

# SIEMENS

## SINUMERIK

### SINUMERIK 802D sl Milling

#### Programming and Operating Manual

Valid for

Controller  
802D sl T/M

Software version SINUMERIK  
1.4


06/2009  
6FC5398-0CP10-5BA0


Preface	
Description	1
Software interface	2
Turning On, Reference Point Approach	3
Set up	4
Manually Controlled Mode	5
Automatic mode	6
Part Programming	7
System	8
Programming	9
Cycles	10
Network operation	11
Data backup	12
PLC diagnostics	13
Appendix	A


## 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.

 <b>DANGER</b>
indicates that death or severe personal injury <b>will</b> result if proper precautions are not taken.

 <b>WARNING</b>
indicates that death or severe personal injury <b>may</b> result if proper precautions are not taken.

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

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

<b>NOTICE</b>
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:

 <b>WARNING</b>
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.

# Preface

## 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:

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

	<b>America</b>
<b>Phone</b>	+1 423 262 2522
<b>Fax</b>	+1 423 262 2200
<b>E-mail</b>	<a href="mailto:techsupport.sea@siemens.com">mailto:techsupport.sea@siemens.com</a>

	<b>Asia/Pacific</b>
<b>Phone</b>	+86 1064 757575
<b>Fax</b>	+86 1064 747474
<b>E-mail</b>	<a href="mailto:support.asia.automation@siemens.com">mailto:support.asia.automation@siemens.com</a>

---

### 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

	<b>Preface .....</b>	<b>3</b>
<b>1</b>	<b>Description.....</b>	<b>13</b>
1.1	Control and display elements.....	13
1.2	Error and status displays .....	14
1.3	Key definition of the full CNC keyboard (vertical format).....	15
1.4	Key definition of the machine control panel.....	17
1.5	Coordinate systems .....	19
<b>2</b>	<b>Software interface.....</b>	<b>23</b>
2.1	Screen layout.....	23
2.2	Standard softkeys .....	27
2.3	Operating areas .....	28
2.4	The help system.....	30
<b>3</b>	<b>Turning On, Reference Point Approach.....</b>	<b>33</b>
3.1	Turning on and approaching reference points.....	33
<b>4</b>	<b>Set up.....</b>	<b>35</b>
4.1	Entering tools and tool offsets.....	36
4.1.1	Create new tool.....	40
4.1.2	Determining the tool offsets (manually) .....	42
4.1.3	Determining tool offsets using a probe (auto).....	46
4.1.4	Probe settings .....	49
4.2	Tool monitoring .....	51
4.3	Entering/modifying a work offset.....	54
4.3.1	Determining the work offset .....	55
4.4	Program setting data.....	58
4.5	Arithmetic parameter R.....	62
<b>5</b>	<b>Manually Controlled Mode .....</b>	<b>63</b>
5.1	Manually Controlled Mode .....	63
5.2	JOG mode - "Position" operating area.....	65
5.2.1	Assigning handwheels .....	69
5.3	MDA mode (manual input) "Position" operating area .....	70
5.3.1	Teach In .....	73
5.3.2	Face milling.....	77
<b>6</b>	<b>Automatic mode.....</b>	<b>79</b>
6.1	AUTOMATIC mode.....	79
6.2	Selection and start of a part program .....	84

6.3	Block search.....	86
6.4	Simultaneous recording .....	88
6.5	Stop / cancel a part program.....	91
6.6	Reapproach after cancellation .....	92
6.7	Repositioning after interruption .....	93
6.8	Execute from external .....	94
<b>7</b>	<b>Part Programming.....</b>	<b>97</b>
7.1	Part programming overview .....	97
7.2	Enter new program.....	101
7.3	Edit the part program .....	102
7.4	Simulation.....	105
7.5	Calculate contour elements.....	109
7.6	Free contour programming.....	118
7.6.1	Program a contour .....	121
7.6.2	Define a start point.....	123
7.6.3	Softkeys and parameters .....	125
7.6.4	Parameterize contour element.....	130
7.6.5	Graphic representation of the contour .....	133
7.6.6	Specify contour elements in polar coordinates, close the contour .....	134
7.6.7	Parameter description of straight line/circle contour elements .....	137
7.6.8	Programming example for milling .....	139
<b>8</b>	<b>System.....</b>	<b>145</b>
8.1	"System" operating area .....	145
8.2	SYSTEM - "Start-up" softkeys.....	150
8.3	SYSTEM - "Machine data" softkeys.....	151
8.4	SYSTEM - "Service display" .....	158
8.4.1	Action log.....	159
8.4.2	Servo trace.....	160
8.4.3	Version/HMI details .....	164
8.4.4	Service MSG .....	168
8.5	SYSTEM - "PLC" softkeys .....	175
8.6	SYSTEM - "Start-up files" softkeys .....	183
8.7	Alarm display.....	188
<b>9</b>	<b>Programming .....</b>	<b>191</b>
9.1	Fundamental Principles of NC Programming.....	191
9.1.1	Program names.....	191
9.1.2	Program structure .....	191
9.1.3	Word structure and address.....	192
9.1.4	Block format .....	193
9.1.5	Fonts .....	195
9.1.6	List of instructions .....	196
9.2	Positional data.....	212
9.2.1	Programming dimensions .....	212

9.2.2	Plane selection: G17 to G19 .....	213
9.2.3	Absolute/incremental dimensioning: G90, G91, AC, IC .....	215
9.2.4	Dimensions in metric units and inches: G71, G70, G710, G700 .....	217
9.2.5	Polar coordinates, pole definition: G110, G111, G112 .....	218
9.2.6	Programmable work offset: TRANS, ATRANS .....	220
9.2.7	Programmable rotation: ROT, AROT .....	221
9.2.8	Programmable scaling factor: SCALE, ASCALE .....	222
9.2.9	Programmable mirroring: MIRROR, AMIRROR .....	224
9.2.10	Workpiece clamping - settable work offset: G54 to G59, G500, G53, G153 .....	225
9.2.11	Programmable working area limitation: G25, G26, WALIMON, WALIMOF .....	227
9.3	Axis movements .....	229
9.3.1	Linear interpolation with rapid traverse: G0 .....	229
9.3.2	Linear interpolation with feedrate: G1 .....	230
9.3.3	Circular interpolation: G2, G3 .....	232
9.3.4	Circular interpolation via intermediate point: CIP .....	238
9.3.5	Circle with tangential transition: CT .....	239
9.3.6	Helix interpolation: G2/G3, TURN .....	240
9.3.7	Thread cutting with constant lead: G33 .....	241
9.3.8	Tapping with compensating chuck: G63 .....	242
9.3.9	Thread interpolation: G331, G332 .....	244
9.3.10	Fixed point approach: G75 .....	246
9.3.11	Reference point approach: G74 .....	247
9.3.12	Measuring with touch-trigger probe: MEAS, MEAW .....	248
9.3.13	Tangential control: TANG, TANGON, TANGOF, TLIFT, TANGDEL .....	249
9.3.14	Feedrate F .....	252
9.3.15	Feedrate override for circles: CFTCP, CFC .....	253
9.3.16	Exact stop / continuous-path control mode: G9, G60, G64 .....	255
9.3.17	Acceleration pattern: BRISK, SOFT .....	258
9.3.18	Percentage acceleration override: ACC .....	259
9.3.19	Traversing with feedforward control: FFWON, FFWOF .....	260
9.3.20	Enhanced surface quality through compressor: COMPCAD .....	261
9.3.21	Fourth axis .....	262
9.3.22	Dwell time: G4 .....	263
9.3.23	Travel to fixed stop .....	264
9.3.24	Feed reduction with corner deceleration (FENDNORM, G62, G621) .....	267
9.3.25	Coupled axes .....	268
9.3.25.1	Coupled motion (TRAILON, TRAILOF) .....	268
9.3.25.2	Coupled axis motion (TRAILON, TRAILOF): Further Information .....	271
9.3.25.3	Master/slave group (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) .....	272
9.3.25.4	Master/slave group (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS): Further Information .....	274
9.4	Spindle movements .....	275
9.4.1	Spindle speed S, directions of rotation .....	275
9.4.2	Spindle speed limitation: G25, G26 .....	276
9.4.3	Spindle positioning: SPOS .....	277
9.4.4	Gear stages .....	278
9.5	Contour programming support .....	279
9.5.1	Rounding, chamfer .....	279
9.5.2	Contour definition programming .....	282
9.6	Tool and tool offset .....	284
9.6.1	General Information .....	284
9.6.2	Tool T .....	285
9.6.3	Tool compensation number D .....	286

