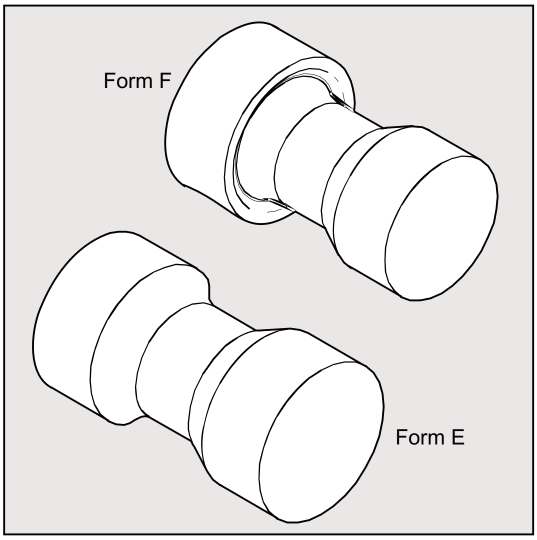


Figure 10-42 Form F and form E

Sequence

Position reached prior to cycle start:

The starting position can be any position from which the undercut can be approached without collision.

The cycle creates the following sequence of motions:

- Approach of the starting point determined in the cycle by using G0
- Selection of the cutter radius compensation according to the active tool point direction and traveling along the undercut contour at the feedrate programmed prior to the cycle call
- Retraction to the starting point with G0 and deselection of the cutter radius compensation with G40

Explanation of parameters: SPD and SPL (starting point)

Use the parameter SPD to specify the finished part diameter for the undercut. The SPL parameter defines the finished dimension in the longitudinal axis.

If a final diameter of <3 mm results for the value programmed for SPD, the cycle is canceled, and alarm 61601 "Finished part diameter too small" is issued.

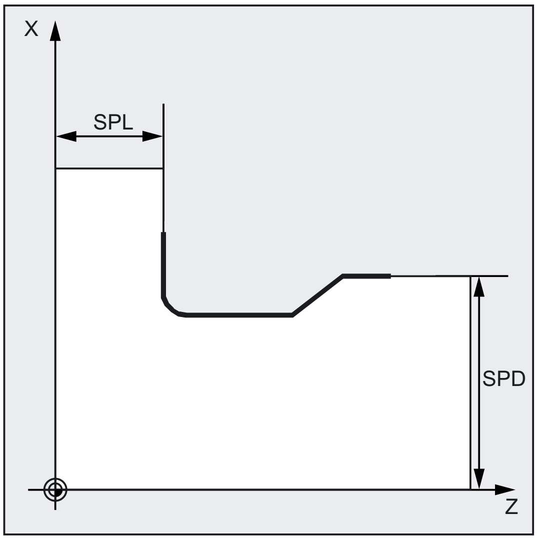


Figure 10-43 Parameters for CYCLE94

FORM (definition)

Form E and form F are fixed in DIN509 and must be defined using this parameter.

If the parameter has a value other than E or F, the cycle aborts and creates alarm 61609 "Form defined incorrectly".

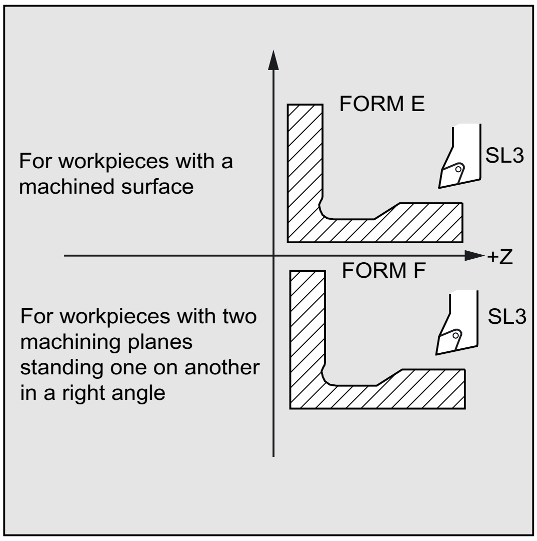

FORM E
SL3
For workpieces with a machined surface
+Z
FORM F
SL3
For workpieces with two machining planes standing one on another in a right angle

Figure 10-44 Parameters for FORM

VARI (undercut position)

The position of the undercut can be either specified directly or derived from the tool point direction with the _VARI parameter.

VARI=0: According to tool point direction

The tool point direction is determined by the cycle automatically from the active tool compensation. The cycle can operate with the tool point directions 1 ... 4.

If the cycle detects any of the tool point directions 5 ... 9, the alarm 61608 "Wrong tool point direction programmed" and the cycle is aborted.

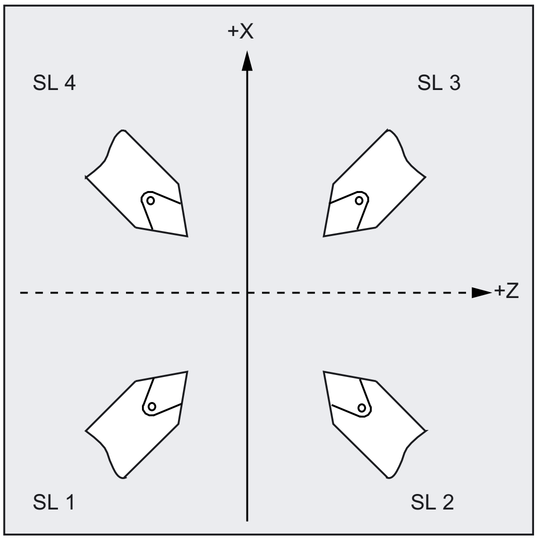


Figure 10-45 Cutting edge position

VARI=0

VARI=1...4: Definition of undercut position

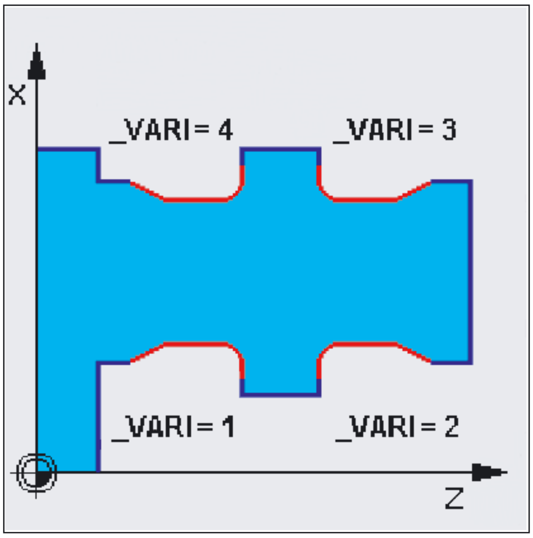


Figure 10-46 Undercut position

VARI=1...4

For VARI<>0, the following applies:

- The actual cutting edge position is not checked, i.e., all positions can be used if technologically suitable.

The clearance angle of the active tool is monitored in the cycle if an appropriate value is specified in the appropriate parameter of the tool compensation. If it turns out that the form of the undercut cannot be machined using the selected tool since its tool clearance angle is too small, the message "Changed form of undercut" is displayed on the control system. The machining, however, is continued.

The cycle determines its starting point automatically. This is by 2 mm away from the end diameter and by 10 mm away from the finishing dimension in the longitudinal axis. The position of this starting point referred to the programmed coordinate values is determined by the tool point direction of the active tool.

Note

Before calling the cycle, a tool compensation must be activated. Otherwise, the cycle is aborted after alarm 61000 "No tool compensation active" has been output.

Programming example: Undercut form E

This program can be used to program an undercut of form E.

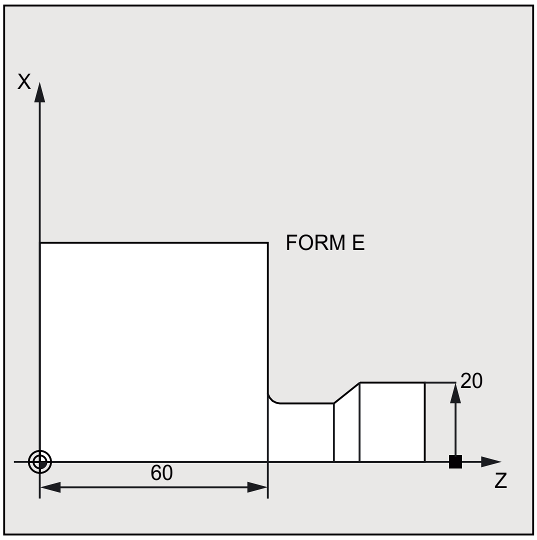


Figure 10-47 Programming example: Undercut form E

```
N10 T1 D1 S300 M3 G95 F0.3                ; Specification of technology
                                          values
N20 G0 G90 Z100 X50                       ; Selection of starting position
N30 CYCLE94(20, 60, "E")                  ; Cycle call
N40 G90 G0 Z100 X50                       ; Approach next position
N50 M02                                   ; End of program
```

10.5.4 Cutting with relief cut – CYCLE95

Programming

CYCLE95(NPP, MID, FALZ, FALX, FAL, FF1, FF2, FF3, VARI, DT, DAM, VRT)

parameters

Table 10- 15 Parameters for CYCLE95

Parameter	Data type	Significance
NPP	STRING	Name of contour subroutine
MID	REAL	Infeed depth (enter without sign)
FALZ	REAL	Finishing allowance in the longitudinal axis (enter without sign)
FALX	REAL	Finishing allowance in the transverse axis (enter without sign)
FAL	REAL	Finishing allowance according to the contour (enter without sign)
FF1	REAL	Feedrate for roughing without undercut
FF2	REAL	Feedrate for insertion into relief cut elements
FF3	REAL	Feedrate for finishing
VARI	REAL	Machining type Range of values: 1 ... 12
TT	REAL	Dwell time fore chip breaking when roughing
DAM	REAL	Path length after which each roughing step is interrupted for chip breaking
VRT	REAL	Lift-off distance from contour when roughing, incremental (to be entered without sign)

Function

Using the rough turning cycle, you can produce a contour, which has been programmed in a subroutine, from a blank by paraxial stock removal. The contour may contain relief cut elements. It is possible to machine contours using longitudinal and face machining, both externally and internally. The technology can be freely selected (roughing, finishing, complete machining). When roughing the contour, paraxial cuts from the maximum programmed infeed depth are programmed and burrs are also removed parallel to the contour after an intersection point with the contour has been reached. Roughing is performed up to the final machining allowance programmed.

Finishing is performed in the same direction as roughing. The tool radius compensation is selected and deselected by the cycle automatically.

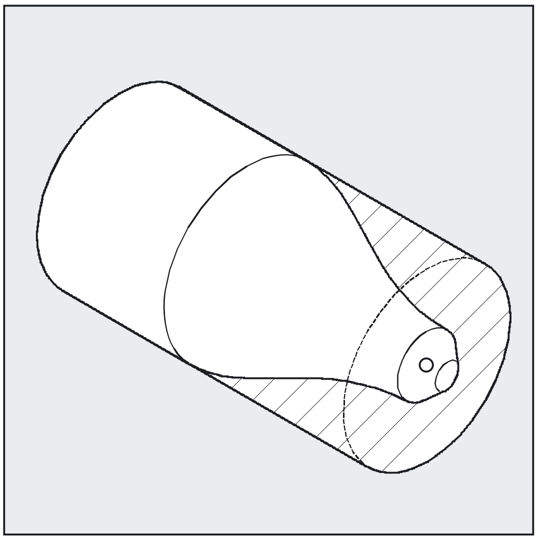


Figure 10-48 Stock removal cycle CYCLE95

Sequence

Position reached prior to cycle start:

The starting position is any position from which the contour starting point can be approached without collision.

The cycle creates the following sequence of motions:

The cycle starting point is calculated internally and approached with G0 in both axes at the same time.

Roughing without relief cut elements:

- The paraxial infeed to the current depth is calculated internally and approached with G0.
- Approach of paraxial roughing intersection point with G1 and at feedrate FF1.
- Rounding parallel to the contour along the contour + finishing allowance with G1/G2/G3 and FF1.
- Lift-off by the amount programmed under _VRT in each axis and retraction with G0.

- This sequence is repeated until the total depth of the machining step is reached.
- When roughing without relief cut elements, retraction to the cycle starting point is carried out axis by axis.

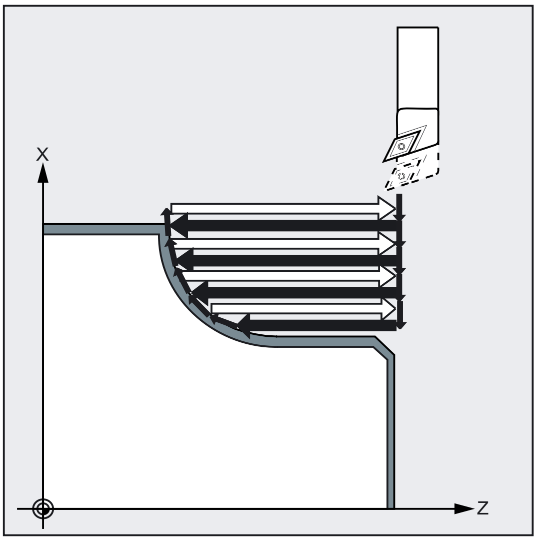


Figure 10-49 Roughing without relief cut elements

Roughing the relief cut elements:

- Approach of the starting point for the next relief cut axis by axis with G0 When doing so, an additional cycle-internal safety clearance is observed.
- Infeed along the contour + finishing allowance with G1/G2/G3 and FF2.
- Approach of paraxial roughing intersection point with G1 and at feedrate FF1.
- Rounding along the contour, retraction and return are carried out as with the first machining step.
- If there are further relief cut elements, this sequence is repeated for each relief cut.

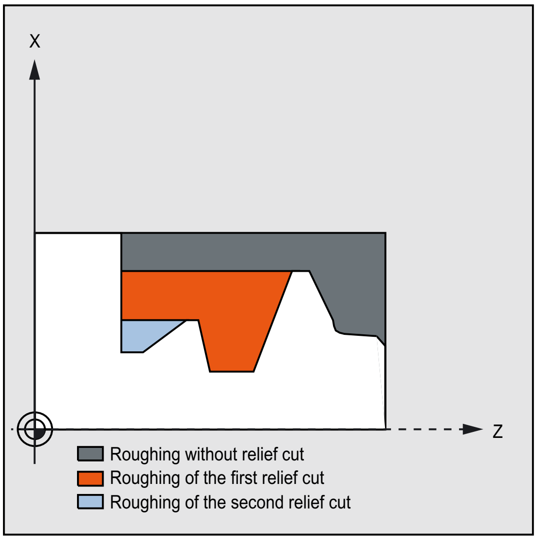

X
Z
Roughing without relief cut
Roughing of the first relief cut
Roughing of the second relief cut

Figure 10-50 Roughing with relief cut elements

Finishing:

- The cycle starting point is approached axis by axis with G0.
- The contour starting point is approached with G0 in both axes at the same time.
- Finishing along the contour with G1/G2/G3 and FF3
- Retraction to the starting point with both axes and G0

Explanation of parameters: NPP (name)

This parameter is used to specify the contour name.

1. The contour can be defined as a subroutine:

   NPP=name of subroutine

   The name of the contour subroutine is subject to all name conventions described in the Programming Manual.

   Input:

   – The subroutine already exists --> enter name, continue

   – The subroutine does not yet exist --> enter name and select the "New file" softkey. A program (main program) with the entered name is created and the program will jump to the contour editor.

   Use the "Technol. mask" softkey to confirm your input and return to the cycle help screen form.

2. The contour can also be a section of the calling program:

   NPP=name of the starting label: name of end label

   Input:

   – Contour is already described --> name of starting label: Name of the end label

   – Contour is not yet described --> enter name of starting label and press softkey "Contour append".

     Starting and end labels are automatically created from the name you have entered; then the program will jump to the contour editor.

   Use the "Technol. mask" softkey to confirm your input and return to the cycle help screen form.

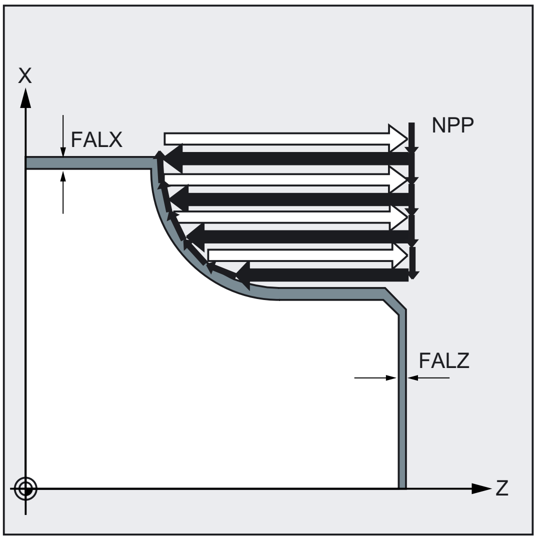


Figure 10-51 Parameters

Examples:

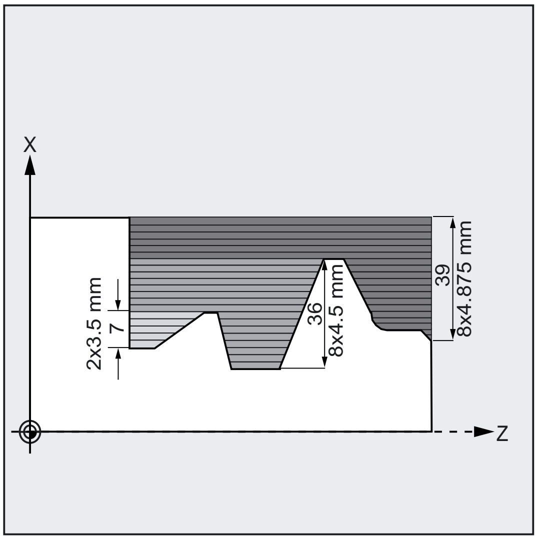
```
NPP=KONTUR_1                                        ;The rough turning contour is the
                                                    complete program Kontur_1.
NPP=ANFANG:ENDE                                     ;The rough turning contour is
                                                    defined as a section in the
                                                    calling program, which starts
                                                    from the block containing label
                                                    ANFANG to the block containing
                                                    label ENDE.
```

MID (infeed depth)

The MID parameter is used to define the maximum possible infeed depth for the roughing process.

The cycle will automatically calculate the current infeed depth used for roughing.

With contours containing relief cut elements, the roughing process is divided by the cycle into individual roughing sections. The cycle calculates a new current infeed depth for each roughing section. This infeed depth is always between the programmed infeed depth and the half of its value. The number of required roughing steps is determined on the basis of the total depth of a roughing section and of the programmed maximum infeed depth to which the total depth to be machined is distributed equally. This provides optimum cutting conditions. For roughing this contour, the machining steps shown in the illustration result.



Example of calculating the current infeed depth:

Machining section 1 has a total depth of 39 mm. If the maximum infeed depth is 5 mm, eight roughing cuts are required. These are carried out with an infeed of 4.875 mm.

In machining step 2, 8 roughing steps, too, are carried out with an infeed of 4.5 mm each (total difference 36 mm).

In machining step 3, two roughing passes are carried out with a current infeed of 3.5 (total difference 7 mm).

FAL, FALZ and FALX (finishing allowance)

A finishing allowance for roughing can be specified either using the parameters FALZ and FALX if you want to specify different finishing allowances axis-specifically or via the parameter FAL for a finishing allowance that follows the contour. In this case, this value is taken into account in both axes as a finishing allowance.

No plausibility check is carried out for the programmed values. In other words: If all three parameters are assigned values, all these finishing allowances are taken into account by the cycle. It is, however, reasonable to decide either on the one or other form of definition of a finishing allowance.

Roughing is always carried out up to these finishing allowances. The resulting residual corner is also removed parallel to the contour after each paraxial roughing process immediately so that no additional residual corner cut is required after completion of roughing. If no finishing allowances are programmed, stock is removed when roughing up to the final contour.

FF1, FF2 and FF3 (feedrate)

It is possible to specify different feedrates for the individual machining steps, as shown in Figure NO TAG.

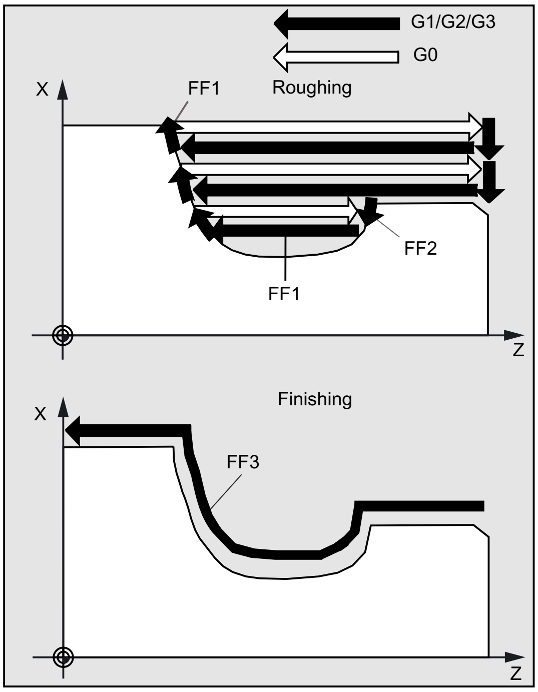


Figure 10-52 Feed parameter

VARI (machining type)

Table 10- 16 Type of machining

Value	Longitudinal/face	Ext./int.	Roughing/finishing/complete
1	L	O	Roughing
2	P	O	Roughing
3	L	I	Roughing
4	P	I	Roughing
5	L	O	Finishing
6	P	O	Finishing
7	L	I	Finishing
8	P	I	Finishing
9	L	O	Complete machining
10	P	O	Complete machining
11	L	I	Complete machining
12	P	I	Complete machining

In longitudinal machining, the infeed is always carried out along the transversal axis, and in face machining - along the longitudinal axis.