9.6.4	Selecting the tool radius compensation: G41, G42 .....	289
9.6.5	Corner behavior: G450, G451 .....	291
9.6.6	Tool radius compensation OFF: G40 .....	293
9.6.7	Special cases of the tool radius compensation .....	295
9.6.8	Example of tool radius compensation .....	296
9.7	Miscellaneous function M .....	298
9.8	H function .....	300
9.9	Arithmetic parameters, LUD and PLC variables .....	301
9.9.1	Arithmetic parameter R .....	301
9.9.2	Local User Data (LUD) .....	303
9.9.3	Reading and writing PLC variables .....	305
9.10	Program jumps .....	306
9.10.1	Jump destination for program jumps .....	306
9.10.2	Unconditional program jumps .....	307
9.10.3	Conditional program jumps .....	308
9.10.4	Program example for jumps .....	310
9.11	Subroutine technique .....	312
9.11.1	General information .....	312
9.11.2	Calling machining cycles .....	315
9.11.3	Modal subroutine call .....	315
9.11.4	Execute external subroutine (EXTCALL) .....	316
9.12	Timers and workpiece counters .....	319
9.12.1	Runtime timer .....	319
9.12.2	Workpiece counter .....	321
9.13	Language commands for tool monitoring .....	323
9.13.1	Tool monitoring overview .....	323
9.13.2	Tool life monitoring .....	326
9.13.3	Workpiece count monitoring .....	328
9.14	Smooth approach and retraction .....	333
9.15	Milling of the peripheral surface - TRACYL .....	340
<b>10</b>	<b>Cycles .....</b>	<b>347</b>
10.1	Overview of cycles .....	347
10.2	Programming cycles .....	349
10.3	Graphical cycle support in the program editor .....	352
10.4	Drilling cycles .....	354
10.4.1	General information .....	354
10.4.2	Requirements .....	355
10.4.3	Drilling, centering - CYCLE81 .....	357
10.4.4	Drilling, counterboring - CYCLE82: .....	360
10.4.5	Deep-hole drilling - CYCLE83 .....	363
10.4.6	Rigid tapping - CYCLE84 .....	368
10.4.7	Tapping with compensating chuck - CYCLE840 .....	372
10.4.8	Reaming 1 (Drilling 1) – CYCLE85 .....	379
10.4.9	Boring (Drilling 2) – CYCLE86 .....	382
10.4.10	Boring with stop 1 (boring 3) – CYCLE87 .....	386
10.4.11	Drilling with stop 2 (Drilling 4) - CYCLE88 .....	389
10.4.12	Reaming 2 (Drilling 5) – CYCLE89 .....	391

10.5	Drilling pattern cycles.....	394
10.5.1	Requirements.....	394
10.5.2	Row of holes - HOLES1.....	395
10.5.3	Circle of holes - HOLES2.....	399
10.6	Milling cycles.....	403
10.6.1	Requirements.....	403
10.6.2	Face milling - CYCLE71.....	404
10.6.3	Contour milling - CYCLE72.....	410
10.6.4	Rectangular spigot milling - CYCLE76.....	422
10.6.5	Circular spigot milling - CYCLE77.....	428
10.6.6	Long holes located on a circle - LONGHOLE.....	433
10.6.7	Slots on a circle - SLOT1.....	439
10.6.8	Circumferential slot - SLOT2.....	446
10.6.9	Milling a rectangular pocket - POCKET3.....	451
10.6.10	Milling a circular pocket - POCKET4.....	460
10.6.11	Thread milling - CYCLE90.....	465
10.7	Error messages and error handling.....	471
10.7.1	General Information.....	471
10.7.2	Error handling in the cycles.....	471
10.7.3	Overview of cycle alarms.....	472
10.7.4	Messages in the cycles.....	474
<b>11</b>	<b>Network operation.....</b>	<b>475</b>
11.1	Network operation prerequisites.....	475
11.2	RCS802 tool.....	476
11.3	Network operation.....	481
11.3.1	Network operation.....	481
11.3.2	Configuring the network connection.....	482
11.3.3	User management.....	484
11.3.4	User log in - RCS log in.....	485
11.3.5	Working on the basis of a network connection.....	486
11.3.6	Sharing directories.....	487
11.3.7	Connecting / disconnecting network drives.....	488
<b>12</b>	<b>Data backup.....</b>	<b>491</b>
12.1	Data transfer via RS232 interface.....	491
12.2	Creating / reading in / reading out a start-up archive.....	493
12.3	Reading in / reading out PLC projects.....	496
12.4	Copying and pasting files.....	497
<b>13</b>	<b>PLC diagnostics.....</b>	<b>499</b>
13.1	PLC diagnosis represented as a ladder diagram.....	499
13.2	Screen layout.....	500
13.3	Operating options.....	501
<b>A</b>	<b>Appendix.....</b>	<b>513</b>
A.1	Miscellaneous.....	513
A.1.1	Pocket calculator.....	513
A.1.2	Editing Asian characters.....	515
A.2	Feedback on the documentation.....	519

A.3	Overview of documentation.....	521
	<b>Index.....</b>	<b>523</b>



# Description

## 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:

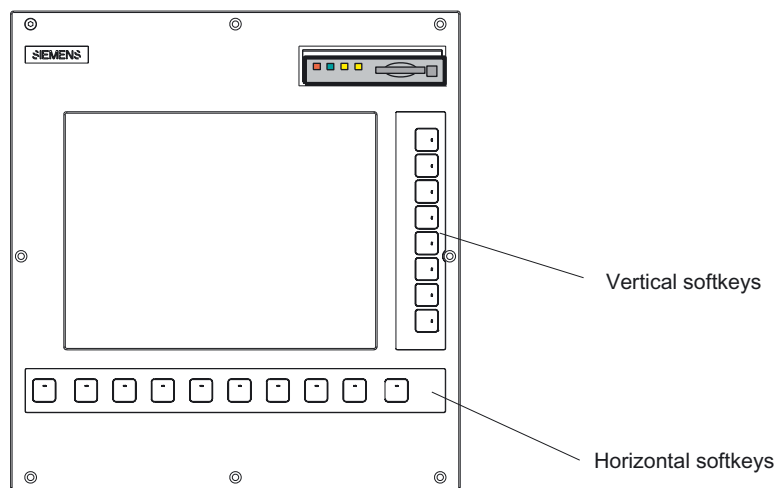
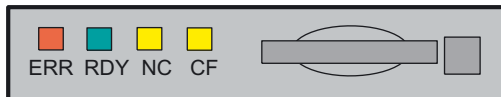