External machining means that the infeed is carried out in the direction of the negative axis. With internal machining, the infeed is carried out in the direction of the positive axis.

The VARI parameter is subjected to a plausibility check. If its value is not in the range 1 ... 12 when the cycle is called, the cycle is aborted with alarm 61002 "Machining type defined incorrectly".

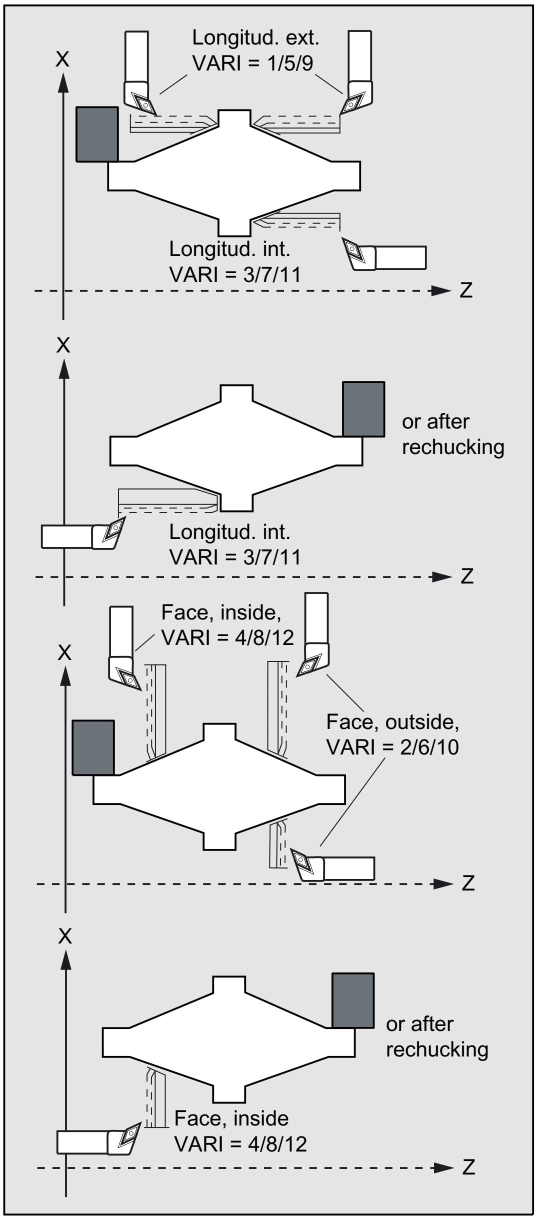


Figure 10-53 Machining type

DT and DAM (dwell time and path length)

These parameters can be used to achieve an interruption of the individual roughing steps after certain distances traversed in order to carry out chip breaking. These parameters are only relevant for roughing. The parameter DAM is used to define the maximum distance after which chip breaking is to be carried out. In DT, an appropriate dwell time (in seconds) can be programmed which is carried out at each of the cut interruption points. If no distance is specified for the cut interruption (DAM=0), uninterrupted roughing steps without dwell times are created.

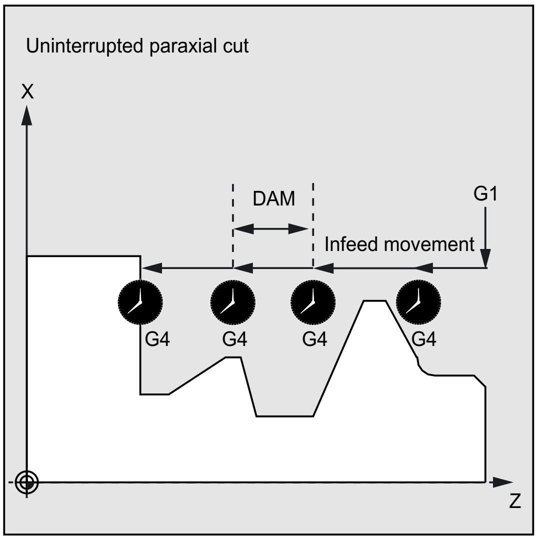


Figure 10-54 Dwell time and distance traveled

VRT (lift-off distance)

Parameter VRT can be used to program the amount by which the tool is retracted in both axes when roughing.

For VRT=0 (parameter not programmed), the tool will retract by 1 mm.

Contour definition

The contour must contain at least 3 blocks with motions in the two axes of the machining plane.

If the contour program is shorter, the cycle is aborted after the alarms 10933 "Number of contour blocks contained in the contour program not sufficient" and 61606 "Error in contour preparation" have been output.

Relief cut elements can be connected directly one after the other. Blocks without motions in the plane can be written without restrictions.

In the cycle, all traversing blocks are prepared for the first two axes of the current plane since only these are involved in the cutting process. The contour program may contain any motions programmed for other axes; their distances to be traversed, however, will not come into effect during the whole cycle.

Only straight line and circle programming with G0, G1, G2 and G3 are permitted as the geometry in the contour. Furthermore, it is also possible to program the commands for rounding and chamfer. If any other motion commands are programmed in the contour, the cycle is aborted with the alarm 10930 "Illegal type of interpolation in the stock removal contour".

The first block with a traversing motion in the current machining plane must contain a motion command G0, G1, G2 or G3; otherwise, the cycle is canceled, and alarm 15800 "Incorrect prerequisites for CONTPRON" is issued. This alarm is also issued if G41/42 is active. The starting point of the contour is the first programmed position in the machining plane.

To machine the programmed contour, a cycle-internal memory is prepared which can accommodate a certain maximum number of contour elements; how many, depends on the contour. If a contour contains too many contour elements, the cycle is canceled, and alarm 10934 "Contour table overflow" is issued. In this case, the contour must be split over several contour sections, and the cycle for each section must be called separately.

If the maximum diameter is not at the programmed end or starting point of the contour, the cycle will automatically add an axis-parallel straight line to complete the contour maximum, and this part is removed as the undercut.

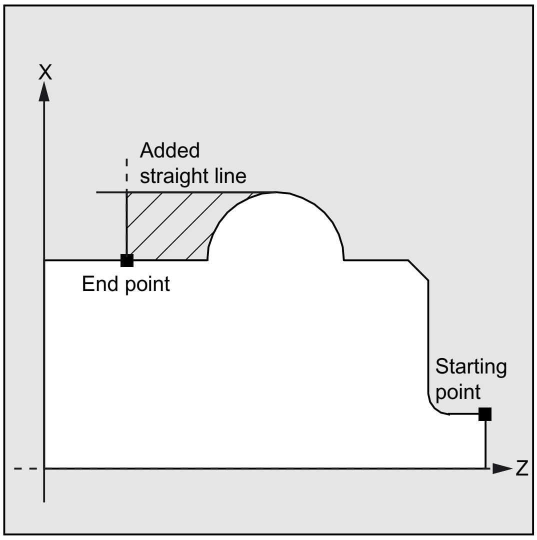


Figure 10-55 Contour definition

If a tool radius compensation is programmed in the contour subroutine with G41/G42, the cycle is canceled, and alarm 10931 "Faulty cutting contour" is issued.

contour direction

The direction in which the stock removal contour is programmed can be freely selected. In the cycle, the machining direction is defined automatically. In complete machining, the contour is finished in the same direction as machining was carried out when roughing.

When deciding on the machining direction, the first and the last programmed contour points are taken into account. Therefore, both coordinates must always be programmed in the first block of the contour subroutine.

Contour monitoring

The cycle provides contour monitoring with regard to the following:

- Clearance angle of the active tool
- Circular programming of arcs with an arc angle > 180 degrees

With relief cut elements, the cycle checks whether the machining is possible using the active tool. If the cycle detects that this machining will result in a contour violation, it will be aborted after alarm 61604 "Active tool violates programmed contour" has been output.

If the tool clearance angle is specified with zero in the tool compensation, this monitoring will not be performed.

If too large arcs are found in the compensation, alarm 10931 "Incorrect machining contour" appears.

Overhanging contours cannot be machined by CYCLE95. Contours of this type are not monitored by the cycle and consequently there is no alarm.

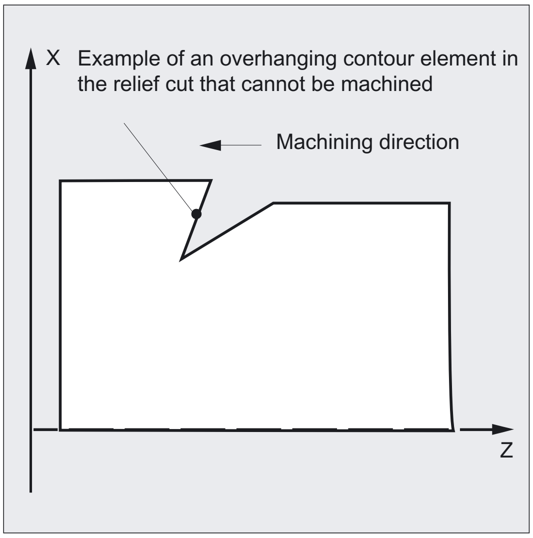

X
Example of an overhanging contour element in the relief cut that cannot be machined
Machining direction
Z

Figure 10-56 CYCLE95 contour monitoring, overhanging contours

Starting point

The cycle determines the starting point for the machining operation automatically. The starting point is located in the axis in which the depth infeed is carried out, shifted from the contour by the amount of the finishing allowance + lift-off distance (parameter _VRT). In the other axis, it is by finishing allowance + _VRT ahead of the contour starting point.

When the starting point is approached, the cutter radius compensation is selected internally in the cycle.

The last point before the cycle is called must therefore be selected such that this approach is possible without collision and space enough is provided to carry out the appropriate compensatory motion.

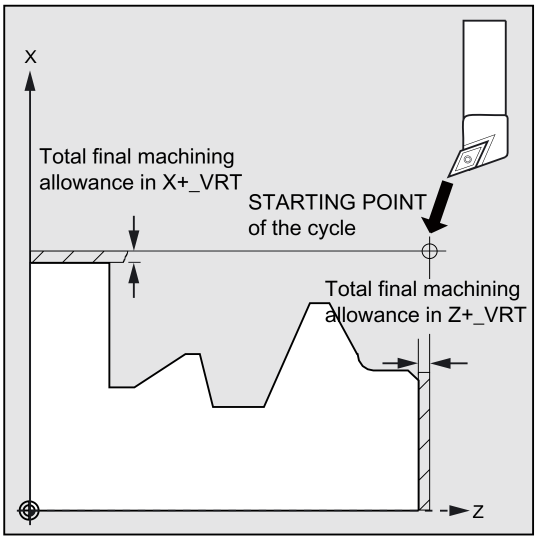


Figure 10-57 Starting point

Cycle approach strategy

In roughing, the starting point determined by the cycle is always approached with both axes simultaneously, and in finishing, axis by axis. In finishing, the infeed axis traverses first.

Programming example 1: Stock removal cycle

The contour shown in the illustration to explain the defining parameters is to be machined longitudinally externally by complete machining. Axis-specific finishing allowances are specified. Cutting will not be interrupted when roughing. The maximum infeed is 5 mm.

The contour is stored in a separate program.

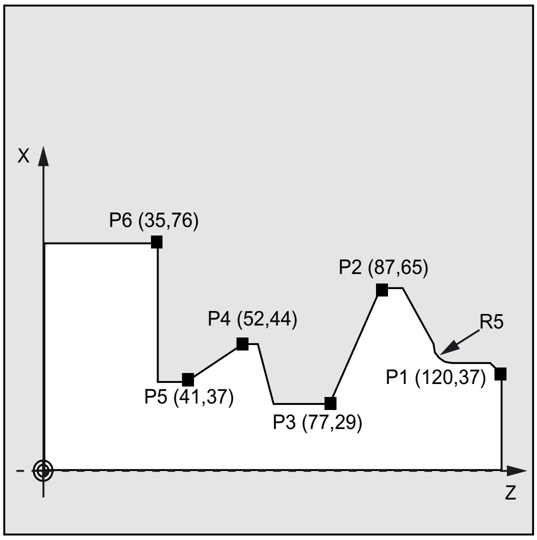


Figure 10-58 Programming example 1: Stock removal cycle

```
N10 T1 D1 G0 G95 S500 M3 Z125 X81               ; Approach position before cycle
                                                call
N20 CYCLE95("KONTUR_1", 5, 1.2, 0.6, , 0.2, 0.1, ; Cycle call
0.2, 9, , , 0.5)
N30 G0 G90 X81                                  ;Reapproach starting position
N40 Z125                                        ; Traverse axis by axis
N50 M2                                          ; End of program
%_N_KONTUR_1_SPF                                ;Beginning of contour subroutine
N100 Z120 X37                                   ; Traverse axis by axis
N110 Z117 X40
N120 Z112 RND=5                                 ;Rounding with radius 5
N130 Z95 X65                                    ; Traverse axis by axis
N140 Z87
N150 Z77 X29
N160 Z62
N170 Z58 X44
N180 Z52
N190 Z41 X37
N200 Z35
N210 X76
N220 M02                                        ; End of subroutine
```

Programming example 2: Stock removal cycle

The stock removal contour is defined in the calling program and is traversed directly after the cycle for finishing has been called.

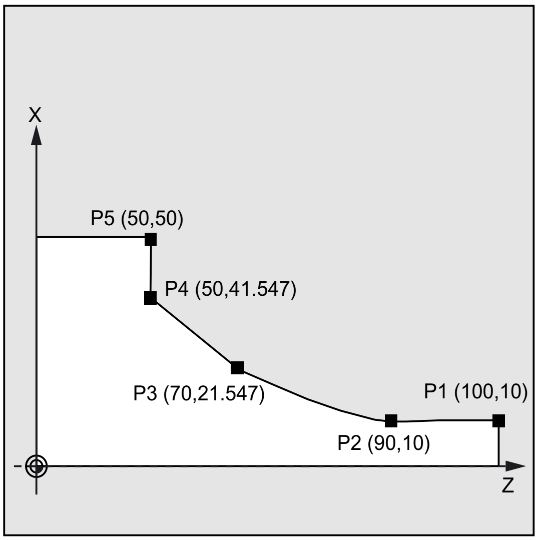


Figure 10-59 Programming example 2: Stock removal cycle

```
N110 G18 DIAMOF G90 G96 F0.8
```

```
N120 S500 M3
N130 T1 D1
N140 G0 X70
N150 Z160
N160 CYCLE95("ANFANG:ENDE",2.5,0.8,                ; Cycle call
0.8,0,0.8,0.75,0.6,1, , , )
N170 G0 X70 Z160
N175 M02
START:
N180 G1 X10 Z100 F0.6
N190 Z90
N200 Z70 ANG=150
N210 Z50 ANG=135
N220 Z50 X50
END:
N230 M02
```

10.5.5 Thread undercut - CYCLE96

Programming

CYCLE96 (DIATH, SPL, FORM, VARI)

Parameter

Table 10- 17 Parameters for CYCLE94

Parameter	Data type	Significance
DIATH	REAL	Nominal diameter of the thread
SPL	REAL	Starting point of the correction in the longitudinal axis
FORM	CHAR	Definition of the form Values: A (for form A), B (for form B), C (for form C), D (for form D)
VARI	INT	Specification of undercut position Values: 0: According to tool point direction 1...4: Define position

Function

You can use this cycle to perform thread undercuts to DIN76 for parts with metrical ISO thread.

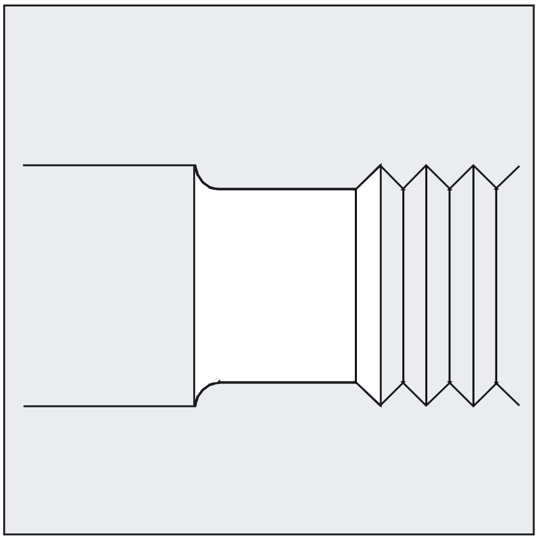


Figure 10-60 Thread undercut

Sequence

Position reached prior to cycle start:

The starting position can be any position from which each thread undercut can be approached without collision.

The cycle creates the following sequence of motions:

- Approach of the starting point determined in the cycle by using G0
- Selection of the tool radius compensation according to the active tool point direction. Traversing along the undercut contour using the feedrate programmed before the cycle was called
- Retraction to the starting point with G0 and deselection of the tool radius compensation with G40

Explanation of parameters: DIATH (nominal diameter)

Use this cycle to perform thread undercuts for metric threads from M3 through M68.

If the value programmed in DIATH results in a final diameter of <3 mm, the cycle is aborted and alarm:

61601 "Finished part diameter too small" is issued.

If the parameter has a value other than specified in DIN76 Part 1, the cycle is also canceled, generating the alarm:

61001 "Thread lead defined incorrectly".

SPL (starting point)

The finished dimension in the longitudinal axis is defined using the parameter SPL.

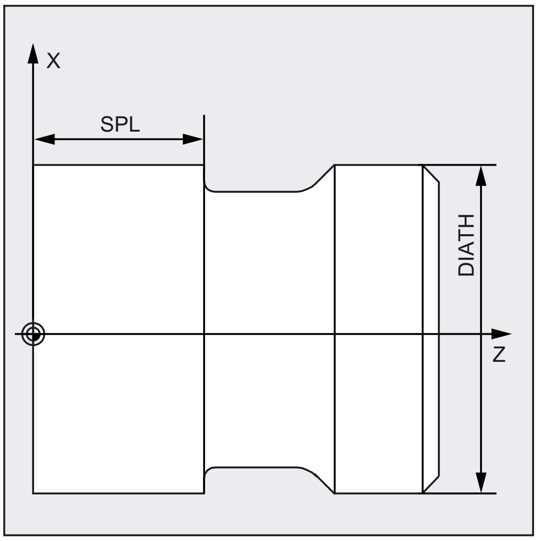


Figure 10-61 Parameter CYCLE96

FORM (definition)

Thread undercuts of the forms A and B are defined for external threads, form A for standard run-outs of threads, and form B for short run-outs of threads.

Thread undercuts of the forms C and D are used for internal threads, form C for a standard run-out of the thread, and form D for a short run-out.

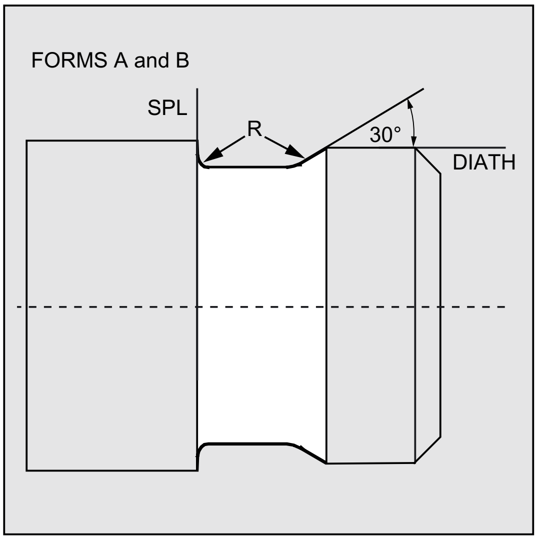


Figure 10-62 FORMS A and B

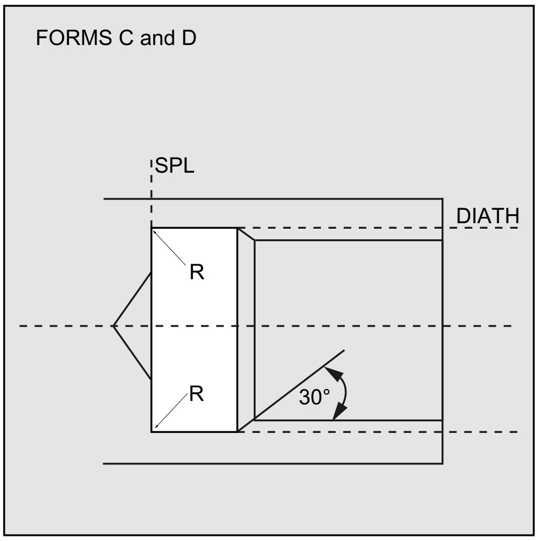


Figure 10-63 FORMS C and D

If the parameter has a value other than A ... D, the cycle aborts and creates alarm 61609 "Form defined incorrectly".