Figure 1-1 CNC operator panel

## 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.

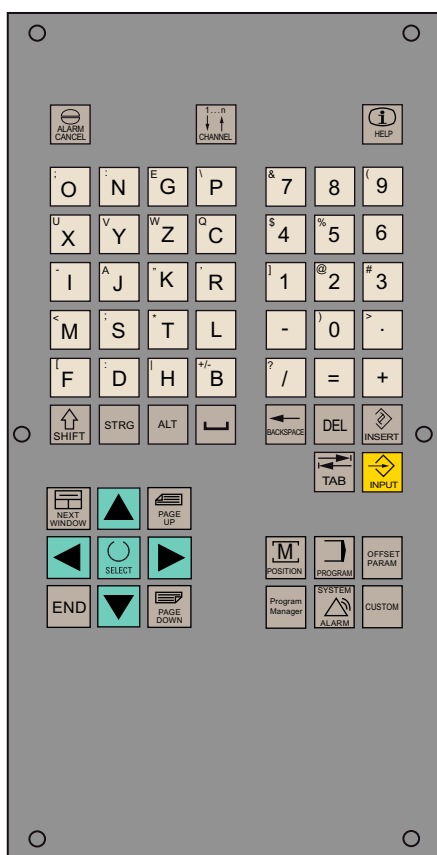
Table 1- 1 Status and error displays











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)





	Delete key
	Insert key
	Tabulator
	ENTER / Input key
	POSITION operating area key (Position operating area)
	PROGRAM operating area key (Program operating area)
	OFFSET PARAM operating area key (Parameter operating area)
	PROGRAM MANAGER operating area key (Program Manager operating area)
	SYSTEM/ALARM operating area key (System/Alarm operating area)
	CUSTOM operating area (User operating area)


	ETC key		Recall key
---	---------	---	------------






	Acknowledge alarm key
---	-----------------------

	No function
---	-------------

	Info key
---	----------



	END
	Selection key / toggle key

	Shift key
---	-----------


		Scroll keys
		Cursor keys
		



	Control key
---	-------------

	Space
---	-------

		Alphanumeric keys Double assignment at the Shift level
---	--	---

	ALT key
---	---------

	Delete key (backspace)
---	------------------------

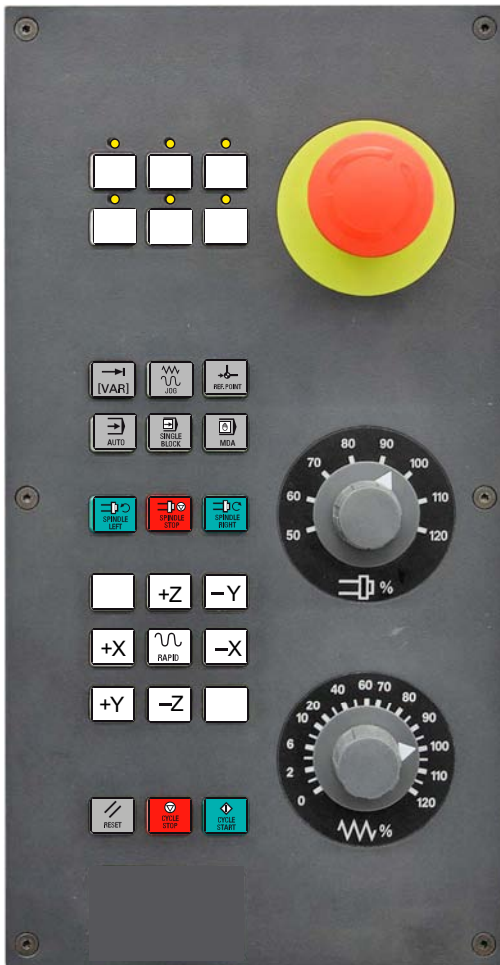
		Numeric keys Double assignment at the Shift level
---	--	--





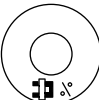
## Hot keys

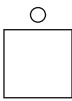

















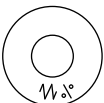
In the part program editor and in the input fields of the HMI, the following functions can be carried out with certain key combinations on the full CNC keyboard:

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
	CYCLE STOP (NC STOP)
	CYCLE START (NC START)
	EMERGENCY STOP
	Spindle Speed Override Spindle override

	User-defined key with LED
	User-defined key without LED
	INCREMENT Increment
	JOG
	REFERENCE POINT Reference point
	AUTOMATIC
	SINGLE BLOCK Single block
	MANUAL DATA Manual input
	SPINDLE START CCW Counterclockwise
	SPINDLE STOP
	SPINDLE START CW Clockwise
	RAPID TRAVERSE OVERLAY Rapid traverse override
	X axis
	
	Y axis
	
	Z axis
	
	Feedrate override Feedrate control

*Description*

---

*1.4 Key definition of the machine control panel*

---

**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.

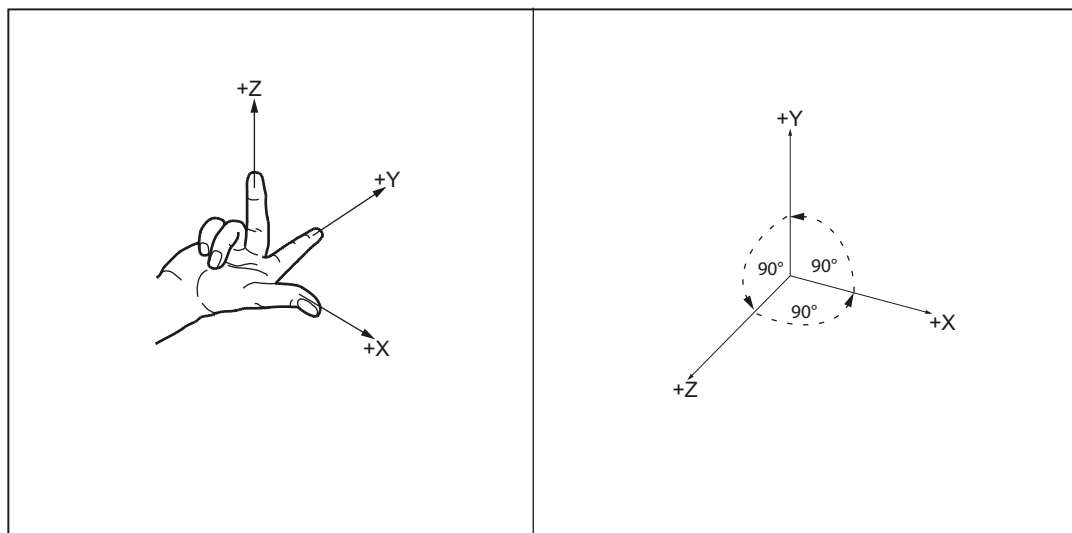


Figure 1-2 Determination of the axis directions to one another; coordinate system for programming

### 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.

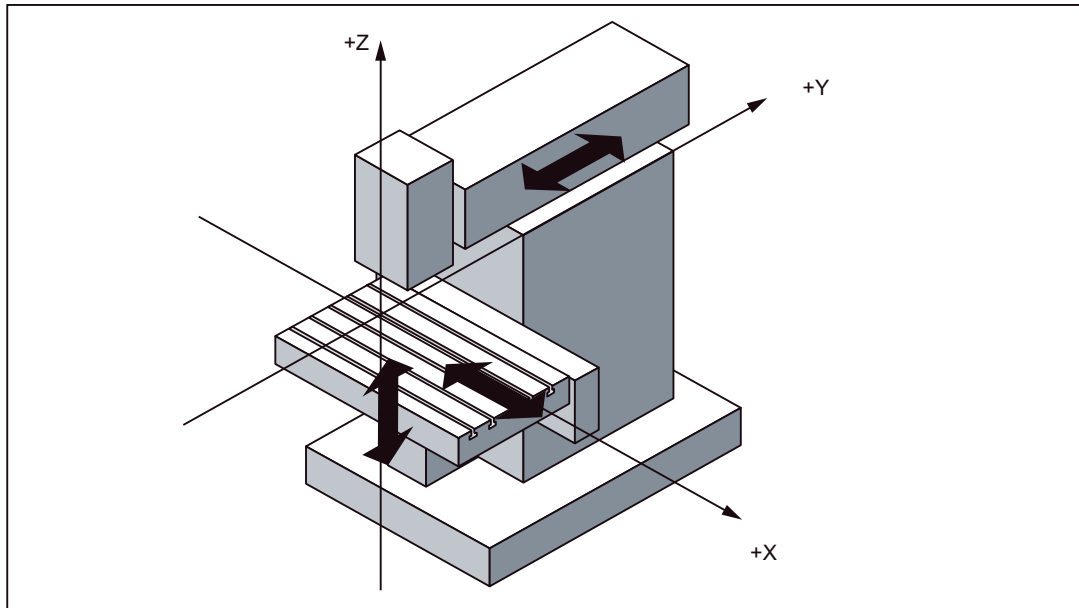


Figure 1-3 Machine coordinates/axes using the example of a milling machine

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 be 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.

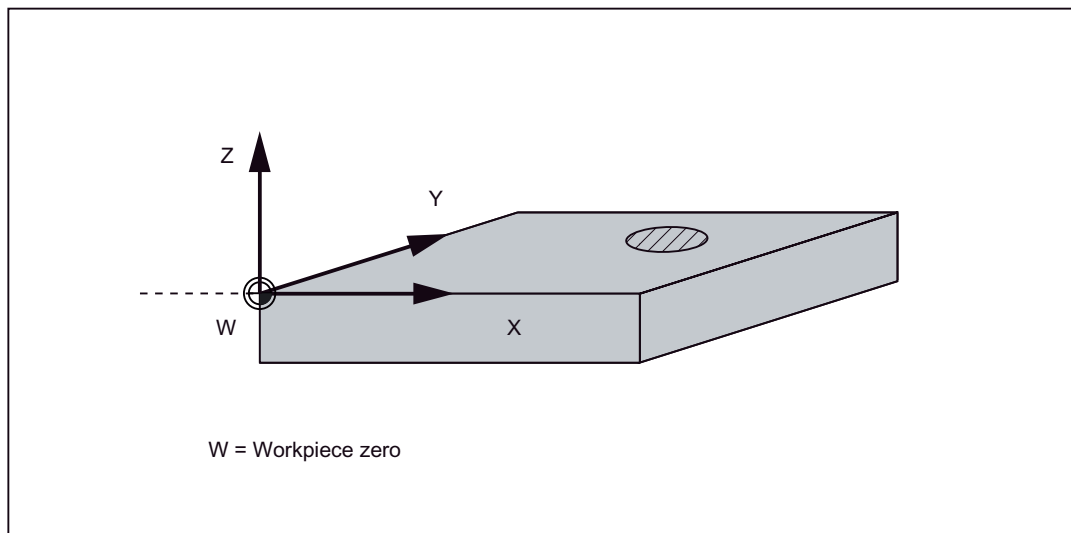


Figure 1-4 Workpiece Coordinate System

### 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

The actual value in the associated coordinate system can be activated and displayed in the "Position" operating area using the "MKS/WKS REL" vertical softkey.