Internally in the cycle, the tool radius compensation is selected automatically.

The cycle only uses the tool point directions 1 ... 4. If the cycle detects a tool point direction 5 ... 9 or if the form of the undercut cannot be machined with the selected tool point direction, alarm 61608 "Wrong tool point direction programmed" is issued, and the cycle is canceled.

VARI (undercut position)

The position of the undercut can be either specified directly or derived from the tool point direction with the _VARI parameter. See _VARI for CYCLE94.

The cycle will find the starting point determined by the tool point direction of the active tool and the thread diameter automatically. The position of this starting point referred to the programmed coordinate values is determined by the tool point direction of the active tool.

For the forms A and B, the undercut angle of the active tool is monitored in the cycle. If it is detected that the form of the undercut cannot be machined using the selected tool, the message "Changed form of undercut" is displayed on the control system; the machining, however, is continued.

Note

Before calling the cycle, a tool compensation must be activated. Otherwise, the cycle is terminated and the error message 61000 "No tool compensation active" is issued.

Programming example: Thread undercut form A

This program can be used to program a thread undercut of form A.

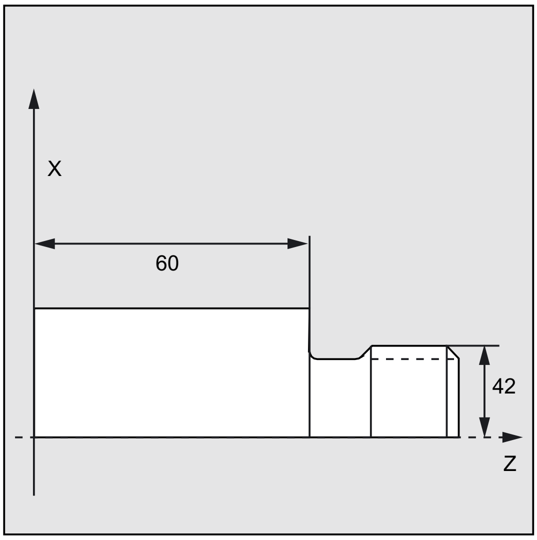


Figure 10-64 Programming example: Thread undercut form A

```
N10 D3 T1 S300 M3 G95 F0.3                  ; Specification of technology
                                            values
N20 G0 G90 Z100 X50                         ; Selection of starting position
N30 CYCLE96 (42, 60, "A")                   ; Cycle call
N40 G90 G0 X100 Z100                        ; Approach next position
N50 M2                                      ; End of program
```

10.5.6 Thread cutting - CYCLE97

Programming

CYCLE97(PIT, MPIT, SPL, FPL, DM1, DM2, APP, ROP, TDEP, FAL, IANG, NSP, NRC, NID, VARI, NUMT, VRT)

parameters

Table 10- 18 Parameters for CYCLE97

Parameter	Data type	Significance
PST	REAL	Thread lead as a value (enter without sign)
MPIT	REAL	Thread lead as thread size Range of values: 3 (for M3) ... 60 (for M60)
SPL	REAL	Thread starting point in the longitudinal axis
FPL	REAL	Thread end point in the longitudinal axis
DM1	REAL	Thread diameter at the starting point
DM2	REAL	Thread diameter at the end point
APP	REAL	Run-in path (enter without sign)
ROP	REAL	Run-out path (enter without sign)
TDEP	REAL	Thread depth (enter without sign)
FAL	REAL	Finishing allowance (enter without sign)
IANG	REAL	Infeed angle Range of values: "+" (for flank infeed at the flank), "-" (for alternating flank infeed)
NSP	REAL	Starting point offset for the first thread turn (enter without sign)
NRC	INT	Number of roughing cuts (enter without sign)
NID	INT	Number of idle passes (enter without sign)
VARI	INT	Definition of the machining type for the thread Range of values: 1 ... 4
NUMT	INT	Number of thread turns (enter without sign)
VRT	REAL	Variable retraction path based on initial diameter, incremental (enter without sign)

Function

Use the thread cutting cycle to produce cylindrical and tapered external and internal threads with constant lead in longitudinal and face machining. The thread can be single or multiple. With multiple threads, the individual thread turns are machined one after the other.

The infeed is performed automatically; you can choose between the variants constant infeed per cut or constant cutting cross-section.

Right-hand or left hand thread is determined by the direction of rotation of the spindle which must be programmed prior to the cycle start.

Both feed and spindle override are ineffective in the traversing blocks with thread.

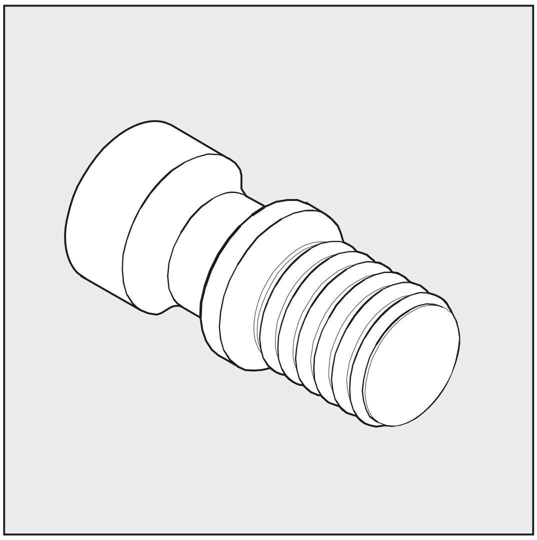


Figure 10-65 Thread

NOTICE
To be able to use this cycle, a speed-controlled spindle with position measuring system is required.

Sequence

Position reached prior to cycle start:

Starting position is any position from which the programmed thread starting point + run-in path can be approached without collision.

The cycle creates the following sequence of motions:

- Approach of the starting point determined in the cycle at the beginning of the run-in path for the first thread turn with G0
- Infeed for roughing according to the infeed type defined under VARI.
- Thread cutting is repeated according to the programmed number of roughing cuts.
- The finishing allowance is removed in the following step with G33.
- This step is repeated according to the number of idle passes.
- The whole sequence of motions is repeated for each further thread turn.

Explanation of the parameters

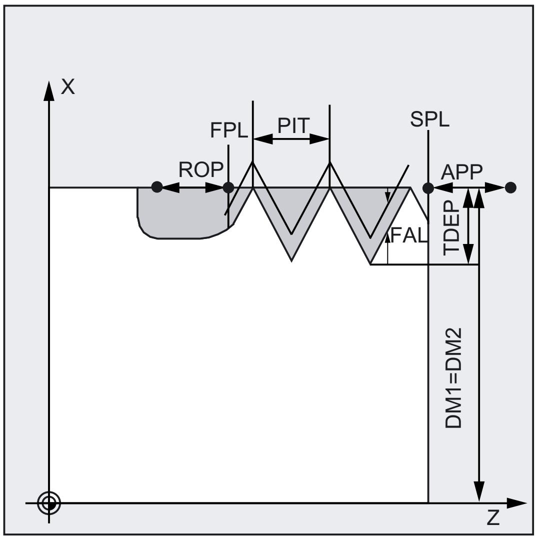


Figure 10-66 Parameters for CYCLE97

PIT and MPIT (value and thread size)

The thread lead is an axis-parallel value and is specified without sign. To produce metric cylindrical threads, it is also possible to specify the thread lead as a thread size via the parameter MPIT (M3 to M60). Only one of the two parameters should be used by option. If they contain contradicting values, the cycle generates the alarm 61001 "Invalid thread lead" and is aborted.

DM1 and DM2 (diameter)

Use this parameter to define the thread diameter of starting and end point of the thread. In the case of internal threads, this is the tap-hole diameter.

Interrelation SPL, FPL, APP and ROP (starting, end point, run-in and run-out path)

The programmed starting point (SPL) or end point (FPL) constitutes the original starting point of the thread. The starting point used in the cycle, however, is the starting point brought forward by the run-in path APP, and, correspondingly, the end point is the programmed end point brought back by the run-out path ROP. In the transversal axis, the starting point defined by the cycle is always by 1 mm above the programmed thread diameter. This lift-off plane is generated automatically within the control system.

Interrelation TDEP, FAL, NRC and NID (thread depth, finishing allowance, number of cuts)

The programmed finishing allowance acts paraxially and is subtracted from the specified thread depth TDEP; the remainder is divided into roughing cuts.

The cycle will calculate the individual infeed depth automatically, depending on the VARI parameter.

When the thread depth is divided into infeeds with constant cutting cross-section, the cutting force will remain constant over all roughing cuts. In this case, the infeed will be performed using different values for the infeed depth.

A second version is the distribution of the whole thread depth to constant infeed depths. When doing so, the cutting cross-section becomes larger from cut to cut, but with smaller values for the thread depth, this technology can result in better cutting conditions.

The finishing allowance FAL is removed after roughing in one step. Then the idle passes programmed under parameter NID are executed.

IANG (infeed angle)

By using parameter IANG, the angle is defined under which the infeed is carried out in the thread. If you wish to infeed at a right angle to the cutting direction in the thread, the value of this parameter must be set to zero. If you wish to infeed along the flanks, the absolute value of this parameter may amount maximally to the half of the flank angle of the tool.

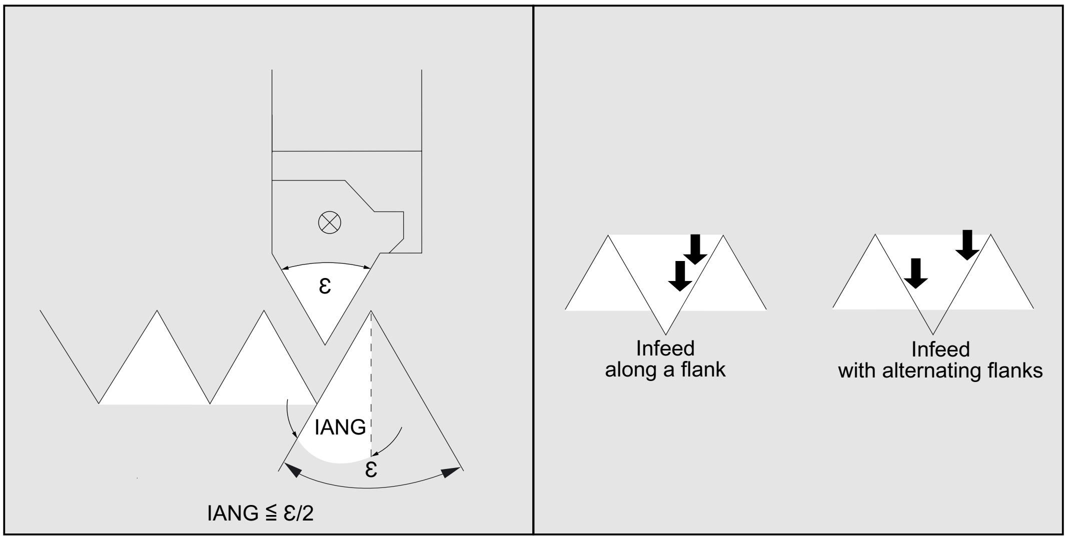


Figure 10-67 Infeed angle

The execution of the infeed is defined by the sign of this parameter. With a positive value, infeed is always carried out at the same flank, and with a negative value, at both flanks alternating. The infeed type with alternating flanks is only possible for cylindrical threads. If the value of IANG for tapered threads is nonetheless negative, the cycle will carry out a flank infeed along a flank.

NSP (starting point offset) and NUMT (number)

You can use this parameter to program the angle value defining the point of the first cut of the thread turn at the circumference of the turned part. This involves a starting point offset. The parameter can assume values between 0 and +359.9999 degrees. If no starting point offset has been specified or the parameter has been omitted from the parameter list, the first thread turn automatically starts at the zero-degree mark.

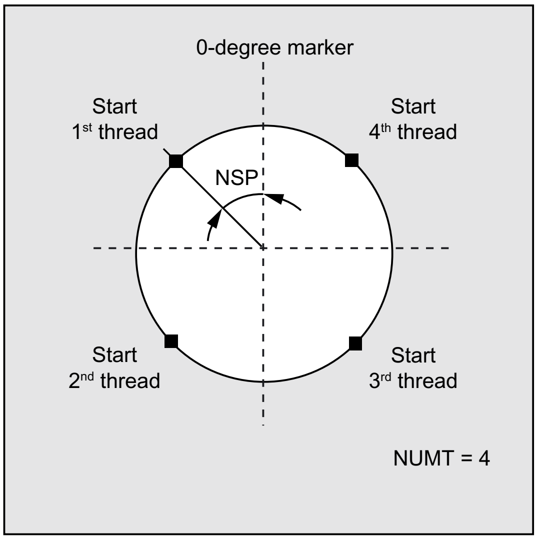


Figure 10-68 Starting point offset and number

Use the NUMT parameter to define the number of thread turns with a multiple-turn thread. For a single-turn thread, the parameter must be assigned zero or can be dropped completely in the parameter list.

The thread turns are distributed equally over the circumference of the turned part; the first thread turn is determined by the NSP parameter.

To produce a multiple-turn thread with an asymmetrical arrangement of the thread turns on the circumference, the cycle for each thread turn must be called when programming the appropriate starting point offset.

VARI (machining type)

By using the VARI parameter, it is defined whether external or internal machining will be carried out and which technology will be used with regard to the infeed when roughing. The VARI parameter can assume values between 1 and 4 with the following meaning:

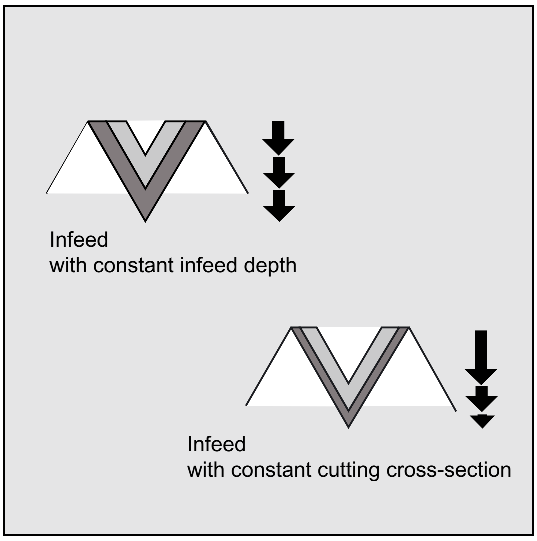

Infeed
with constant infeed depth
Infeed
with constant cutting cross-section

Figure 10-69 Machining type

Table 10- 19 Type of machining

Value	Ext./int.	Const. Infeed/const. cutting cross-section
1	O	Constant infeed
2	I	Constant infeed
3	O	Constant cutting cross-section
4	I	Constant cutting cross-section

If a different value is programmed for the VARI parameter, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

VRT (variable retraction path)

The retraction path can be programmed on the basis of the initial thread diameter in the VRT parameter. For VRT = 0 (parameter not programmed), the retraction path is 1 mm. The retraction path is always measured according to the programmed system of units, inch or metric.

Differentiation between longitudinal and face thread

The decision whether a longitudinal or face thread is to be machined is made by the cycle itself. This depends on the angle of the taper at which the threads are cut. If the angle at the taper is ≤45 degrees, the longitudinal axis thread is machined, otherwise it will be the face thread.

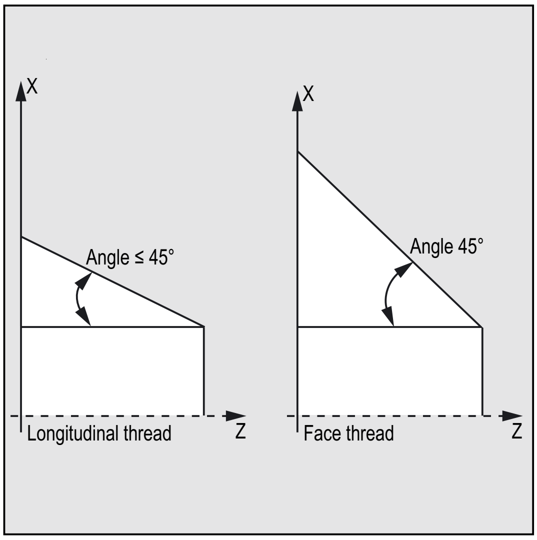


Figure 10-70 Longitudinal and face thread

Programming example: Thread cutting

By using this program, you can produce a metric external thread M42x2 with flank infeed. Infeed is carried out with constant cutting cross-section. 5 roughing cuts are carried out at a thread depth of 1.23 mm without finishing allowance. At completion of this operation, 2 idle passes will be carried out.

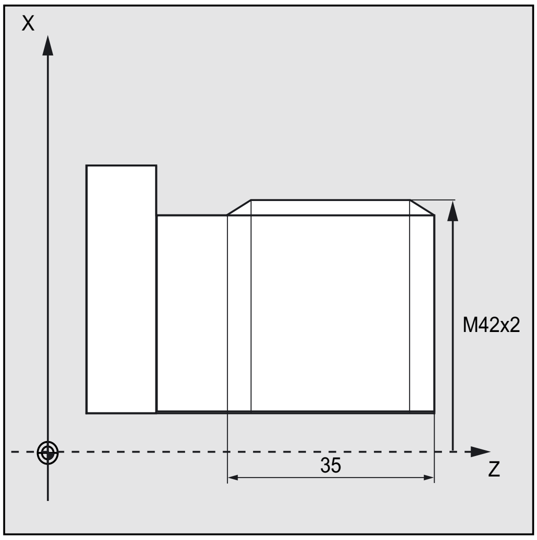


Figure 10-71 Programming example: Thread cutting

```
N10 G0 G90 Z100 X60                              ; Selection of starting position
N20 G95 D1 T1 S1000 M4                           ; Specification of technology
                                                 values
N30 CYCLE97( , 42, 0, -35, 42, 42, 10, 3, 1.23,  ; Cycle call
0, 30, 0, 5, 2, 3, 1)
N40 G90 G0 X100 Z100                             ; Approach next position
N50 M2                                           ; End of program
```

10.5.7 Chaining of threads – CYCLE98

Programming

CYCLE98 (PO1, DM1, PO2, DM2, PO3, DM3, PO4, DM4, APP, ROP, TDEP, FAL, IANG, NSP, NRC, NID, PP1, PP2, PP3, VARI, NUMT, _VRT)

parameters

Table 10- 20 Parameters for CYCLE98

Parameter	Data type	Significance
PO1	REAL	Thread starting point in the longitudinal axis
DM1	REAL	Thread diameter at the starting point
PO2	REAL	First intermediate point in the longitudinal axis
DM2	REAL	Diameter at the first intermediate point
PO3	REAL	Second intermediate point
DM3	REAL	Diameter at the second intermediate point
PO4	REAL	Thread end point in the longitudinal axis
DM4	REAL	Diameter at the end point
APP	REAL	Run-in path (enter without sign)
ROP	REAL	Run-out path (enter without sign)
TDEP	REAL	Thread depth (enter without sign)
FAL	REAL	Finishing allowance (enter without sign)
IANG	REAL	Infeed angle Range of values: "+" (for flank infeed at the flank), "-" (for alternating flank infeed)
NSP	REAL	Starting point offset for the first thread turn (enter without sign)
NRC	INT	Number of roughing cuts (enter without sign)
NID	INT	Number of idle passes (enter without sign)
PP1	REAL	Thread lead 1 as a value (enter without sign)
PP2	REAL	Thread lead 2 as a value (enter without sign)
PP3	REAL	Thread lead 3 as a value (enter without sign)
VARI	INT	Definition of the machining type for the thread Range of values: 1 ... 4
NUMT	INT	Number of thread turns (enter without sign)
VRT	REAL	Variable retraction path based on initial diameter, incremental (enter without sign)

Function

This cycle can be used to produce several cylindrical or tapered threads in succession. The individual thread sections can have different leads whereby the lead within one and the same thread section must be constant.

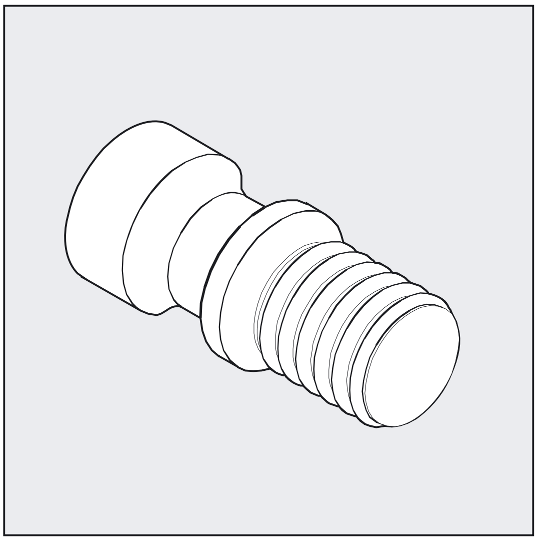


Figure 10-72 Side-by-side thread mounting

Sequence

Position reached prior to cycle start:

Starting position is any position from which the programmed thread starting point + run-in path can be approached without collision.