---

### 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**.

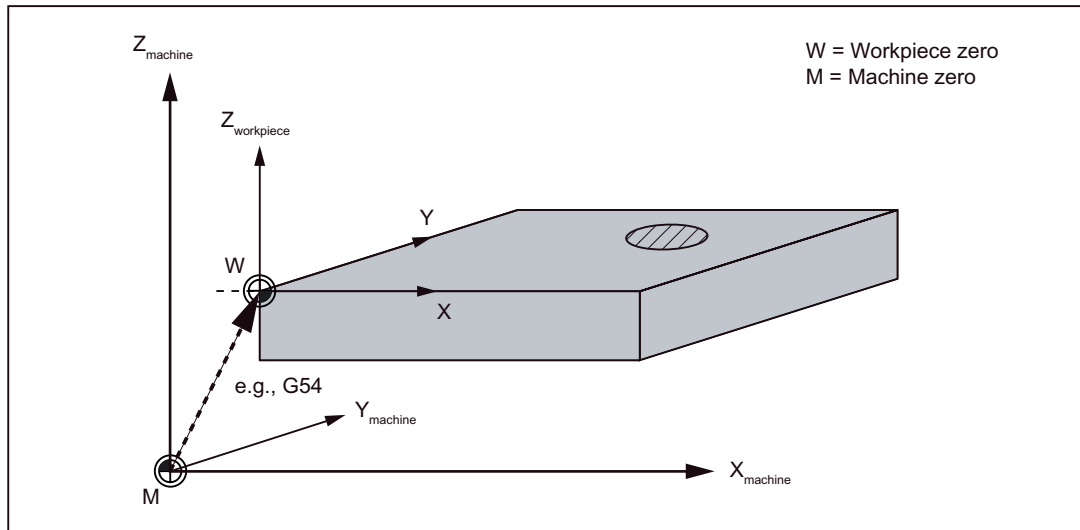


Figure 1-5 Workpiece on the machine

### 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 (see Section "Programmable work offset: TRANS").

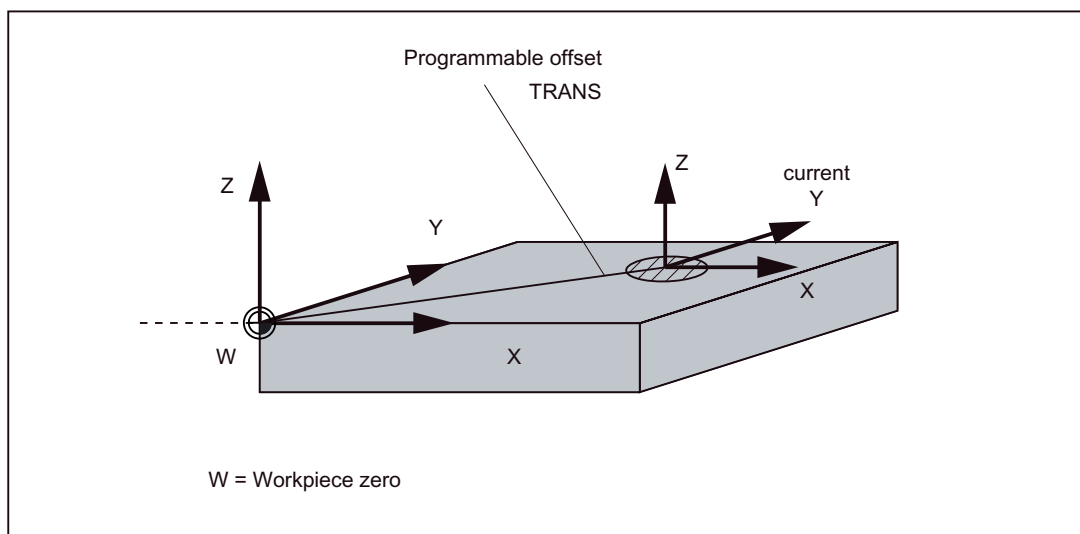


Figure 1-6 Coordinates on the workpiece; current workpiece coordinate system

## Software interface

### 2.1 Screen layout

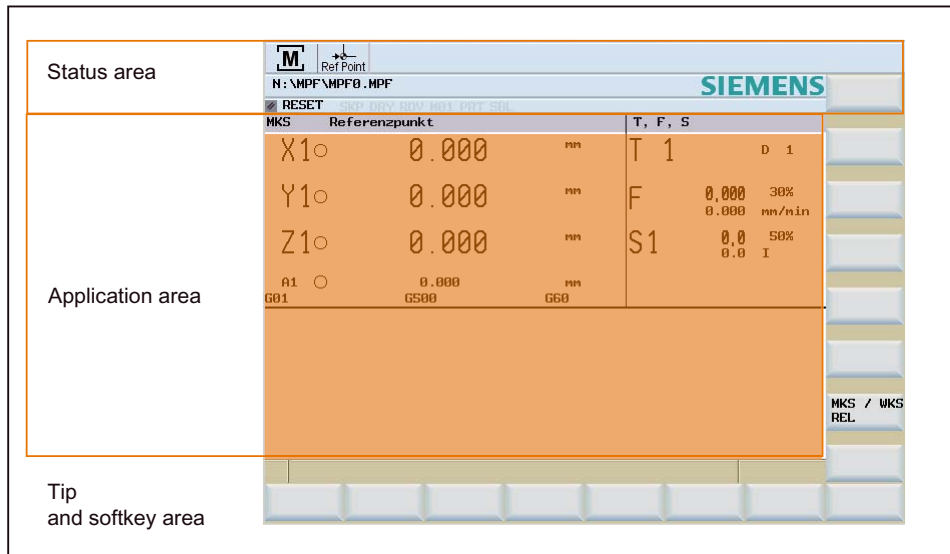


Figure 2-1 Screen layout

The screen is divided into the following main areas:

- Status area
- Application area
- Note and softkey area

#### Status area

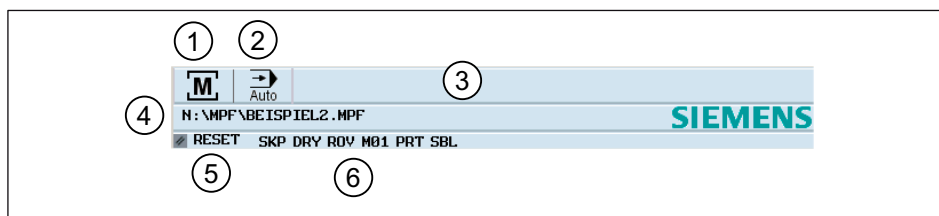













Figure 2-2 Status area

2.1 Screen layout

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

Numbering	Display	Icon	Significance
①	Active operating area		Position (operating area key <POSITION>)
			System (operating area key <SYSTEM>)
			Program (operating area key <PROGRAM>)
			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
			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

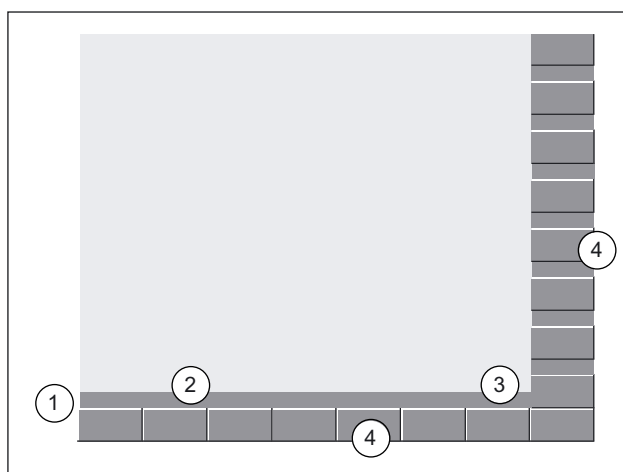








Figure 2-3 Note and softkey area

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

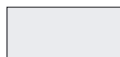
Screen item	Display	Significance
③		HMI status information
		ETC is possible (pressing this key displays the horizontal softkey bar providing further functions.)
		Mixed notation active (uppercase/lowercase letters)
		RS232 connection active
		Connection to commissioning and diagnostic tools (e.g. Programming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

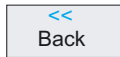


Horizontal softkey



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:

	POSITION	Machine operation
	OFFSET PARAM	Entering the compensation values and setting data
	PROGRAM	Creation of part programs
	PROGRAM MANAGER	Part program directory
	SYSTEM	Diagnostics, commissioning
	ALARM	Alarm and message lists
	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 rights.

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



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

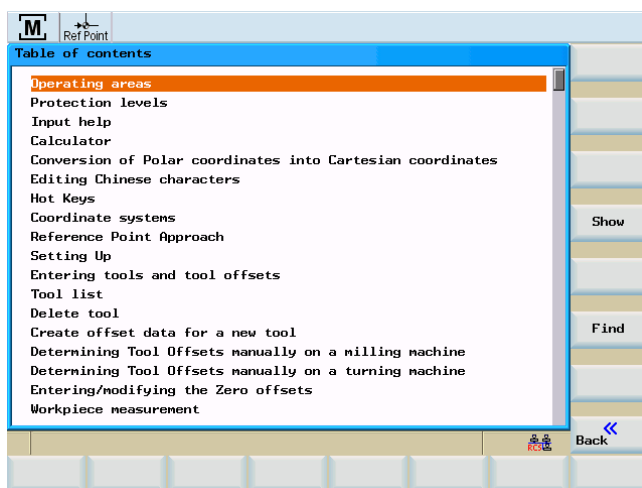


Figure 2-4 Help system: Table of contents

## Softkeys

Show

This function opens the selected topic.

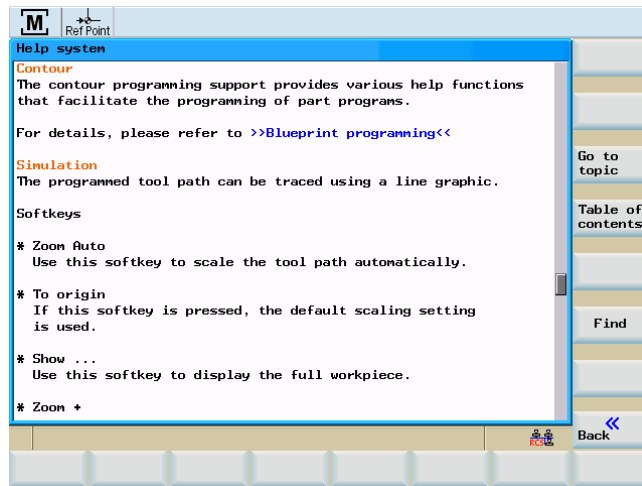


Figure 2-5 Help system: Description of the topic

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

### 3.1 Turning on and approaching reference points

#### Note

When you turn 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.

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



The "Reference point" window is active.