The cycle creates the following sequence of motions:

- Approach of the starting point determined in the cycle at the beginning of the run-in path for the first thread turn with G0
- Infeed for roughing according to the infeed type defined under VARI.
- Thread cutting is repeated according to the programmed number of roughing cuts.
- The finishing allowance is removed in the following step with G33.
- This step is repeated according to the number of idle passes.
- The whole sequence of motions is repeated for each further thread turn.

Explanation of the parameters

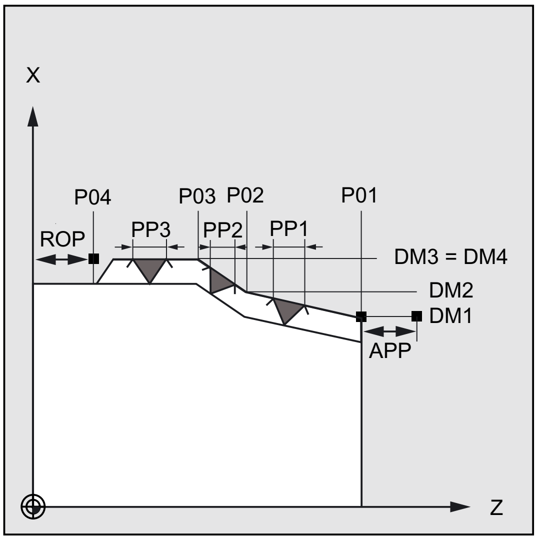


Figure 10-73 Parameters for CYCLE98

PO1 and DM1 (starting point and diameter)

These parameters are used to define the original starting point for the thread series. The starting point determined by the cycle itself and approached at the beginning using G0 is located by the run-in path before the programmed starting point (starting point A in the diagram on the previous page).

PO2, DM2 and PO3, DM3 (intermediate point and diameter)

These parameters are used to define two intermediate points in the thread.

PO4 and DM4 (end point and diameter)

The original end point of the thread is programmed under parameters PO4 and DM4.

With an inside thread, DM1...DM4 corresponds to the tap hole diameter.

Interrelation between APP and ROP (run-in/run-out paths)

The starting point used in the cycle, however, is the starting point brought forward by the run-in path APP, and, correspondingly, the end point is the programmed end point brought back by the run-out path ROP.

In the transversal axis, the starting point defined by the cycle is always by 1 mm above the programmed thread diameter. This lift-off plane is generated automatically within the control system.

Interrelation between TDEP, FAL, NRC and NID (thread depth, finishing allowance, number of roughing and idle passes)

The programmed finishing allowance acts paraxially and is subtracted from the specified thread depth TDEP; the remainder is divided into roughing cuts. The cycle will calculate the individual infeed depth automatically, depending on the VARI parameter. When the thread depth is divided into infeeds with constant cutting cross-section, the cutting force will remain constant over all roughing cuts. In this case, the infeed will be performed using different values for the infeed depth.

A second version is the distribution of the whole thread depth to constant infeed depths. When doing so, the cutting cross-section becomes larger from cut to cut, but with smaller values for the thread depth, this technology can result in better cutting conditions.

The finishing allowance FAL is removed after roughing in one step. Then the idle passes programmed under parameter NID are executed.

IANG (infeed angle)

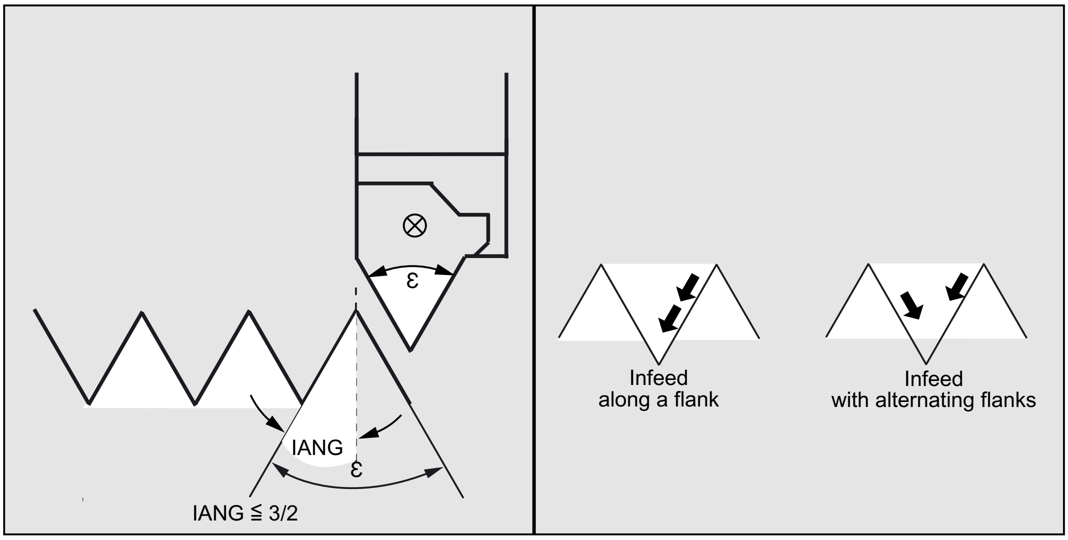


Figure 10-74 Infeed angle

By using parameter IANG, the angle is defined under which the infeed is carried out in the thread. If you wish to infeed at a right angle to the cutting direction in the thread, the value of this parameter must be set to zero. This means that the parameter can be omitted in the parameter list, as in this case the value is defaulted automatically with zero. If you wish to infeed along the flanks, the absolute value of this parameter may amount maximally to the half of the flank angle of the tool.

The execution of the infeed is defined by the sign of this parameter. With a positive value, infeed is always carried out at the same flank, and with a negative value, at both flanks alternating. The infeed type with alternating flanks is only possible for cylindrical threads. If the value of IANG for tapered threads is nonetheless negative, the cycle will carry out a flank infeed along a flank.

NSP (starting point offset)

You can use this parameter to program the angle value defining the point of the first cut of the thread turn at the circumference of the turned part. This involves a starting point offset. The parameter can assume values between 0.0001 and +359.9999 degrees. If no starting point offset has been specified or the parameter has been omitted from the parameter list, the first thread turn automatically starts at the zero-degree mark.

PP1, PP2 and PP3 (thread lead)

These parameters are used to define the value of the thread lead in the three sections of the thread series. The lead value must be entered as a paraxial value without sign.

VARI (machining type)

By using the VARI parameter, it is defined whether external or internal machining will be carried out and which technology will be used with regard to the infeed when roughing. The VARI parameter can assume values between 1 and 4 with the following meaning:

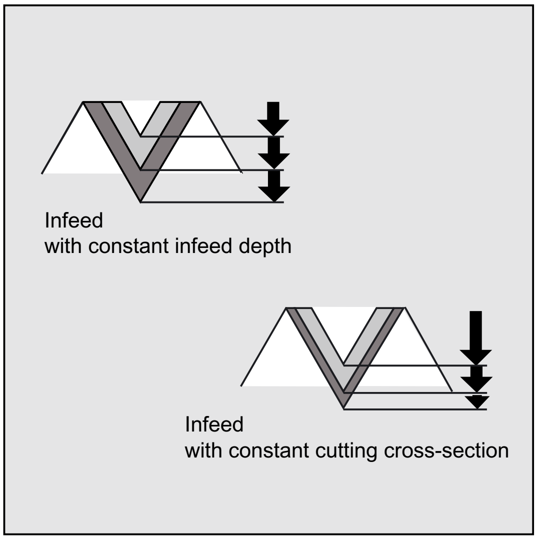

Infeed
with constant infeed depth
Infeed
with constant cutting cross-section

Figure 10-75 Machining type

Value	Ext./int.	Const. Infeed/const. cutting cross-section
1	External	Constant infeed
2	Internal	Constant infeed
3	External	Constant cutting cross-section
4	Internal	Constant cutting cross-section

If a different value is programmed for the VARI parameter, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

NUMT (number of thread turns)

Use the NUMT parameter to define the number of thread turns with a multiple-turn thread. For a single-turn thread, the parameter must be assigned zero or can be dropped completely in the parameter list.

The thread turns are distributed equally over the circumference of the turned part; the first thread turn is determined by the NSP parameter.

To produce a multiple-turn thread with an asymmetrical arrangement of the thread turns on the circumference, the cycle for each thread turn must be called when programming the appropriate starting point offset.

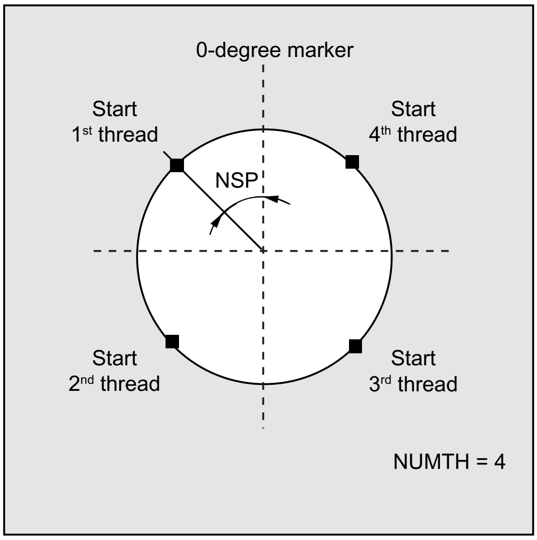


Figure 10-76 Number of threads

VRT (variable retraction path)

The retraction path can be programmed on the basis of the initial thread diameter in the VRT parameter. For VRT = 0 (parameter not programmed), the retraction path is 1 mm. The retraction path is always measured according to the programmed measuring system, inch or metric.

Programming example: Thread chain

You can use this program to produce a thread chain starting with a cylindrical thread. The infeed is performed vertically to the thread; neither finishing allowance, nor starting point offset are programmed. Five roughing cuts and one noncut are performed. The machining type is defined as longitudinal, external, with constant cross-section of cut.

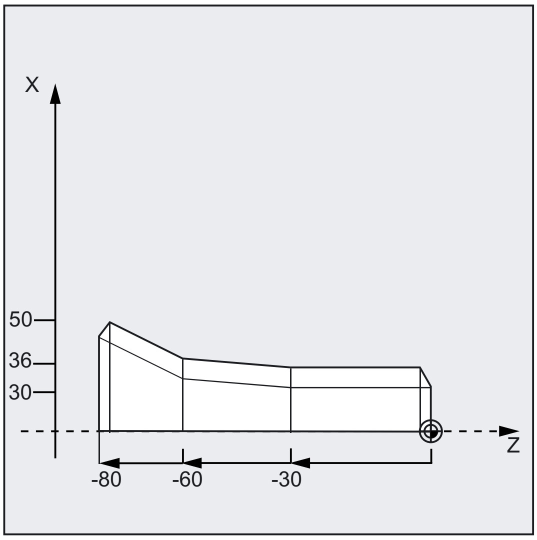


Figure 10-77 Programming example: Thread chain

```
N10 G95 T5 D1 S1000 M4                           ; Specification of technology
                                                 values
N20 G0 X40 Z10                                   ; Approach starting position
N30 CYCLE98 (0, 30, -30, 30, -60, 36, -80, 50,   ; Cycle call
10, 10, 0.92, , , , 5, 1, 1.5, 2, 2, 3, 1)
N40 G0 X55                                       ; Traverse axis by axis
N50 Z10
N60 X40
N70 M2                                           ; End of program
```

10.6 Error messages and error handling

10.6.1 General Information

If error conditions are detected in the cycles, an alarm is generated and the execution of the cycle is aborted.

Furthermore, the cycles display their messages in the message line of the control system. These message will not interrupt the program execution.

The errors with their reactions and the messages in the message line of the control system are described in conjunction with the individual cycles.

10.6.2 Error handling in the cycles

Alarms with numbers between 61000 and 62999 generated in the cycles. This range of numbers, in turn, is divided again with regard to alarm responses and cancel criteria.

The error text that is displayed together with the alarm number gives you more detailed information on the error cause.

Alarm number	Clearing criterion	Alarm Response
61000 ... 61999	NC_RESET	Block preparation in the NC is aborted
62000 ... 62999	Clear key	The block preparation is interrupted; the cycle can be continued with NC START after the alarm has been cleared.

10.6.3 Overview of cycle alarms

The error numbers are classified as follows:

6	_	X	_	_

- X=0 General cycle alarms
- X=1 Alarms generated by the drilling, drilling pattern and milling cycles
- X=6 Alarms generated by the drilling cycles

The Table below includes a list of all errors occurring in the cycles with their location of occurrence and appropriate instructions for fault correction.

Alarm No.	Alarm text	Source	Explanation, Remedy
61000	"No tool offset active"	CYCLE93 to CYCLE96	D offset must be programmed prior to cycle call
61001	"Illegal thread lead"	CYCLE84 CYCLE840 CYCLE96 CYCLE97	Check the parameters for the thread size or the specifications for the lead (are contradicting)
61002	"Machining type defined incorrectly"	CYCLE93 CYCLE95 CYCLE97	The value of parameters VARI for the machining type is specified incorrectly and must be changed
61101	"Reference plane defined incorrectly"	CYCLE81 to CYCLE89 CYCLE840	Either different values for reference and retraction plane must be selected in the case of relative specification of the depth or an absolute value must be specified for the depth.
61102	"No spindle direction programmed"	CYCLE88 CYCLE840	The parameter SDIR (or SDR in CYCLE840) must be programmed
61107	"First drilling depth defined incorrectly"	CYCLE83	First drilling depth is opposite to total drilling depth
61601	"Finished part diameter too small"	CYCLE94 CYCLE96	The finished part diameter programmed is too small.
61602	"Tool width defined incorrectly"	CYCLE93	Cutting tool is larger than programmed groove width
61603	"Groove shape defined incorrectly"	CYCLE93	• Radii/chamfers on recess base do not match with groove width • Face groove at a contour element running parallel to the longitudinal axis is not possible.
61604	"Active tool violates programmed contour"	CYCLE95	Contour violation in relief cut elements due to clear cutting angle of the tool used, i.e. use a different tool or check the contour subroutine
61605	"Contour programmed incorrectly"	CYCLE95	Illegal relief cut element detected

Alarm No.	Alarm text	Source	Explanation, Remedy
61606	"Error in contour preparation"	CYCLE95	An error has been found in the contour preparation; this alarm always occurs in conjunction with an NCK alarm 10930 ... 10934, 15800 or 15810
61607	"Starting point programmed incorrectly"	CYCLE95	The starting point reached prior to the cycle call is not outside the rectangle described by the contour subroutine.
61608	"Invalid tool point direction programmed"	CYCLE94 CYCLE96	A tool point direction 1...4 matching the form of the undercut must be programmed.
61609	"Form defined incorrectly"	CYCLE94 CYCLE96	Check the parameters for the form of the undercut.
61611	"No point of intersection found"	CYCLE95	No intersection could be calculated with the contour. Check contour programming or change infeed depth.

10.6.4 Messages in the cycles

The cycles display their messages in the message line of the control system. These message will not interrupt the program execution.

Messages provide information with regard to a certain behavior of the cycles and with regard to the progress of machining and are usually kept beyond a machining step or until the end of the cycle. The following messages are possible:

Message text	Source
"Depth: according to the value for the relative depth"	CYCLE82...CYCLE88, CYCLE840
"1. drilling depth: according to the value for the relative depth"	CYCLE83
"Thread start <No.> - longitudinal thread machining"	CYCLE97
"Thread start <No.> - face thread machining"	CYCLE97

In each case, <No.> stands for the number of the figure that is currently being machined.

Network operation 11

11.1 Network operation prerequisites

Introduction

A network function is available for communicating between the control system and a PG/PC.

Prerequisites

The RCS802 tool is required on the PG/PC for communication.

For connecting the control system via the network, various options are available.

These options are described in the chapters "RCS tool" and "Network operation".

The connections are enabled via the following control system interfaces:

- RS232 interface
- Ethernet peer-to-peer interface
- Interface Ethernet network (available only for SINUMERIK 802D sl)

11.2 RCS802 tool

With the RCS802 tool (Remote Control System), you have a tool for your PG/PC that will support you in your daily work with SINUMERIK 802D sl.

The RCS802 tool is part of the SINUMERIK802Dsl and is supplied as CD with each control.

You can connect the control system and the PG/PC using the following interfaces:

Table 11- 1 Interfaces

Interfaces	SINUMERIK 802D sl	RCS802 on PG/PC
RS232	Is available for value, plus and pro.	Are available.
Peer-to-peer Ethernet	Is available for value, plus and pro.	Are available.
Ethernet network	Only available for SINUMERIK 802D sl pro.	Function that requires a license

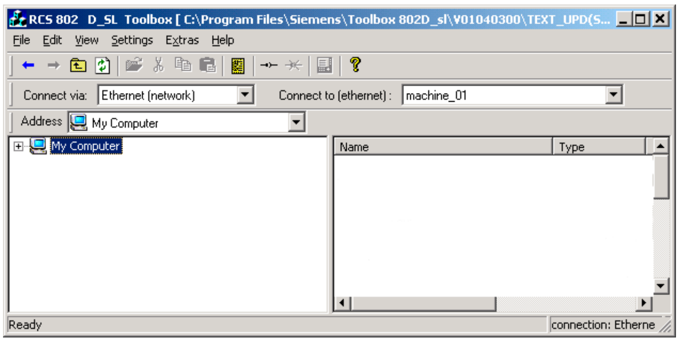
Functions of the RCS802 tool with license key

NOTICE
You will only obtain the full functionality of the RCS802 tool after importing the license key RCS802.

Table 11- 2 Functions of the RCS802 tool that require a license

Function	RCS802 tool without license key	RCS802 tool with license key
Managing projects	Yes	Yes
Data exchange with SINUMERIK 802D sl	Yes	Yes
Commissioning SINUMERIK 802D sl	Yes	Yes
Setting-up a share drive	No	Yes
Remote control	No	Yes
Screen shot	No	Yes

RCS802 tool



RCS 802 D_SL Toolbox [C:\Program Files\Siemens\Toolbox 802D_sl\V01040300\TEXT_UPD(S...

Connect via: Ethernet (network)

Connect to (ethernet) : machine_01

Address My Computer

Ready

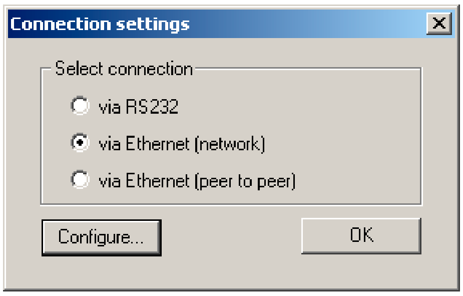
connection: Etherne

Figure 11-1 Explorer window of the RCS802 tool

After starting the RCS802 tool, you will be in OFFLINE mode. In this mode you only manage files on your PC.

In the ONLINE mode, the directory **Control 802** is also available. This directory makes data exchange with the control system possible. In addition, a remote control function is provided for process monitoring.

The ONLINE connections from the PG/PC to the control are parameterized/activated via the "Setting" > "Connection" menu items in the "Connection Settings" dialog box.



Connection settings

Select connection

○ via RS232

● via Ethernet (network)

○ via Ethernet (peer to peer)

Figure 11-2 Connection Settings

Note

The RCS802 tool includes a detailed online help function. Refer to this help menu for further details e.g. establishing a connection, project management etc.

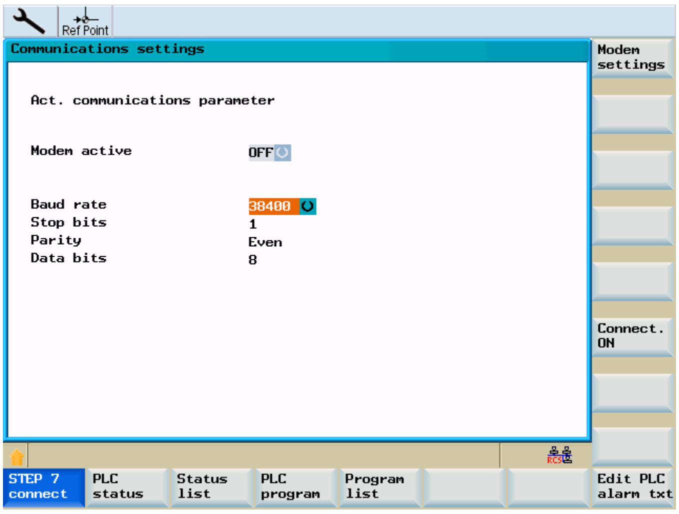
Operating sequence to make an RS232 connection to the control

SHIFT + SYSTEM ALARM

- You are now in the <SYSTEM> operating area.

PLC

- Press the "PLC" softkey.



Communications settings

Act. communications parameter

Modem active	OFF
Baud rate	38400
Stop bits	1
Parity	Even
Data bits	8

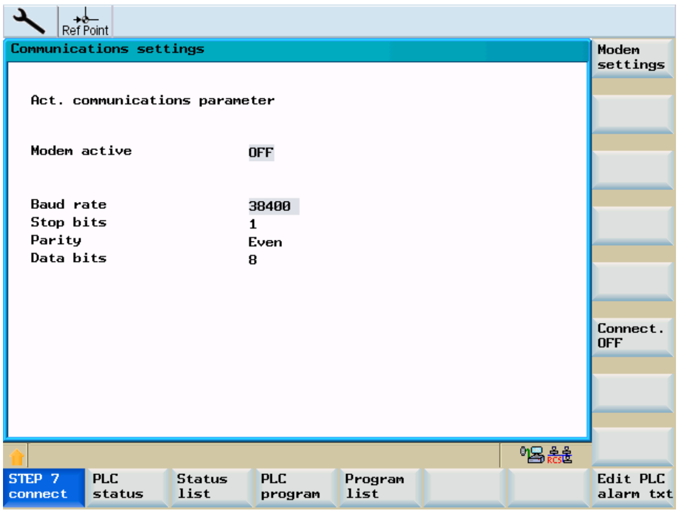
Figure 11-3 Communication settings RS232

Connect to STEP 7

- Set the parameters for communication in the "STEP 7 Connect" dialog.

Connect. on

- Activate the RS232 connection with the "Connect. ON" softkey.



Communications settings

Act. communications parameter

Modem active	OFF
Baud rate	38400
Stop bits	1
Parity	Even
Data bits	8

Figure 11-4 RS232 connection active

No modifications to the settings are possible in this state.

The softkey label changes to "Connect. OFF".

In the lower right corner of the screen, the icon shows that the connection to PG/PC via the RS232 interface is active.

Operating sequence to make an Ethernet peer-to-peer connection to the control

SHIFT + SYSTEM ALARM

- You are now in the <SYSTEM> operating area.

Service display

- Press the softkeys "Service display" >"Service control".

Service control

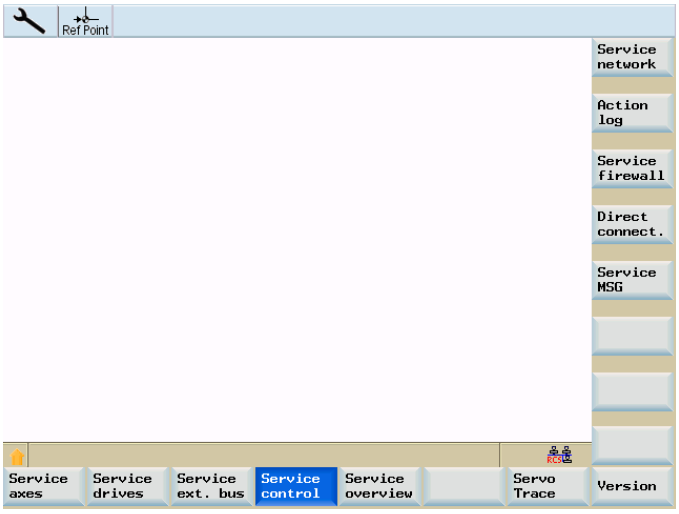


Figure 11-5 "Service control"

Direct connection

- Press the "Direct connect." softkey.

  The following message is shown on the HMI:

  "Connection is set up"

  – IP Address: 169.254.11.22

  – Subnet mask: 255.255.0.0

  The IP address and subnet mask shown are fixed values.

  These values cannot be changed.

Direct connection

- You can cancel the Ethernet peer-to-peer connection once more using the "Direct connect." softkey.

Operating sequence to make an Ethernet network connection to the control

SHIFT + SYSTEM ALARM

- You are now in the <SYSTEM> operating area.

Service display

Service control

- Press the softkeys "Service display" >"Service control".

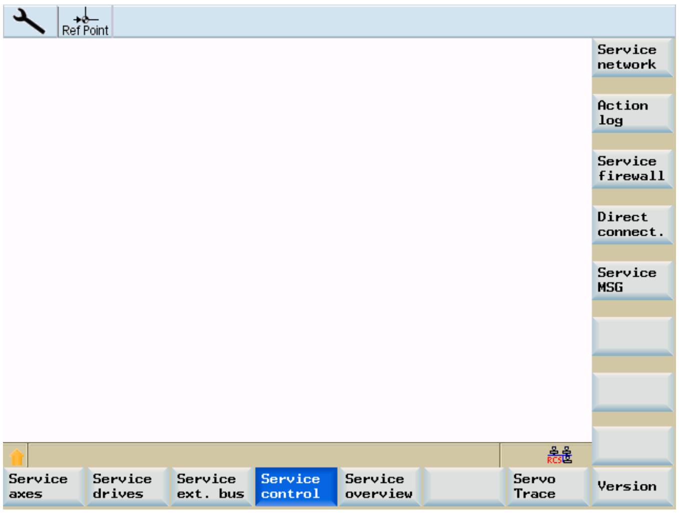


Figure 11-6 "Service control"

Service network

- Press the softkey "Service network" (only available for SINUMERIK 802D sl pro).

Reference

SINUMERIK 802D sl Programming and Operating Manual; Network Operation

11.3 Network operation

11.3.1 Network operation

Note

The network function is only available for SINUMERIK 802D sl.

Thanks to the integrated network adapter, the control system is network-capable. The following connections are possible:

- Ethernet peer-to-peer: Direct connection between control system and PC using a cross-over cable
- Ethernet network: Integrating the control system into an existing Ethernet network using a patch cable.

Screened network operation with encrypted data transfer is possible using an 802D specific transmission protocol. This protocol is used, e.g. for transmitting and executing part programs in conjunction with the RCS tool.

11.3.2 Configuring the network connection

Prerequisite

The control system is connected to the PC or the local network via the X5 interface.

Entering network parameters

SHIFT + SYSTEM ALARM

Switch to the the <SYSTEM> operating area.

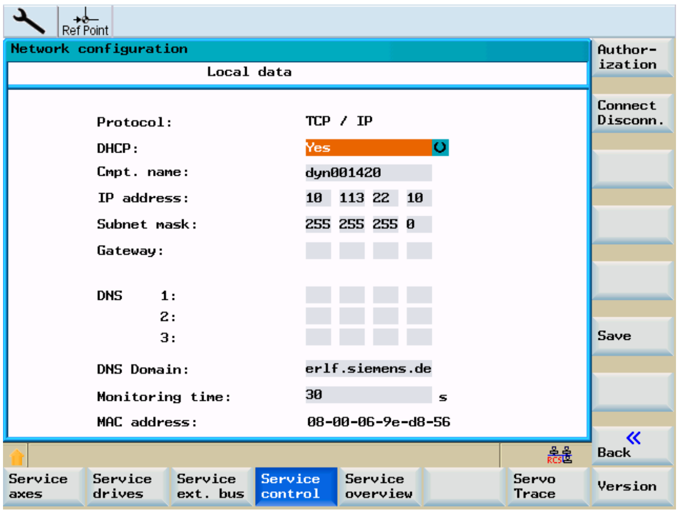
Service display

Service control

Press the "Service display" "Service control system" softkeys.

Service network

Select the "Service network" softkey to display the network configuration window.



Network configuration

Local data

Protocol: TCP / IP

DHCP: Yes

Cmpt. name: dyn001420

IP address: 10 113 22 10

Subnet mask: 255 255 255 0

Gateway:

DNS 1:
2:
3:

DNS Domain: erlf.siemens.de

Monitoring time: 30 s

MAC address: 08-00-06-9e-d8-56

Figure 11-7 "Network configuration" start screen

Table 11- 3 Network configuration required

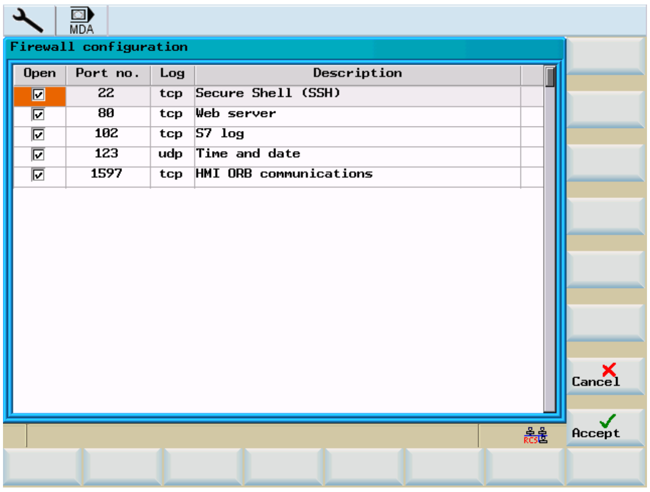
Parameter	Explanation
DHCP	DHCP log: A DHCP server is needed in the network which dynamically distributes the IP addresses. When **No** is selected fixed network addresses will be assigned. When **Yes** is selected the network addresses are assigned dynamically. Input fields that are no longer needed will be hidden. If you selected "yes", the following steps are necessary to activate the fields for the computer name, IP address and Subnet mask: 1. Press the vertical softkey "Save". 2. Switch the control system off and on again.
Computer name	Name of the control system in the network
IP address	Network address of the control system (e.g. 192.168.1.1)
Subnet mask	Network identification (e.g. 255.255.252.0)

Enabling the communication ports

Service Firewall

Use the "Service Firewall" softkey to enable or disable communication ports.

To ensure maximum possible safety, all ports not needed should be closed.



Firewall configuration

Open	Port no.	Log	Description
☑	22	tcp	Secure Shell (SSH)
☑	80	tcp	Web server
☑	102	tcp	S7 log
☑	123	udp	Time and date
☑	1597	tcp	HMI ORB communications

Figure 11-8 Firewall configuration

The RCS network requires the ports 80 and 1597 for communication.

To change the port status, select the relevant port using the cursor. Pressing the <Input> key changes the port status.

Open ports are shown with the checkbox enabled.

11.3.3 User management

SHIFT + SYSTEM ALARM

Press the "Service display" "Service control system" softkeys in the <SYSTEM> operating area.

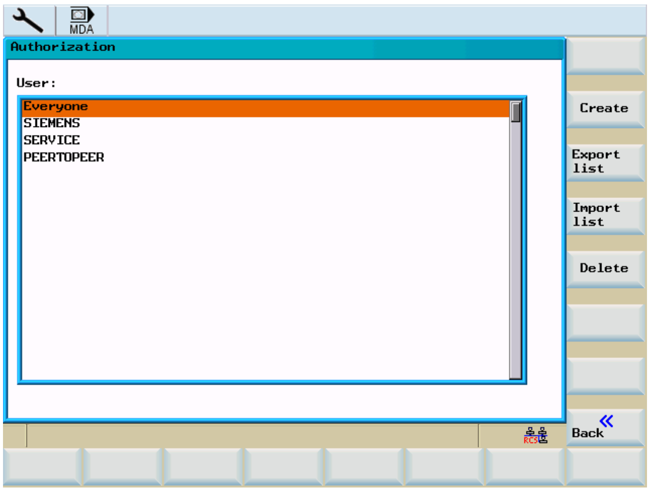
Service display

Service control

Service network

Author-ization

Select the "Service network" "Authorization" softkey to display the user account input screen.



Everyone
SIEMENS
SERVICE
PEERTOPEER

Figure 11-9 User accounts

The user accounts serve for saving personal settings of the users. To create a new account, type the user name and the log-in password in the input fields.

A user account is required for communication between HMI and the RCS tool on the programming device/PC.

For this purpose the user has to enter this password on the HMI during RCS log-in via network.

This password is required also, if the user want to communicate with the control system from the RCS tool.

Use the "Create" softkey to insert a new user into the user management.

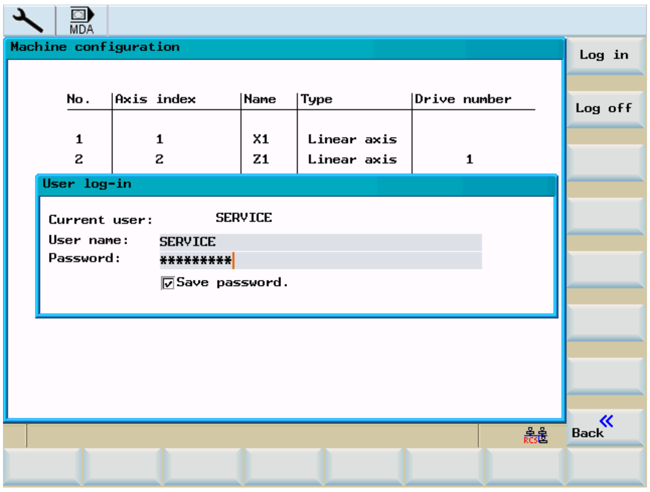
Use the "Delete" softkey to delete the selected user from the user management.

11.3.4 User log in - RCS log in

SHIFT + SYSTEM ALARM

RCS log-in

In the <SYSTEM> operating area, select the "RCS Connect" softkey. The user log-in input screen will appear.



Machine configuration

No.	Axis index	Name	Type	Drive number
1	1	X1	Linear axis	
2	2	Z1	Linear axis	1

User log-in

Current user: SERVICE
User name: SERVICE
Password: *********
☑ Save password.

Figure 11-10 User log-in

Logon

Type user name and password into the appropriate input fields and select the "Log in" softkey to confirm your input.

After successful log-in, the user name is displayed in the **Current user** line.

Select the "Back" softkey to close the dialog box.

Note

This log-in simultaneously serves for user identification for remote connections.

Logoff

Press the "Log off" softkey. This will log out the current user, all user-specific settings are saved, and any enables already granted are canceled.

11.3.5 Working on the basis of a network connection

The remote access (access to the control system from a PC or from a network) to the control system is disabled by default.

After log-in of a local user, the following functions are offered to the RCS tool:

- Commissioning functions
- Data transfer (transfer of part programs)
- Remote control for the control system

To grant access to a part of the file system, first share the relevant directories with other users.

Note

If you share directories with other users, the authorized network nodes are granted access to the shared files in the control system. Depending on the sharting option, the user can modify or delete files.

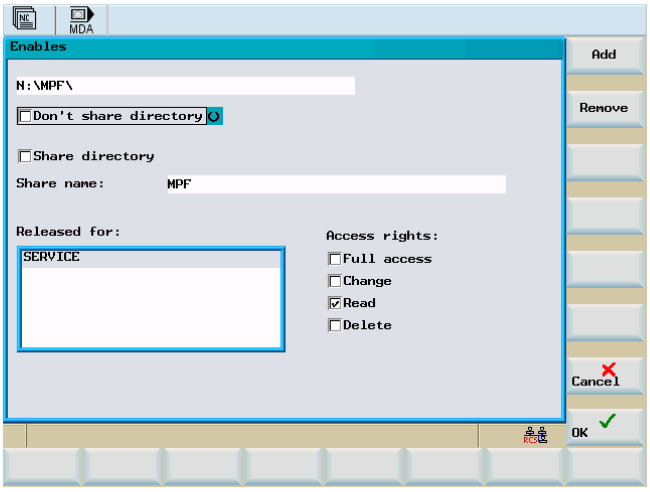
11.3.6 Sharing directories

This function defines the rights for access of remote users to the file system of the control system.

PROGRAM MANAGER

Use the **Program manager** to select the directory you want to share.

Use the "Next..." > "Share" softkeys to open the input screen for sharing the selected directory.



Enables

N:\MPF\

□ Don't share directory

□ Share directory

Share name: MPF

Released for:

SERVICE

Access rights:

□ Full access

□ Change

☑ Read

□ Delete

Figure 11-11 Sharing status

- Select the sharing status for the selected directory:
  - **Do not share this directory** Directory will not be shared
  - **Share this directory** The directory will be shared and a share name must be entered.
- Type an identifier into the **Share name** field through which authorized users can access the files in the directory.
- By pressing the "Add" softkey, you arrive at the user list. Select the user. With "Add" you can make any entries in the "Shared" field.
- Define the user rights (**Authorizations**).
  - **Full access** User has full access
  - **Change** User may modify files.
  - **Read** User may read files.
  - **Delete** User may delete files.

By pressing the "OK" softkey the set properties are confirmed. As in Windows, shared directories are marked with a "hand".

11.3.7 Connecting / disconnecting network drives

SHIFT + SYSTEM ALARM

Press the "Service display" "Service control system" "Service network" softkeys in the <SYSTEM> operating area.

Service display

Service control

Service network

Connect Disconn.

Use "Connect/Disconnect" to enter the network drive configuration area.

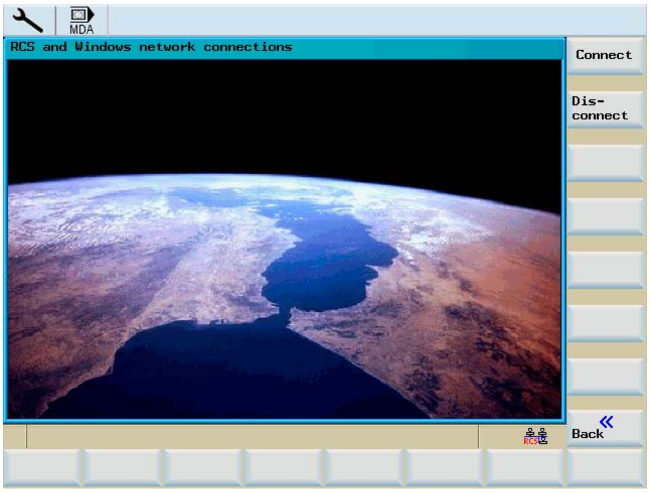


Figure 11-12 Network connections

Connecting network drives

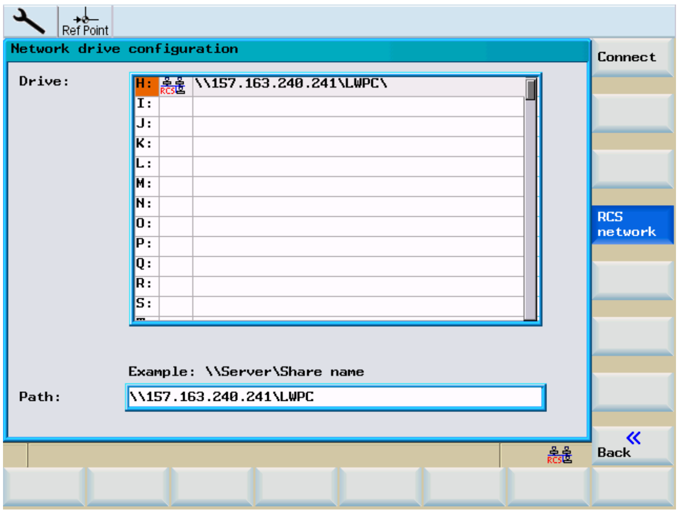
Connect

The "Connect" function is used to assign a local drive to a network drive.

Note

You have shared a directory for a network connection with a certain user on a programming device/PC.

The RCS802 tool includes a detailed online help function. The procedure for using this help function is described in Chapter "RCS802 share drive".



Network drive configuration

Drive:

H: \\157.163.240.241\LWPC\
I:
J:
K:
L:
M:
N:
O:
P:
Q:
R:
S:

Example: \\Server\Share name

Path: \\157.163.240.241\LWPC

Figure 11-13 Connecting network drives

Sequences of operation for connecting network drives

1. Place the cursor on a free drive.
2. Change to the "Path" input field using the TAB key.

   Specify the IP address of the server and the sharing name.

   Example: \\157.163.240.241\

Connect

Press "Connect".

The server connection is connected with the drive of the control system.

Note

For example, for executing an external subprogram, please see Chapter "Automatic Mode" -> "Execution from external".

Disconnecting network drives

Disconnect

By selecting the ">>Back" softkey and the "Disconnect" function you can disconnect an existing network connection.

1. Place the cursor on the relevant drive.
2. Press the "Disconnect" softkey.

   The selected network drive is disconnected from the control.

Data backup 12

12.1 Data transfer via RS232 interface

Functionality

The RS232 interface of the control system can be used to output data (e.g. part programs) to an external data backup device or to read in data from there. The RS232 interface and your data backup device must be matched with each other.

Operating sequence

PROGRAM MANAGER

You have selected the <PROGRAM MANAGER> operating area and you are in the overview of the NC programs already created.

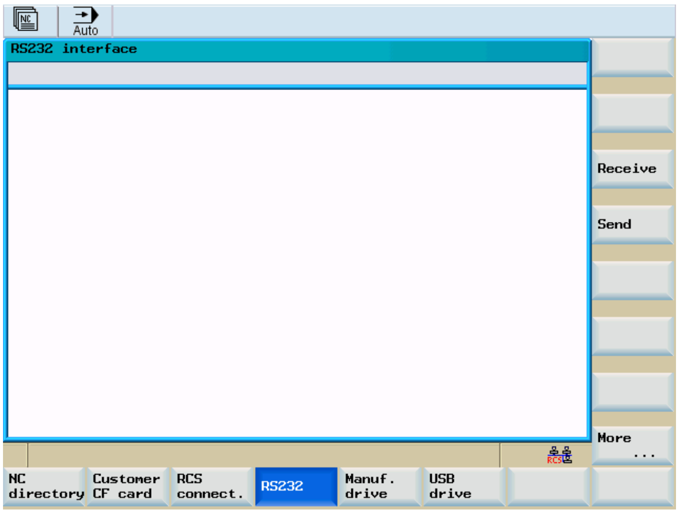
Select the data to be transmitted using either the cursor or the "Select all" softkey,

Copy

and copy the data to the clipboard.