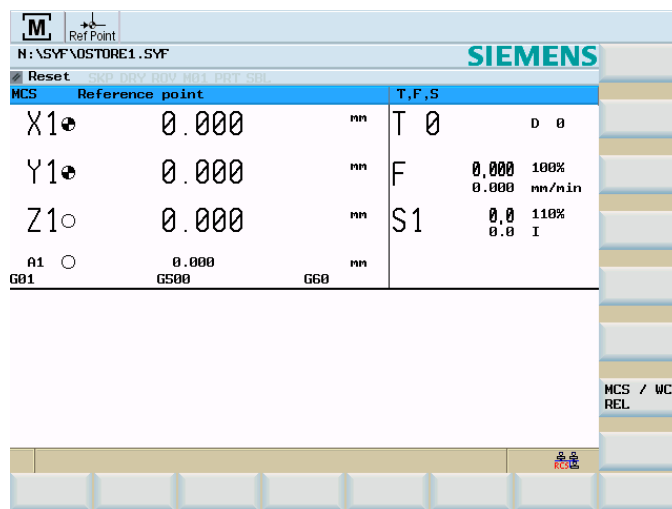


Figure 3-1 Reference-point approach start screen

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

- Axis must be referenced
- Axis is referenced/synchronized

3.1 Turning on and approaching reference points

---



Press the arrow keys.



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).



To access the functions described below, you need to select "Jog" mode.

## 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

### Functionality

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

See also Chapter "Tool and tool offset (Page 284)"

### Operating sequences



Press the <OFFSET PARAM> key.



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).



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

### Standard tool list

Type	T	D <sub>z</sub>	Geometry		Active tool no
			Length1	Radius	
1	1		0.000	0.000	8
2	1		0.000	0.000	8
3	1		0.000	0.000	5
4	1		0.000	0.000	8
5	1		0.000	0.000	8
6	1		0.000	0.000	8
7	1		0.000	0.000	8
8	1		0.000	0.000	8
9	1		0.000	0.000	8
10	1		0.000	0.000	8
11	1		0.000	0.000	8
12	1		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	Contents
Type	Cutting edge type of the tool and tool monitoring symbols (refer to Chapter "Tool monitoring")
T	Tool number
D <sub>Σ</sub>	Number of tool cutting edges
Geometry	Tool geometry
	Cutting edge position of the cutting edge

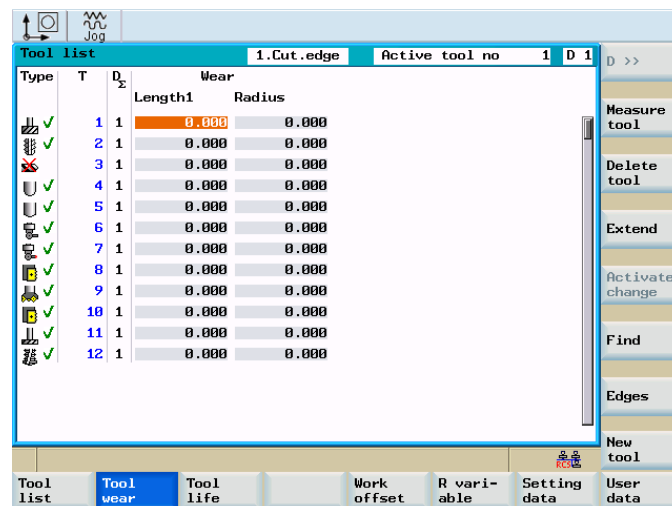
The following is displayed in the "tool list" line:

- 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. 1, 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.















Type	T	D <sub>Σ</sub>	Wear	
			Length1	Radius
	1	1	0.000	0.000
	2	1	0.000	0.000
	3	1	0.000	0.000
	4	1	0.000	0.000
	5	1	0.000	0.000
	6	1	0.000	0.000
	7	1	0.000	0.000
	8	1	0.000	0.000
	9	1	0.000	0.000
	10	1	0.000	0.000
	11	1	0.000	0.000
	12	1	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).

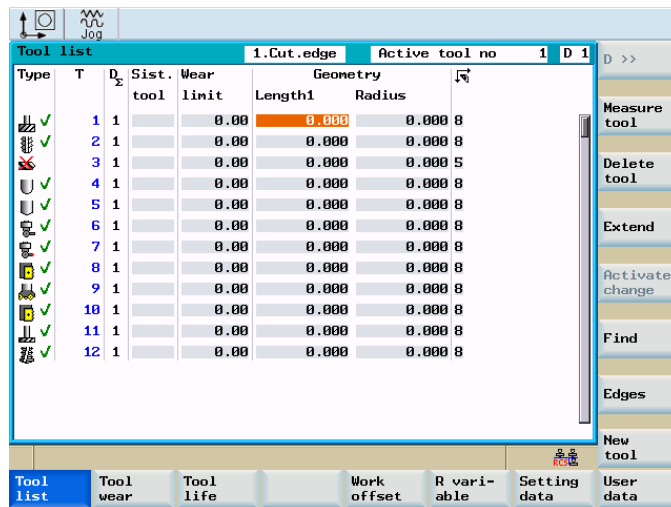


Figure 4-3 User-defined tool list

If you activate the following display machine data with "1", the tool list will be augmented with the parameters for the "position number" and the "H number", and the "oversize" checkbox:

- TOOL\_LIST\_PLACE\_NO MD332 display
- DISPLAY\_TOOL\_H\_NO MD393 display
- COL\_OVERSIZE\_TYPE\_CHECKBOX MD395 display

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 sensing 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.

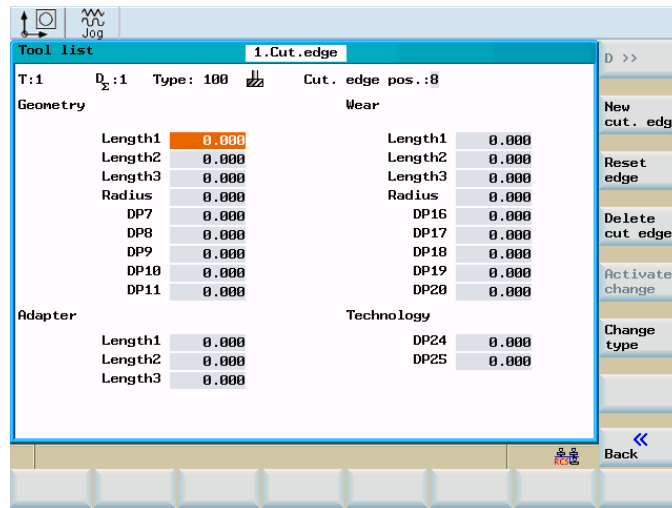


Figure 4-4 Input screen for special tools

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 compensation values of the edge to zero.

Delete cutting edge

Cutting edge is deleted.

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 compensation data for a new tool.

### 4.1.1 Create new tool

#### Operating sequence

New tool

This function offers another two softkey functions to select the "drill" and "milling cutter" tool type. After selecting the tool type, enter the desired "Tool number" (3 digits max.) in the input field and select the "Type".

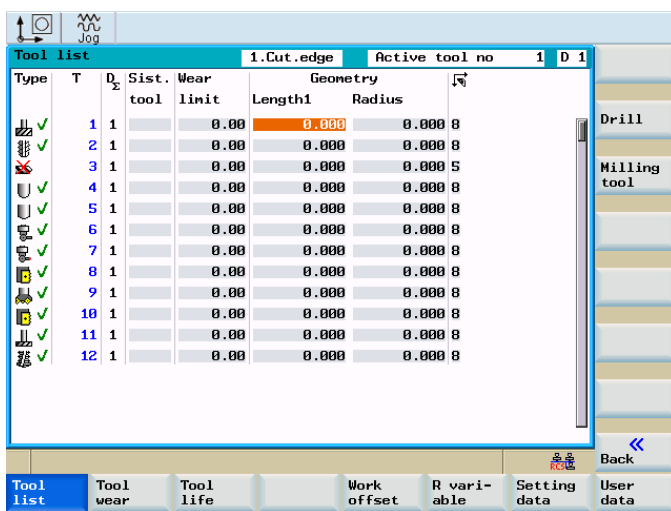


Figure 4-5 "New tool" window

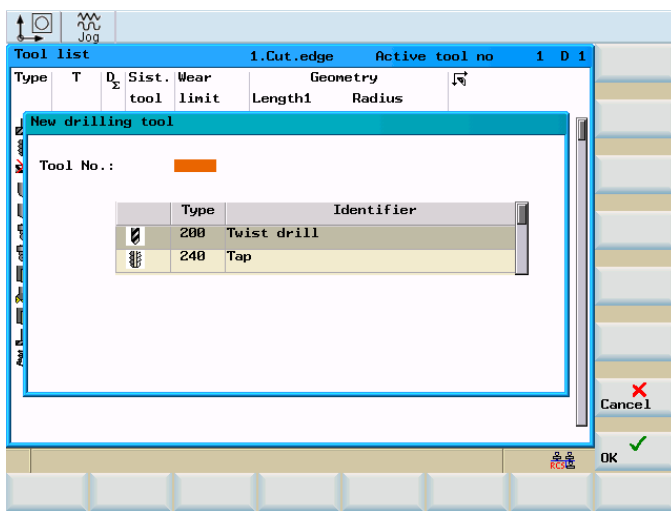


Figure 4-6 Input for tool number and type selection for a drill

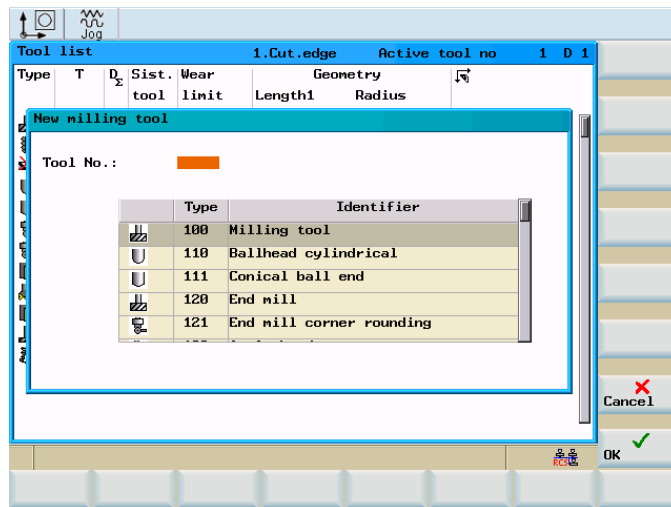


Figure 4-7 Input for tool number and type selection for a milling cutter



Confirm your input using <OK>. A data record loaded with zero will be included in the tool list.

### 4.1.2 Determining the tool offsets (manually)

**Note**

For milling tools, length 1 and the radius, and for drilling tools (see following figure), only length 1 must be determined.

**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.

By using the actual position of point F (machine coordinate) and the reference point, the control system can calculate the offset value assigned to length 1 or the tool radius for the selected axis.

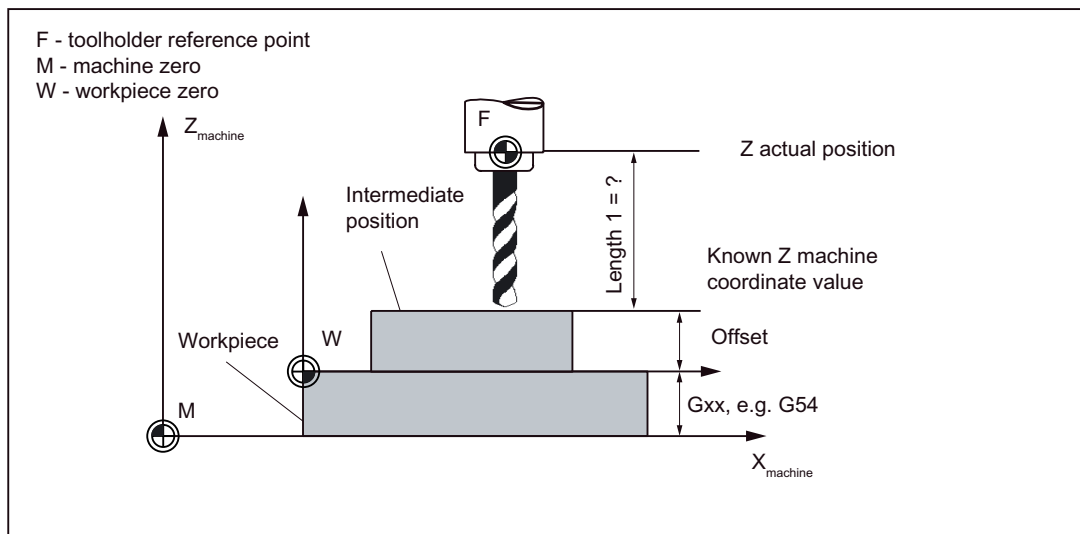


Figure 4-8 Determination of the length offset using the example of a drill 1/Z axis length

### 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:

- 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

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

## Operating sequence

Tool  
measurement

Use this softkey to open the list box for manual and semiautomatic measuring.

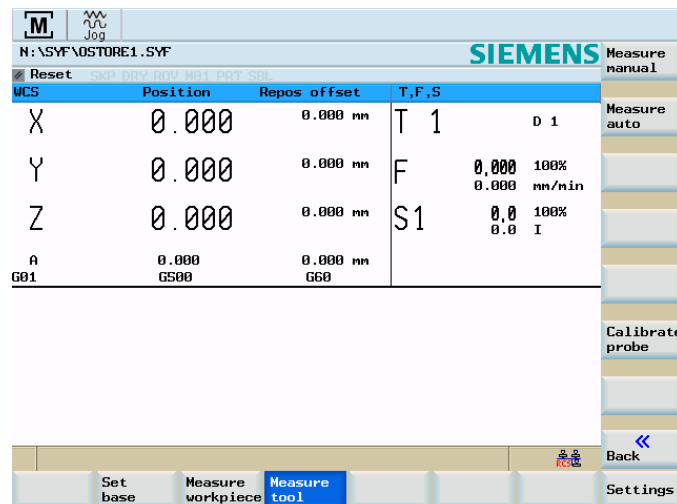


Figure 4-9 Selection "Manual or semiautomatic measuring"

Measuring Manual

The "Tool measurement manual" window is opened with the default setting "Measure length".

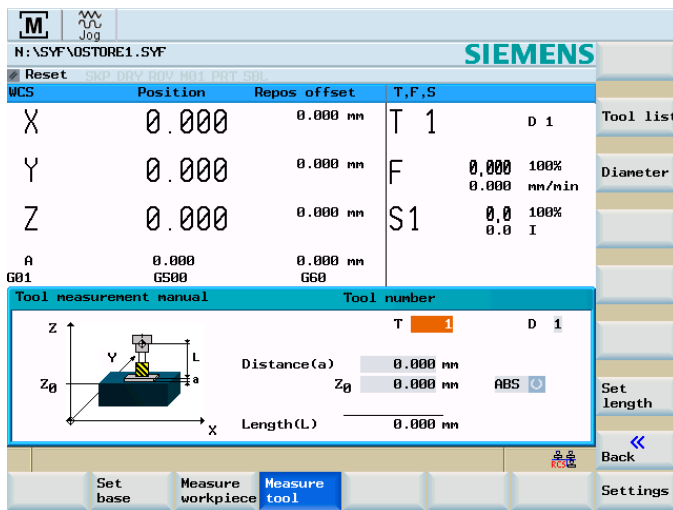


Figure 4-10 "Measure tool manually" window, measure length

### Workpiece parameters and operating sequence to manually measure the tool "Length"

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 "Z<sub>0</sub>", if "ABS" was pre-selected in the adjacent toggle field.
- Press "Set length".