RS232

Press the "RS232" softkey and select the desired transfer mode.



RS232 interface

Figure 12-1 Reading out a program

Send

Press "Send" to start the data transfer. All data copied to the clipboard will be transmitted.

Further softkeys

Receive

Load files via the RS232 interface.

More ...

The following function is provided at this level:

Error log

Transmission protocol
This log contains all transmitted files including status information:

- For files to be output
  : name of file
  error log
- For files to be input
  : name of file and path
  error log

Table 12- 1 Transmission messages

OK	Transmission completed successfully
ERR EOF	End-of-text character received, but archive file incomplete
Time Out	The time monitoring is reporting an interruption of the data transfer
User Abort	Data transfer aborted by the <Stop> softkey
Error Com	Error at the COM 1 port
NC / PLC Error	Error message from the NC
Error Data	Data error 1. Files read in with / without header or 2. Files transmitted without file names in the punched-tape format
Error File Name	The file name does not correspond to the name convention of the NC.

12.2 Creating / reading in / reading out a start-up archive

References

SINUMERIK 802D sl Operating Instructions for Turning, Milling, Grinding, Nibbling; Data Backup and Series Start-Up

Operating sequence

SHIFT + SYSTEM ALARM

Start-up files

Press the "Start-up files" softkey in the <SYSTEM> operating area.

Creating a start-up archive

A start-up archive can be created either with all components or with some selected components.

To create an archive with selected components, the following operator actions are required:

802D data

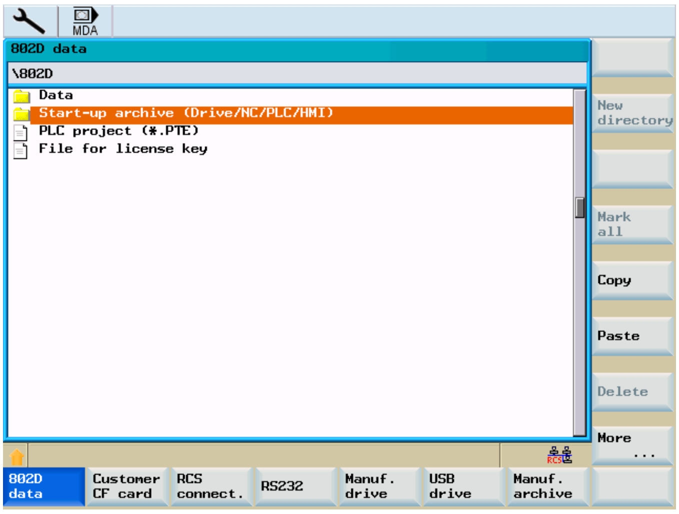
Press "802D data". Please select the line "Start-up archive (drive/NC/PLC/HMI)" using the direction keys.

INPUT

Press the "Input" key to open the directory and select the desired lines using the "Select" key.

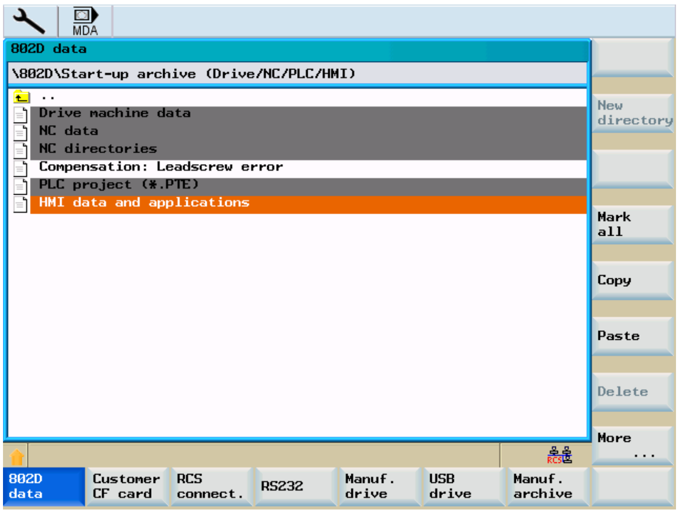
Copy

Press the "Copy" softkey. The files are copied to the clipboard.



```
802D data
\802D
Data
Start-up archive (Drive/NC/PLC/HMI)
PLC project (*.PTE)
File for license key
```

Figure 12-2 Copy entire start-up archive



802D data

\802D\Start-up archive (Drive/NC/PLC/HMI)

- ..
- Drive machine data
- NC data
- NC directories
- Compensation: Leadscrew error
- PLC project (*.PTE)
- HMI data and applications

Figure 12-3 Contents of the start-up archive

SELECT

By pressing the <Select> key, the respective files can be individually selected/deselected in the start-up archive.

Writing the start-up archive to a customer CompactFlash card/USB FlashDrive

Requirement: The CompactFlash Card/USB FlashDrive is inserted, and the start-up archive has been copied to the clipboard.

Operating sequence:

Customer CF card

or

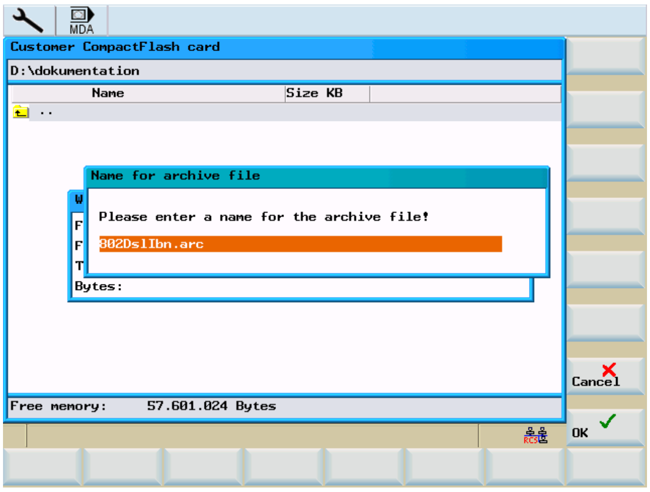
USB drive

Press the "Customer CF card" or "USB drive" softkey. In the directory, select the saving location (directory).

Paste

Use the "Insert" softkey to start writing of the start-up archive.

In the dialog that follows, confirm the name that is specified or enter a new name. Close the dialog box by pressing "OK".



Customer CompactFlash card

D:\dokumentation

Name	Size KB
..	

Name for archive file

Please enter a name for the archive file!

802DslIbn.arc

Bytes:

Free memory: 57.601.024 Bytes

Figure 12-4 Insert files

Reading in start-up archive from customer CompactFlash card/USB FlashDrive

To import a start-up archive, perform the following operator actions:

1. CompactFlash card/USB FlashDrive are inserted
2. Press the "Customer CF card"/"USB drive" softkey and select the line with the desired archive file.
3. Press "Copy" to copy the file to the clipboard.
4. Press the "802D data" softkey and position the cursor on the start-up archive (drive/NC/PLC/HMI) line.
5. Press the "Paste" softkey; commissioning starts.
6. Acknowledge the start dialog on the control system.

12.3 Reading in / reading out PLC projects

When reading in a project, this will be transferred to the file system of the PLC and then activated. To complete the activation, the control system is restarted (warm start).

Reading in project from CompactFlash card/USB FlashDrive

To read in a PLC project, perform the following operator actions:

1. CompactFlash card/USB FlashDrive are inserted
2. Press the "Customer CF card"/"USB drive" softkey and select the line with the desired project file in PTE format.
3. Press "Copy" to copy the file to the clipboard.
4. Press the "802D data" softkey and position the cursor on the **PLC Project (PT802D *.PTE**) line.
5. Press the "Paste" softkey; reading in and activation starts.

Writing project to CompactFlash card/USB FlashDrive

Perform the following operator actions:

1. CompactFlash card/USB FlashDrive are inserted
2. Select the "802D data" softkey and position the direction keys on the **PLC project (PT802D *.PTE)** line.
3. Press "Copy" to copy the file to the clipboard.
4. Press the "Customer CF card"/"USB drive" softkey and select the saving location for the file.
5. Press the "Paste" softkey; the writing process starts.

12.4 Copying and pasting files

In the <PROGRAM MANAGER> operating area and in the "Start-up files" function, files or directories can be copied into another directory or onto a different drive using the softkey functions "Copy" and "Paste". When doing so, the "Copy" function enters the references to the files or directories in a list which is subsequently executed by the "Paste" function. This function will perform the actual copying process.

The list is kept until a new copying process overwrites this list.

Special situation:
If the RS232 interface has been selected as the data target, "Paste" will be replaced by the "Send" softkey function.. When reading in files ("Receive" softkey), it is not necessary to specify a target, since the name of the target directory is not contained in the data flow.

PLC diagnostics 13

Functionality

A PLC user program consists to a large degree of logical operations to realize safety functions and to support process sequences. These logical operations include the linking of various contacts and relays. As a rule, the failure of a single contact or relay results in a failure of the whole system/installation.

To locate causes of faults/failures or of a program error, various diagnostic functions are offered in the "System" operating area.

Operating sequence

SHIFT + SYSTEM ALARM

PLC

Press the "PLC" softkey in the <SYSTEM> operating area.

PLC program

Press "PLC program".

The project stored in the residual memory is opened.

13.1 Screen layout

The screen layout with its division into the main areas corresponds to the layout already described in section "Software Interface".

Any deviations and supplements pertaining to the PLC diagnostics are shown in the following screen.

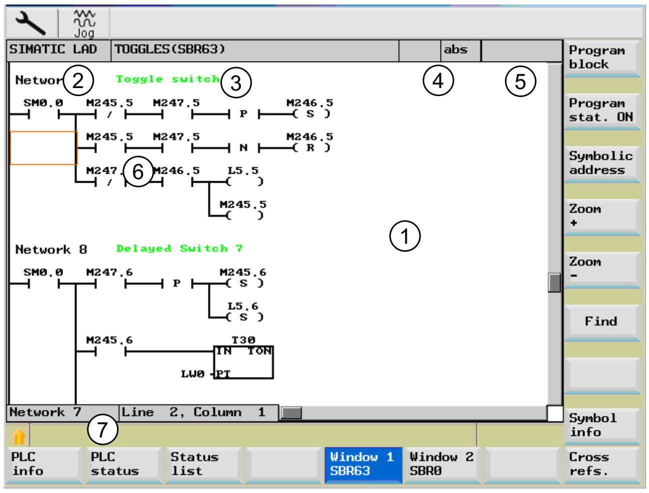


Figure 13-1 Screen layout

Table 13- 1 Key to screen layout

Screen item	Display	Meaning
①	Application area	
②	Supported PLC program language	
③	Name of the active program block Representation: Symbolic name (absolute name)	
④	Program status	
	RUN	Program is running
	STOP	Program stopped
	Status of the application area	
	Sym	Symbolic representation
	abs	Absolute representation
⑤		Display of the active keys
⑥	Focus Performs the tasks of the cursor	
⑦	Tip line contains notes for searching	

13.2 Operating options

In addition to the softkeys and the navigation keys, this area provides still further key combinations.

Hotkeys

The cursor keys move the focus over the PLC user program. When reaching the window borders, it is scrolled automatically.

Table 13- 2 Hotkeys

Keystroke combination	Action
NEXT WINDOW or CTRL ◀	To the first line of the row
END or CTRL ▶	To the last line of the row
PAGE UP	Up a screen
PAGE DOWN	Down a screen
◀	One field to the left
▶	One field to the right
▲	Up a field

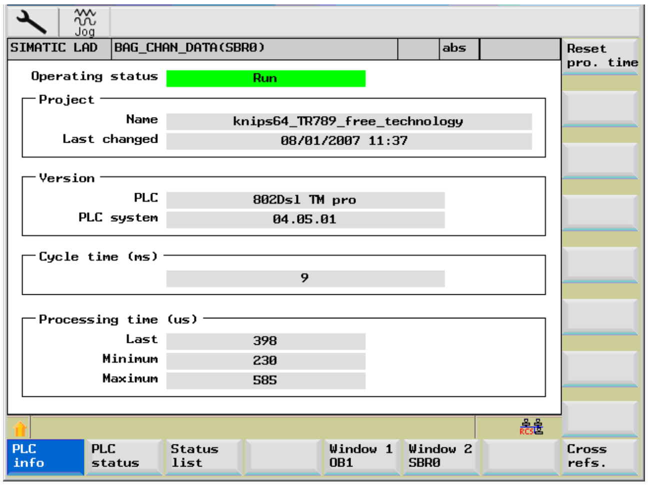
Keystroke combination	Action
▼	Down a field
CTRL NEXT WINDOW or CTRL ▲	To the first field of the first network
CTRL END or CTRL ▼	To the last field of the first network
CTRL PAGE UP	Opens the next program block in the same window
CTRL and PAGE DOWN	Opens the previous program block in the same window
SELECT	The function of the Select key depends on the position of the input focus. • Table line: Displays the complete text line • Network title: Displays the network comment • Command: Displays the complete operands
INPUT	If the input focus is positioned on a command, all operands including the comments are displayed.

Softkeys

PLC
info

The following PLC properties are shown with this softkey:

- Mode
- Name of the PLC project
- PLC system version
- Cycle time
- Machining time of the PLC user program



SIMATIC LAD | BAG_CHAN_DATA(SBR0) | abs

Operating status: Run

Project
Name: knips64_TR789_free_technology
Last changed: 08/01/2007 11:37

Version
PLC: 802Dsl TM pro
PLC system: 04.05.01

Cycle time (ms)
9

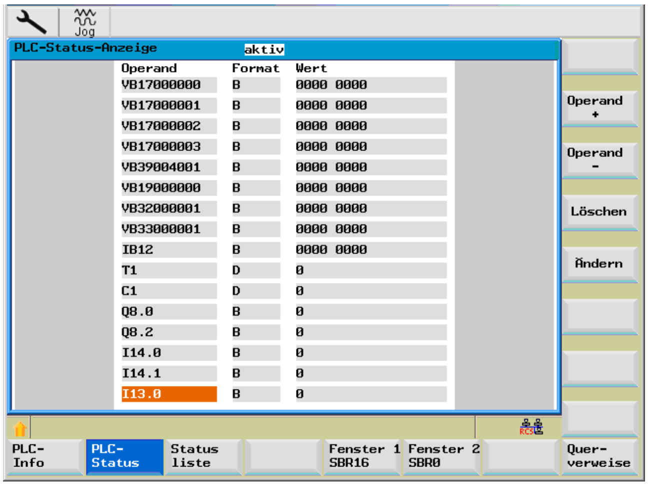
Processing time (us)
Last: 398
Minimum: 230
Maximum: 585

Figure 13-2 PLC info

By pressing the "Reset machining time" softkey, machining time data is reset.

PLC
status

The values of the operands can be monitored and changed during program execution using the "PLC status display" window.



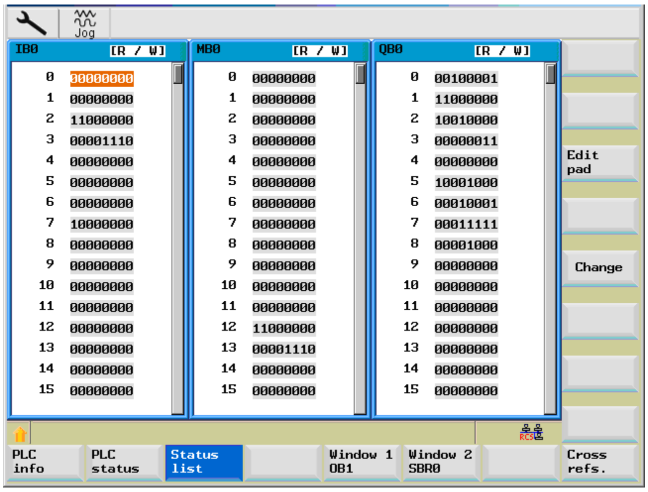
PLC-Status-Anzeige aktiv

Operand	Format	Wert
VB17000000	B	0000 0000
VB17000001	B	0000 0000
VB17000002	B	0000 0000
VB17000003	B	0000 0000
VB39004001	B	0000 0000
VB19000000	B	0000 0000
VB32000001	B	0000 0000
VB33000001	B	0000 0000
IB12	B	0000 0000
T1	D	0
C1	D	0
Q8.0	B	0
Q8.2	B	0
I14.0	B	0
I14.1	B	0
I13.0	B	0

Figure 13-3 PLC status display

Status list

Use the "Status list" softkey to display and modify PLC signals.



IB0 [R / W]		MB0 [R / W]		QB0 [R / W]	
0	00000000	0	00000000	0	00100001
1	00000000	1	00000000	1	11000000
2	11000000	2	00000000	2	10010000
3	00001110	3	00000000	3	00000011
4	00000000	4	00000000	4	00000000
5	00000000	5	00000000	5	10001000
6	00000000	6	00000000	6	00010001
7	10000000	7	00000000	7	00011111
8	00000000	8	00000000	8	00001000
9	00000000	9	00000000	9	00000000
10	00000000	10	00000000	10	00000000
11	00000000	11	00000000	11	00000000
12	00000000	12	11000000	12	00000000
13	00000000	13	00001110	13	00000000
14	00000000	14	00000000	14	00000000
15	00000000	15	00000000	15	00000000

Figure 13-4 Status list

Window 1 OB1

Using the "Window 1 ..." and "Window 2 ..." softkeys you can display any logical and graphical information of a program block. The program block is one of the components of the PLC user program.

The program block can be selected in the "Program list" using the "Open" softkey. The name of the program block will be displayed on the softkey (for "..." e.g. "Window 1 SBR16").

The logics in the ladder diagram (LAD) display the following:

- Networks with program parts and current paths
- Electrical current flow through a number of logical operations

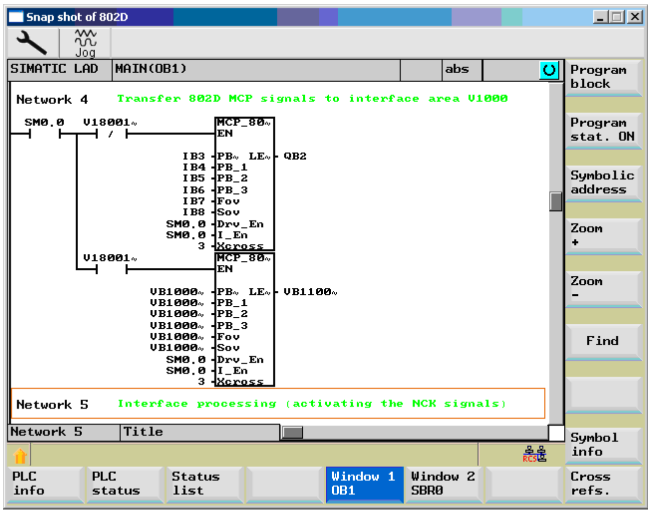
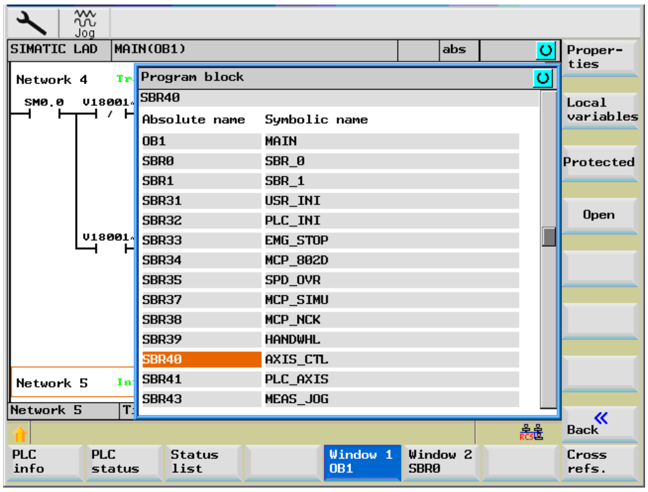


Figure 13-5 Window 1, OB1

Program block

This softkey can be used to select the list of the PLC program blocks.



Program block

SBR40

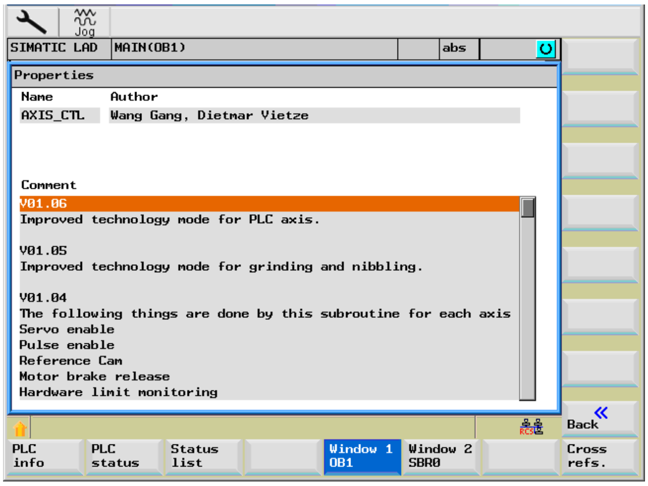
Absolute name	Symbolic name
OB1	MAIN
SBR0	SBR_0
SBR1	SBR_1
SBR31	USR_INI
SBR32	PLC_INI
SBR33	EMG_STOP
SBR34	MCP_802D
SBR35	SPD_OVR
SBR37	MCP_SIMU
SBR38	MCP_NCK
SBR39	HANDWHL
SBR40	AXIS_CTL
SBR41	PLC_AXIS
SBR43	MEAS_JOG

Figure 13-6 Select the PLC program block

Proper-ties

Using this softkey will display the following properties of the selected program block:

- Symbolic name
- Author
- Comments



Properties

Name	Author
AXIS_CTL	Wang Gang, Dietmar Vietze

Comment

```
V01.06
Improved technology mode for PLC axis.

V01.05
Improved technology mode for grinding and nibbling.

V01.04
The following things are done by this subroutine for each axis
Servo enable
Pulse enable
Reference Cam
Motor brake release
Hardware limit monitoring
```

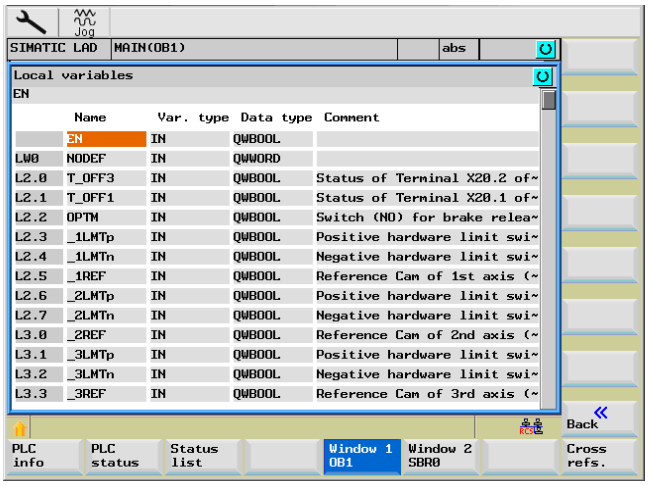
Figure 13-7 Properties of the selected PLC program block

Local variables

Selecting this softkey displays the table of local variables of the selected program block.

There are two types of program blocks.

- OB1 only temporary local variable
- SBRxx temporary local variable



Local variables

EN

	Name	Var. type	Data type	Comment
	EN	IN	QWBOOL	
LW0	NODEF	IN	QWWORD	
L2.0	T_OFF3	IN	QWBOOL	Status of Terminal X20.2 of~
L2.1	T_OFF1	IN	QWBOOL	Status of Terminal X20.1 of~
L2.2	OPTM	IN	QWBOOL	Switch (NO) for brake relea~
L2.3	_1LMTp	IN	QWBOOL	Positive hardware limit swi~
L2.4	_1LMTn	IN	QWBOOL	Negative hardware limit swi~
L2.5	_1REF	IN	QWBOOL	Reference Cam of 1st axis (~
L2.6	_2LMTp	IN	QWBOOL	Positive hardware limit swi~
L2.7	_2LMTn	IN	QWBOOL	Negative hardware limit swi~
L3.0	_2REF	IN	QWBOOL	Reference Cam of 2nd axis (~
L3.1	_3LMTp	IN	QWBOOL	Positive hardware limit swi~
L3.2	_3LMTn	IN	QWBOOL	Negative hardware limit swi~
L3.3	_3REF	IN	QWBOOL	Reference Cam of 3rd axis (~

Figure 13-8 Table of local variables for the selected PLC program block

The text of the current cursor position is additionally displayed in a text field above the table.

With longer texts, it is possible to display the whole text by pressing the SELECT key.

Cover

When a program block is protected by a password, this softkey can be used to enable the display of the ladder diagram.

A password is required for this. The password can be allocated during creation of a program block in Programming Tool PLC802.

Open

The selected program block is opened.

The name (absolute) of the program block will then be displayed on "Window 1..." softkey (for "..." e.g. "Window 1 OB1").

Program stat. OFF

Selecting this softkey activates or deactivates the program status display.

You can monitor the current status of the networks from the PLC cycle end.

The states of all operands are displayed in the "Program status" ladder diagram (top right in the window). This LAD acquires the values for the status display in several PLC cycles and then refreshes the status display.

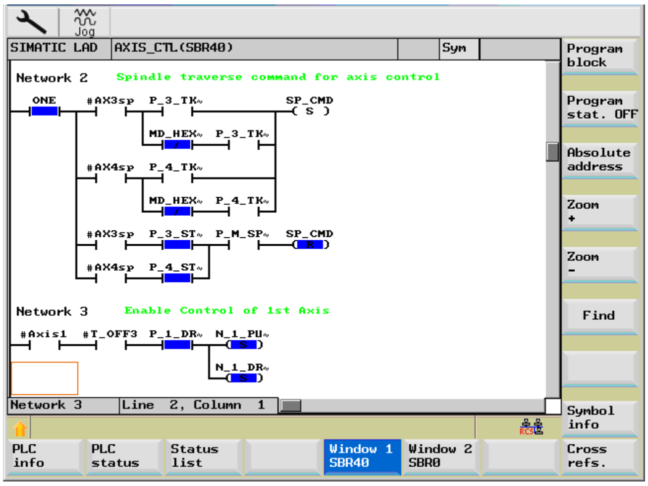


Figure 13-9 "Program status" ON – symbolic representation

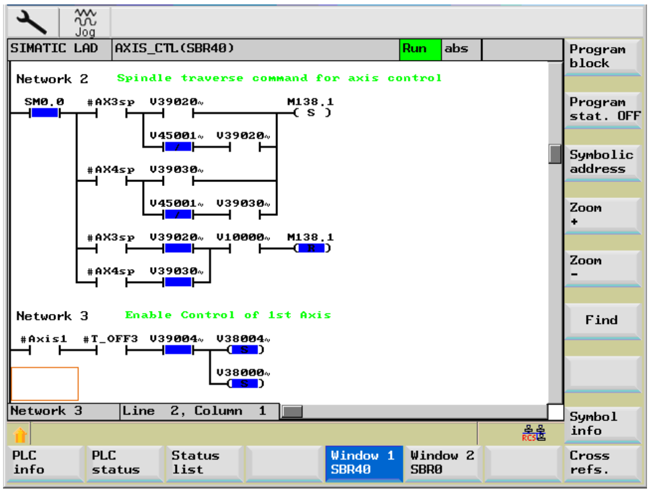


Figure 13-10 "Program status" ON – absolute representation

Symbolic address

Use this softkey to switch between the absolute and symbolic representation of the operands. The softkey labelling changes accordingly.

Depending on the selected type of representation, the operands are displayed either with absolute or symbolic identifiers.

If no symbol exists for a variable, this is automatically displayed absolutely.

Zoom +

Zoom -

The representation in the application area can be zoomed in or zoomed out step by step. The following zoom stages are provided:

20% (default), 60%, 100% and 300%

Find

Can be used to search for operands in the symbolic or absolute representation (see following screen).

A dialog box is displayed from which various search criteria can be selected. Use the "Absolute/symbol. address" softkey to search for a certain operand matching this criterion in both PLC windows (see the following screen). When searching, uppercase and lowercase letters are ignored.

Selection in the upper toggle field:

- Search for absolute and symbolic operands
- Go to network number
- Find SBR command

Further search criteria:

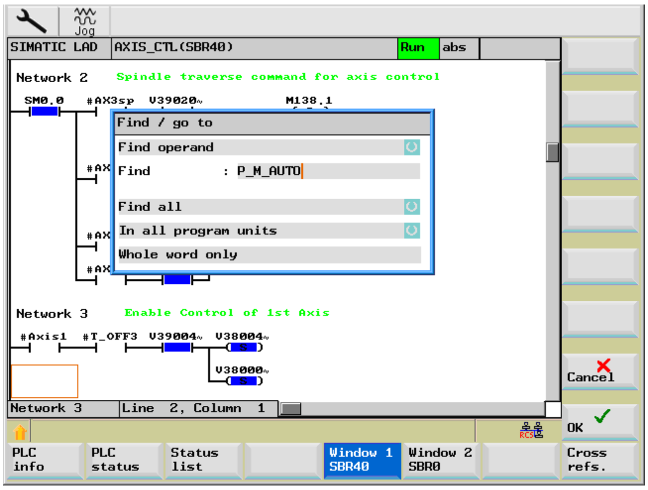
- Search direction down (from the current cursor position)
- Whole program block (from the beginning)
- In one program block
- Over all program blocks

You can search for the operands and constants as whole words (identifiers).

Depending on the display settings, you can search for symbolic or absolute operands.

"OK" starts the search. The found search element is highlighted by the focus. If nothing is found, an appropriate error message will appear in the notes line.

Use the "Abort" softkey to exit the dialog box. no search is carried out.



Find / go to

Find operand

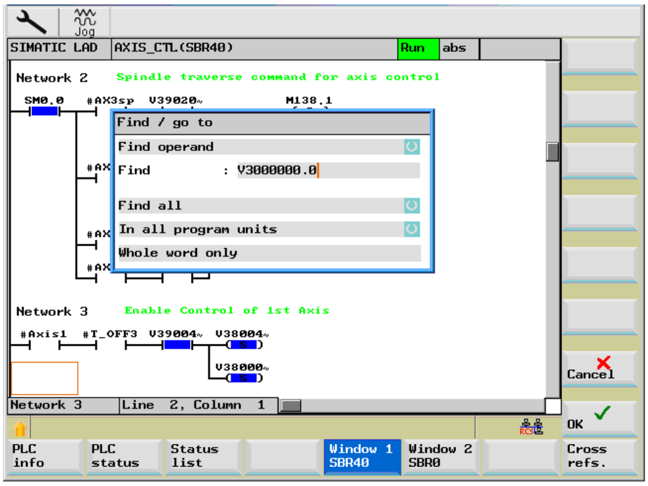
Find : P_M_AUTO

Find all

In all program units

Whole word only

Figure 13-11 Search for symbolic operands



SIMATIC LAD | AXIS_CTL(SBR40) | Run | abs

Network 2 Spindle traverse command for axis control

Find / go to

Find operand ○

Find : V3000000.0

Find all ○

In all program units ○

Whole word only

Network 3 Enable Control of 1st Axis

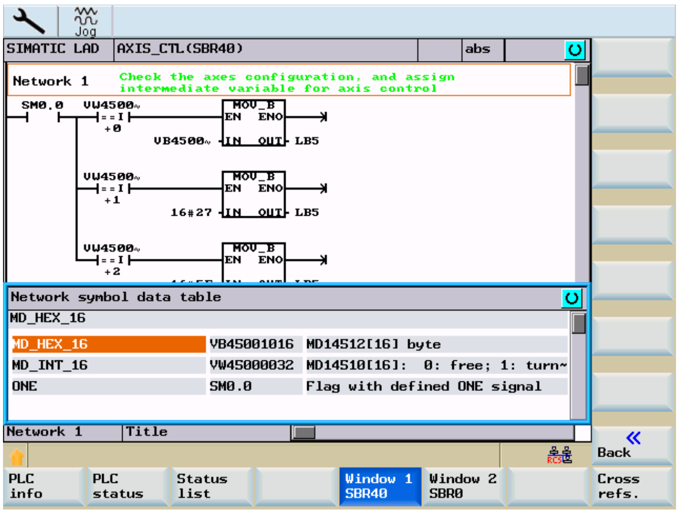
Network 3 | Line 2, Column 1

Figure 13-12 Search for absolute operands

If the search object is found, use the "Continue search" softkey to continue the search.

Symbol info

Selecting this softkey displays all symbolic identifiers used in the highlighted network.



SIMATIC LAD | AXIS_CTL(SBR40) | abs

Network 1 Check the axes configuration, and assign intermediate variable for axis control

Network symbol data table

MD_HEX_16

MD_HEX_16	VB45001016	MD14512[16] byte
MD_INT_16	VW45000032	MD14510[16]: 0: free; 1: turn~
ONE	SM0.0	Flag with defined ONE signal

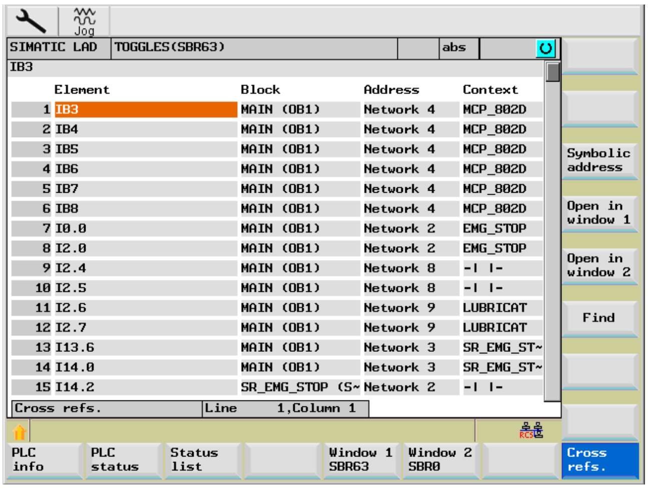
Network 1 | Title

Figure 13-13 Network symbol information table

Cross refs.

Use this softkey to display the list of cross references. All operands used in the PLC project are displayed.

This list indicates in which networks an input, output, flag etc. is used.



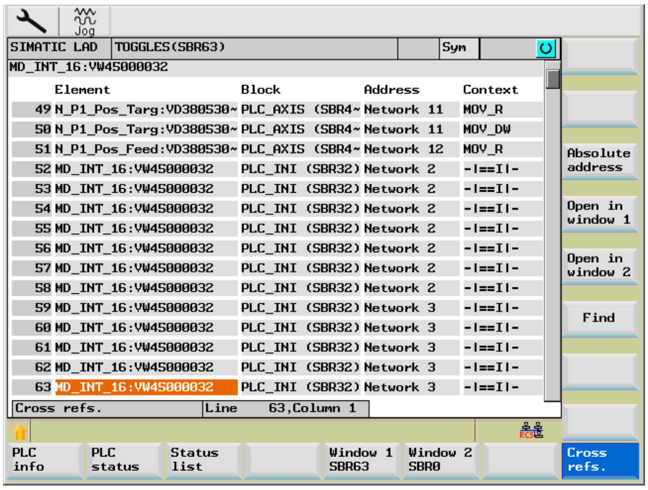
SIMATIC LAD | TOGGLES(SBR63) | abs

IB3

	Element	Block	Address	Context
1	IB3	MAIN (OB1)	Network 4	MCP_802D
2	IB4	MAIN (OB1)	Network 4	MCP_802D
3	IB5	MAIN (OB1)	Network 4	MCP_802D
4	IB6	MAIN (OB1)	Network 4	MCP_802D
5	IB7	MAIN (OB1)	Network 4	MCP_802D
6	IB8	MAIN (OB1)	Network 4	MCP_802D
7	I0.0	MAIN (OB1)	Network 2	EMG_STOP
8	I2.0	MAIN (OB1)	Network 2	EMG_STOP
9	I2.4	MAIN (OB1)	Network 8	-I I-
10	I2.5	MAIN (OB1)	Network 8	-I I-
11	I2.6	MAIN (OB1)	Network 9	LUBRICAT
12	I2.7	MAIN (OB1)	Network 9	LUBRICAT
13	I13.6	MAIN (OB1)	Network 3	SR_EMG_ST~
14	I14.0	MAIN (OB1)	Network 3	SR_EMG_ST~
15	I14.2	SR_EMG_STOP (S~	Network 2	-I I-

Cross refs. | Line 1,Column 1

Figure 13-14 Cross references main menu (absolute)



SIMATIC LAD | TOGGLES(SBR63) | Sym

MD_INT_16:VW45000032

	Element	Block	Address	Context
49	N_P1_Pos_Targ:VD380530~	PLC_AXIS (SBR4~	Network 11	MOV_R
50	N_P1_Pos_Targ:VD380530~	PLC_AXIS (SBR4~	Network 11	MOV_DW
51	N_P1_Pos_Feed:VD380530~	PLC_AXIS (SBR4~	Network 12	MOV_R
52	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 2	-I==I I-
53	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 2	-I==I I-
54	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 2	-I==I I-
55	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 2	-I==I I-
56	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 2	-I==I I-
57	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 2	-I==I I-
58	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 2	-I==I I-
59	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 3	-I==I I-
60	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 3	-I==I I-
61	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 3	-I==I I-
62	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 3	-I==I I-
63	MD_INT_16:VW45000032	PLC_INI (SBR32)	Network 3	-I==I I-

Cross refs. | Line 63,Column 1

Figure 13-15 Cross references main menu (symbolic))

Open in window 1

You can open the appropriate program segment directly in the 1/2 window using the "Open in Window 1" or "Open in Window 2" function.

Symbolic address

Use this softkey to switch between the absolute and symbolic representation of the components. The softkey labelling changes accordingly.

Depending on the selected type of representation, the components are displayed either with absolute or symbolic identifiers.

If no symbol exists for an identifier, the description is automatically absolute.

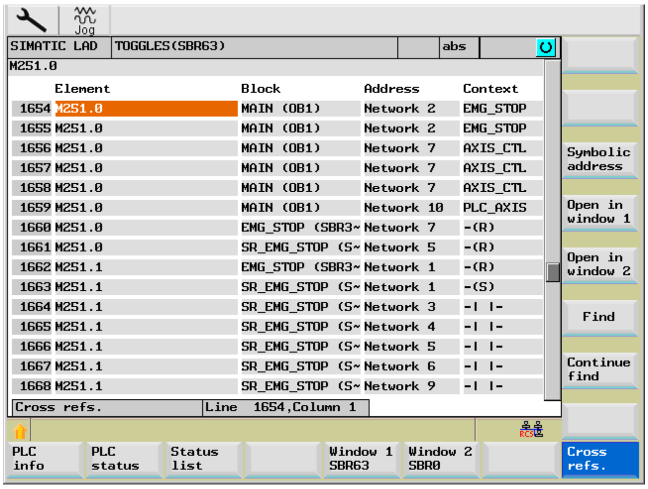
The type of representation is displayed in the status line at the top right of the window (e.g. "Abs"). The absolute representation is set by default.

Example:

You want to view the logic interrelation of the absolute operand M251.0 in network 2 in program block OB1.

After the operand has been selected from the cross-reference list and the "Open in Window 1" softkey has been pressed, the corresponding program section is displayed in window 1.

Cross refs.



SIMATIC LAD | TOGGLES(SBR63) | abs

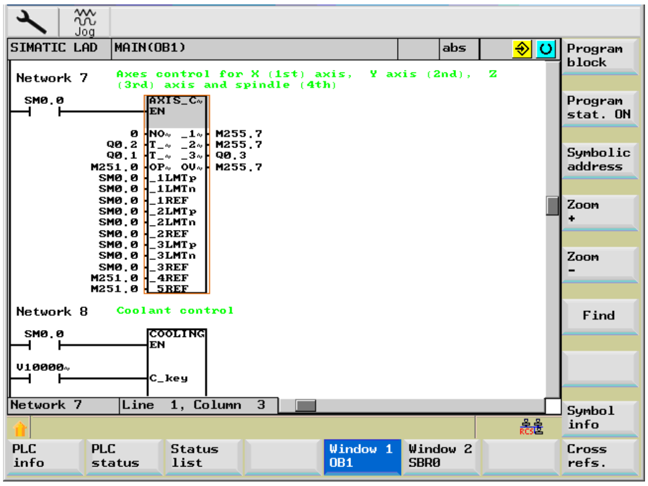
M251.0

	Element	Block	Address	Context
1654	M251.0	MAIN (OB1)	Network 2	EMG_STOP
1655	M251.0	MAIN (OB1)	Network 2	EMG_STOP
1656	M251.0	MAIN (OB1)	Network 7	AXIS_CTL
1657	M251.0	MAIN (OB1)	Network 7	AXIS_CTL
1658	M251.0	MAIN (OB1)	Network 7	AXIS_CTL
1659	M251.0	MAIN (OB1)	Network 10	PLC_AXIS
1660	M251.0	EMG_STOP (SBR3~	Network 7	-(R)
1661	M251.0	SR_EMG_STOP (S~	Network 5	-(R)
1662	M251.1	EMG_STOP (SBR3~	Network 1	-(R)
1663	M251.1	SR_EMG_STOP (S~	Network 1	-(S)
1664	M251.1	SR_EMG_STOP (S~	Network 3	-I I-
1665	M251.1	SR_EMG_STOP (S~	Network 4	-I I-
1666	M251.1	SR_EMG_STOP (S~	Network 5	-I I-
1667	M251.1	SR_EMG_STOP (S~	Network 6	-I I-
1668	M251.1	SR_EMG_STOP (S~	Network 9	-I I-

Cross refs. | Line 1654,Column 1

Figure 13-16 Cursor M251.0 in OB1 network 2

Open in window 1



SIMATIC LAD | MAIN(OB1) | abs

Network 7 Axes control for X (1st) axis, Y axis (2nd), Z (3rd) axis and spindle (4th)

Network 8 Coolant control

Network 7 | Line 1, Column 3

Figure 13-17 M251.0 in OB1 network 2 in window 1

Find

Searching operands in the cross-reference list (see following screen).

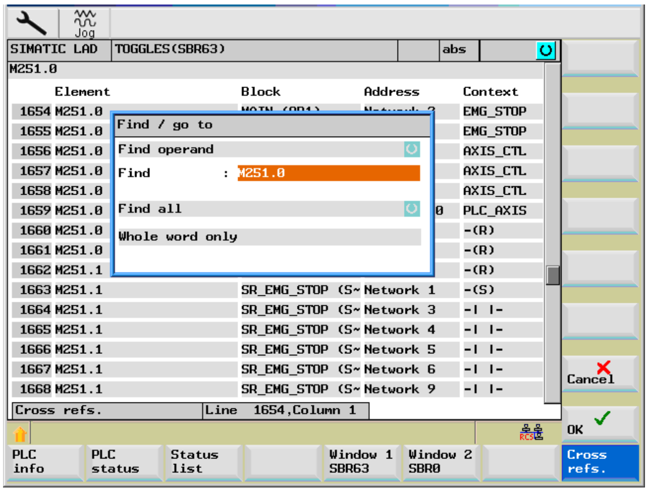
You can search for the operands as whole words (identifiers). When searching, uppercase and lowercase letters are ignored.

Search options:

- Search for absolute and symbolic operands
- Go to line

Search criteria:

- Down (from the current cursor position)
- Whole program block (from the beginning)



SIMATIC LAD | TOGGLES(SBR63) | abs

M251.0

	Element	Block	Address	Context
1654	M251.0	MAIN (OB1)	Network 2	EMG_STOP
1655	M251.0			EMG_STOP
1656	M251.0			AXIS_CTL
1657	M251.0			AXIS_CTL
1658	M251.0			AXIS_CTL
1659	M251.0		0	PLC_AXIS
1660	M251.0			-(R)
1661	M251.0			-(R)
1662	M251.1			-(R)
1663	M251.1	SR_EMG_STOP (S~	Network 1	-(S)
1664	M251.1	SR_EMG_STOP (S~	Network 3	-I I-
1665	M251.1	SR_EMG_STOP (S~	Network 4	-I I-
1666	M251.1	SR_EMG_STOP (S~	Network 5	-I I-
1667	M251.1	SR_EMG_STOP (S~	Network 6	-I I-
1668	M251.1	SR_EMG_STOP (S~	Network 9	-I I-

Find / go to

Find operand

Find : M251.0

Find all

Whole word only

Cross refs. | Line 1654,Column 1

Figure 13-18 Searching for operands in cross references

The text you are looking for is displayed in the notes line. If the text is not found, a corresponding error message is displayed which must be confirmed with "OK".

Appendix A

A.1 Miscellaneous

A.1.1 Pocket calculator

⇧ SHIFT

=

The calculator function can be activated from any operating area using <SHIFT> and <=> or <CTRL> and <A>.

For calculating, the four basic arithmetic operations are available, as well as the functions "sine", "cosine", "squaring" and "square root". A bracket function is provided to calculate nested terms. The bracket depth is unlimited.

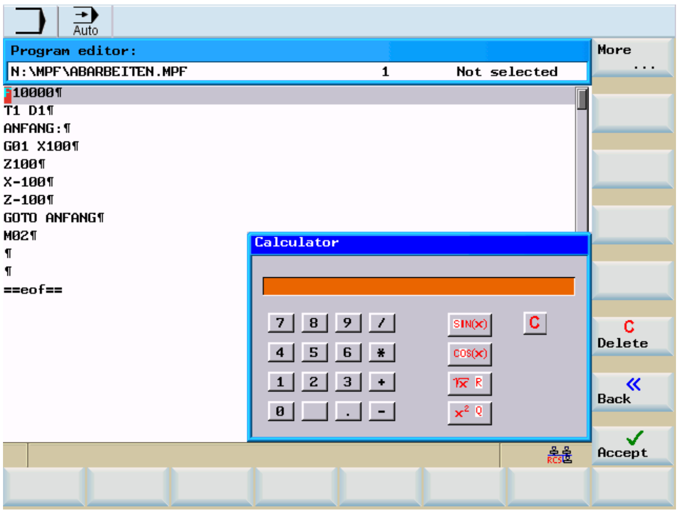
If the input field is already occupied by a value, the function will accept this value into the input line of the pocket calculator.

<Input> starts the calculation. The result is displayed in the pocket calculator.

Selecting the "Accept" softkey enters the result in the input field at the current cursor position of the part program editor and closes the pocket calculator automatically.

Note

If an input field is in the editing mode, it is possible to restore the original status using the "Toggle" key.



Program editor:
N:\MPF\ABARBEITEN.MPF 1 Not selected

```
F10000¶
T1 D1¶
ANFANG:¶
G01 X100¶
Z100¶
X-100¶
Z-100¶
GOTO ANFANG¶
M02¶
¶
¶
==eof==
```

Figure A-1 Pocket calculator

Characters that may be entered

+, -, *, / Basic arithmetic operations

S Sine function
The X value (in degrees) in front of the input cursor is replaced by the sin(X) value.

O Cosine function
The X value (in degrees) in front of the input cursor is replaced by the cos(X) value.

Q Square root function
The X value in front of the input cursor is replaced by the X^2 value.

R Square root function
The X value in front of the input cursor is replaced by the √X value.

( ) Bracket function (X+Y)*Z

Calculation examples

Task	Input -> Result
100 + (67*3)	100+67*3 -> 301
sin(45_)	45 S -> 0.707107
cos(45_)	45 O -> 0.707107
4^2	4 Q -> 16
√4	4 R -> 2
(34+3*2)*10	(34+3*2)*10 -> 400

To calculate auxiliary points on a contour, the pocket calculator offers the following functions:

- Calculating the tangential transition between a circle sector and a straight line
- Moving a point in the plane

- Converting polar coordinates to Cartesian coordinates
- Adding the second end point of a straight line/straight line contour section given from an angular relation

A.1.2 Editing Asian characters

The program editor and PLC alarm text editor both allow you to edit Asian characters.

This function is available in the following Asian language versions:

- Simplified Chinese
- Traditional Chinese (as used in Taiwan)
- Korean

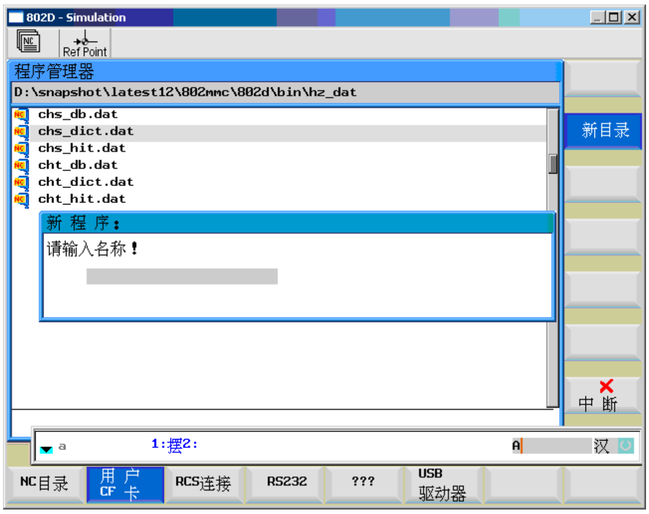
Press <Alt+S> to switch the editor on or off.

Simplified/Traditional Chinese

Characters can be selected according to the pinyin input method, which involves combining letters of the Roman alphabet in order to reproduce the sound of the character.

The editor will then show a list of characters that correspond to that particular sound.

You can then select the character you need.



802D - Simulation

程序管理器

D:\snapshot\latest12\802mmc\802d\bin\hz_dat

chs_db.dat
chs_dict.dat
chs_hit.dat
cht_db.dat
cht_dict.dat
cht_hit.dat

新 程 序：

请输入名称！

Figure A-2 Example of editing Simplified Chinese

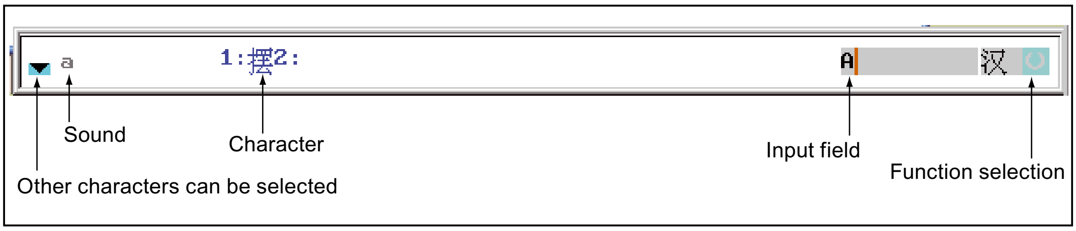
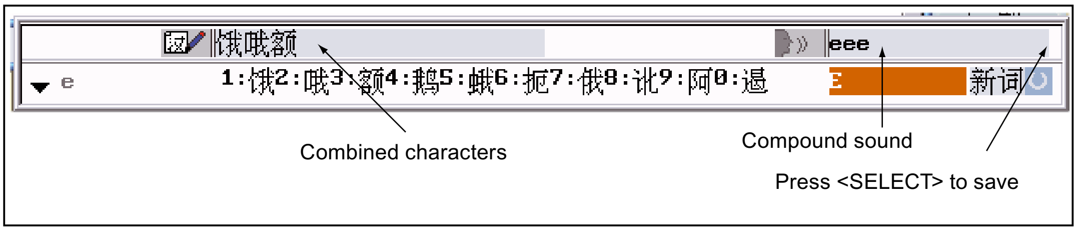


Figure A-3 Structure of editor

The "Function selection" toggle field enables switching between the PinYin-entry method and entering Latin graphic characters as well as activating the function for editing the dictionary.

When a character is selected, the editor records the frequency with which it is selected for a specific phonetic notation and when the editor is again opened, it offers the most frequently used characters.



饿哦额

eee

e 1:饿2:哦3:额4:鹅5:蛾6:扼7:俄8:讹9:阿0:遏

新词

Figure A-4 Structure of editor when learning function is active

- Editing the dictionary

  When this function is activated, another line showing the combined characters and sounds will appear.

  The editor will then offer various characters for this sound, from which you can choose the desired one by entering either of the digits (1 to 9).

  You can toggle the input cursor between the compound phonetic notations field and the phonetic input field by pressing the <TAB> key.

  When the cursor is positioned in the upper field, you can undo the combination by pressing the <backspace> key.

  Press <select> to save the characters currently being displayed.

  Press the <delete> key if you want to delete the currently displayed group of characters from the dictionary.

Korean

To enter Korean characters, you will need a keyboard with the keyboard assignment shown below.

In terms of key layout, this keyboard is the equivalent of an English QWERTY keyboard and individual characters must be grouped together to form syllabic blocks.

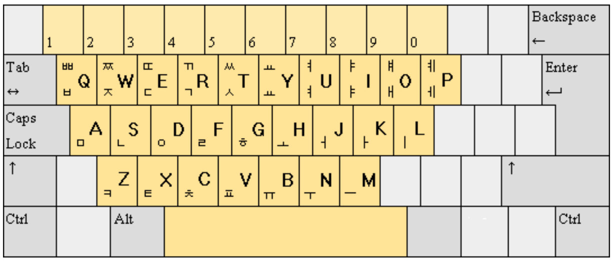
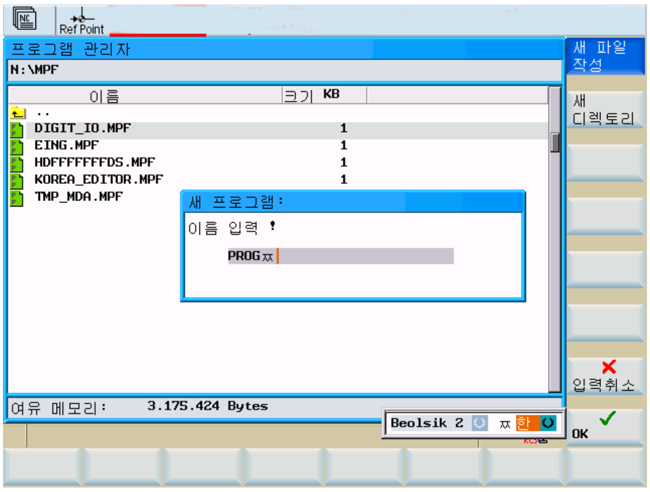


Figure A-5 Korean keyboard assignment

The Korean alphabet (Hangeul) consists of 24 letters: 14 consonants and 10 vowels. The syllable blocks are created by combining consonants and vowels.



프로그램 관리자

N:\MPF

이름	크기 KB
..	
DIGIT_IO.MPF	1
EING.MPF	1
HDFFFFFFFDS.MPF	1
KOREA_EDITOR.MPF	1
TMP_MDA.MPF	

새 프로그램:

이름 입력 !

PROGㅉ

여유 메모리: 3.175.424 Bytes

Figure A-6 Korean editor with standard keyboard assignment

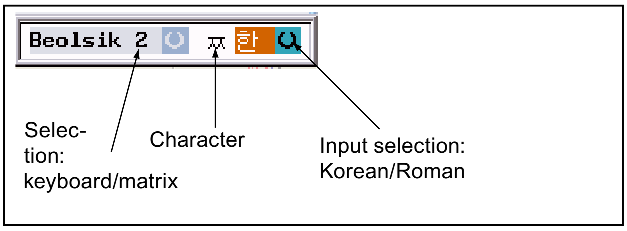
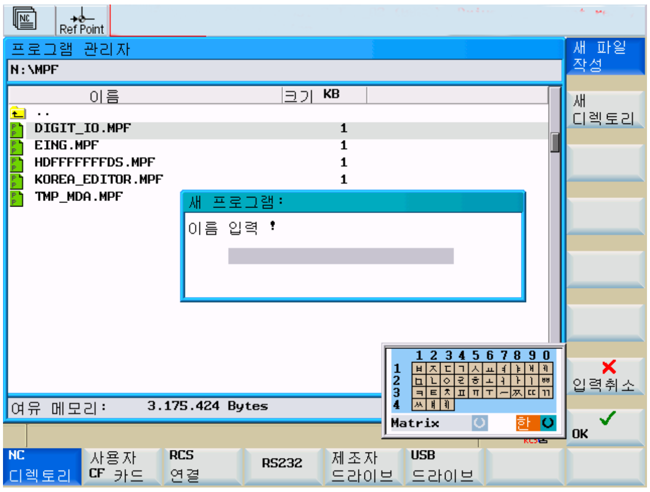


Figure A-7 Structure of Korean editor

- Input via matrix

  If you only have access to a control keyboard, then you can use a matrix input method as an alternative to the keyboard assignment shown above. All you will need for this is the numeric keypad.



프로그램 관리자

N:\MPF

이름	크기 KB
..	
DIGIT_IO.MPF	1
EING.MPF	1
HDFFFFFFFDS.MPF	1
KOREA_EDITOR.MPF	1
TMP_MDA.MPF	

새 프로그램:

이름 입력 !

여유 메모리: 3.175.424 Bytes

Figure A-8 Korean editor with selection matrix

To select characters, proceed as follows:

- Select a row (the row will be color-highlighted)
- Select a column (the character will briefly be color-highlighted and then transferred to the "Character" field).
- Press the <input> key to transfer the character into the edit field.

A.2 Feedback on the documentation

This document will be continuously improved with regard to its quality and ease of use. Please help us with this task by sending your comments and suggestions for improvement via e-mail or fax to:

E-mail: mailto:docu.motioncontrol@siemens.com

Fax: +49 9131 - 98 2176
Please use the fax form on the back of this page.

To
SIEMENS AG
I DT MC MS1
P.O. Box 3180

D-91050 Erlangen / Germany

Fax: +49 9131 - 98 2176 (Documentation)

From

Name:

Address of your company/department

Street:

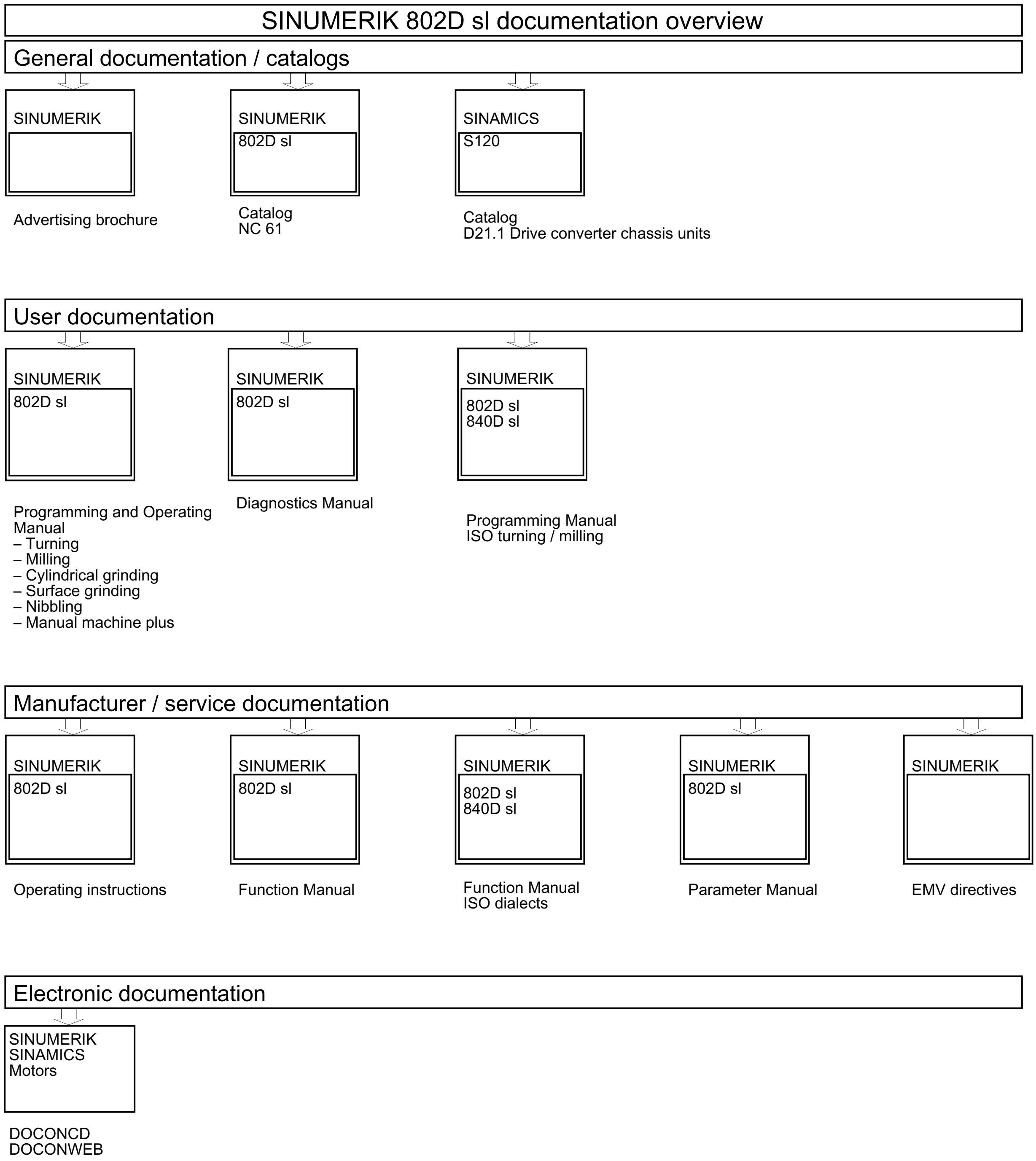
Zip code: City:

Phone: /

Fax: /

Suggestions and/or corrections

A.3 Overview of documentation



SINUMERIK 802D sl documentation overview

General documentation / catalogs

SINUMERIK
Advertising brochure

SINUMERIK 802D sl
Catalog
NC 61

SINAMICS S120
Catalog
D21.1 Drive converter chassis units

User documentation

SINUMERIK 802D sl
Programming and Operating Manual
– Turning
– Milling
– Cylindrical grinding
– Surface grinding
– Nibbling
– Manual machine plus

SINUMERIK 802D sl
Diagnostics Manual

SINUMERIK 802D sl 840D sl
Programming Manual
ISO turning / milling

Manufacturer / service documentation

SINUMERIK 802D sl
Operating instructions

SINUMERIK 802D sl
Function Manual

SINUMERIK 802D sl 840D sl
Function Manual
ISO dialects

SINUMERIK 802D sl
Parameter Manual

SINUMERIK
EMV directives

Electronic documentation

SINUMERIK
SINAMICS
Motors

DOCONCD
DOCONWEB

Index