Set length

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

#### Note

You can also use a zero already determined (e.g value of G54). This should be selected in the toggle field for the reference point.

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



## Workpiece parameters and operating sequence to manually measure the tool "Diameter"

Diameter

To measure the diameter "Diameter".

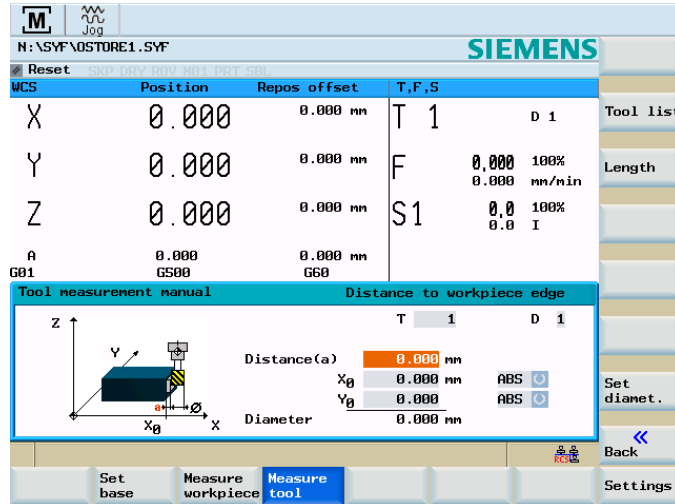


Figure 4-11 "Measure tool manually" window, measure tool diameter

Enter the following workpiece parameters to calculate the diameter 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 fields "X<sub>0</sub>" and "Y<sub>0</sub>", if "ABS" was pre-selected in the adjacent toggle field.

Set diameter

- Press "Set diameter".

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

### 4.1.3 Determining tool offsets using a probe (auto)

#### Operating sequence

Tool measurement

Press the "Measure tool" softkey.

Measuring Auto

The "Measure tool auto" window opens up.

#### Measuring the tool length

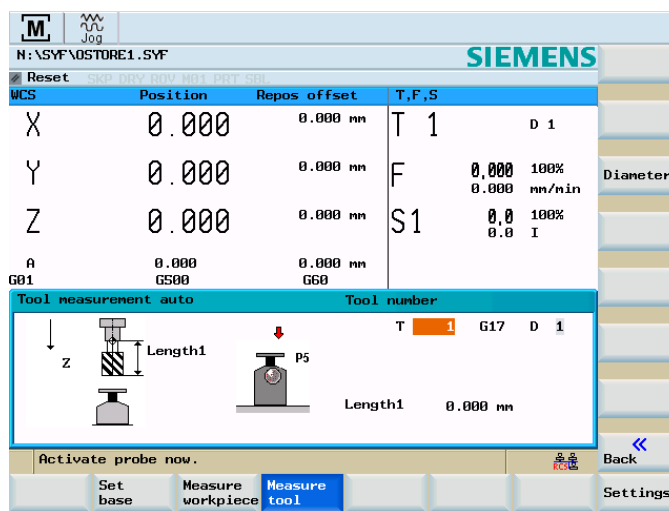


Figure 4-12 "Measure tool auto" window, measure length

#### "Tool measurement auto" screen form

After the screen form has appeared, the input fields are loaded with the tool currently working, and the plane, in which the measurements are to be performed, are displayed.

#### Note

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

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

### Measuring the tool length

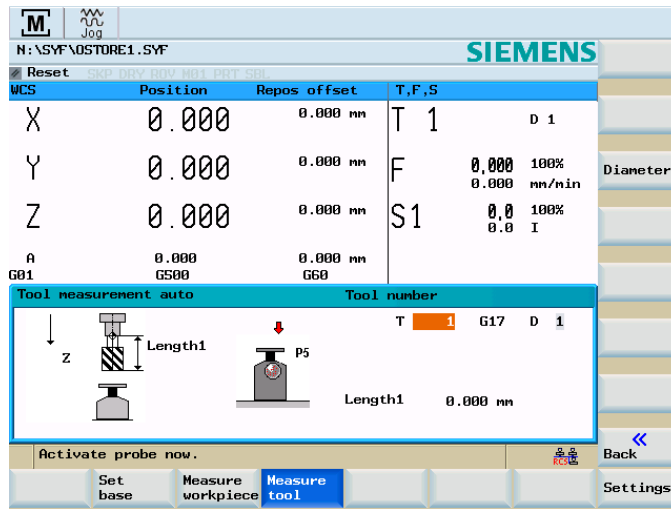


Figure 4-13 "Measure tool auto" window, measure length

Use the feed axis to traverse to the probe.



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.

### Measuring the tool diameter

The diameter can only be determined with the spindle rotating. To this end, enter the spindle speed and the direction of rotation of the spindle in the "Probe data" screen form.

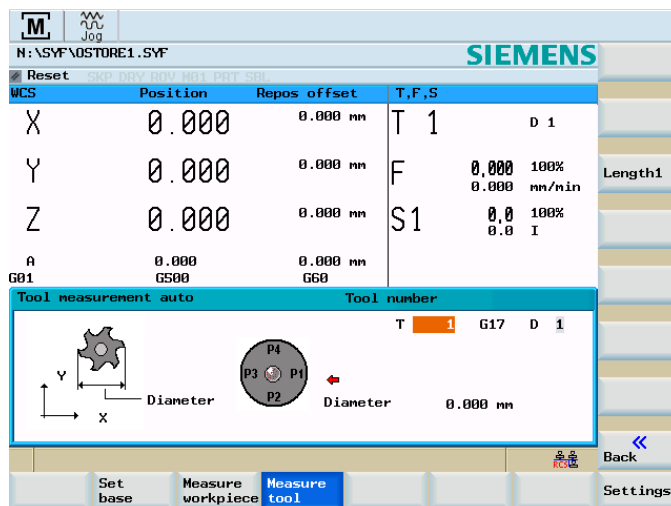


Figure 4-14 "Measure tool auto" window, measure diameter

Use any axis from the plane to traverse to the probe. Depending on the axis selected, traverse either to point P1 or P3, or P2 or P4.



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.

<b>WARNING</b>
----------------

The spindle is run at the speed stored in the probe data.
---

### Action for "Probe tripped"



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

After activating the probe, the axis direction key must be released.

After releasing the axis direction key, the control system automatically creates and starts an internal measuring program in the program memory.

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

If after the third approach of the probe no measured value is transferred to the control system, a message will appear in the display to inform the operator that no measured values could be recorded.

All axes involved in the measurement process must be approached in this manner.

### 4.1.4 Probe settings

Settings

Select "Settings" softkey.

Probe data

The screen form below is used to store the coordinates of the probe and to set the following parameters for the automatic measuring process:

- Probe level
- Axial feedrate
- Speed and direction of rotation of the spindle

The direction of rotation of the spindle must be opposite to the cutting direction of the milling cutter.

All position values refer to the machine coordinate system.

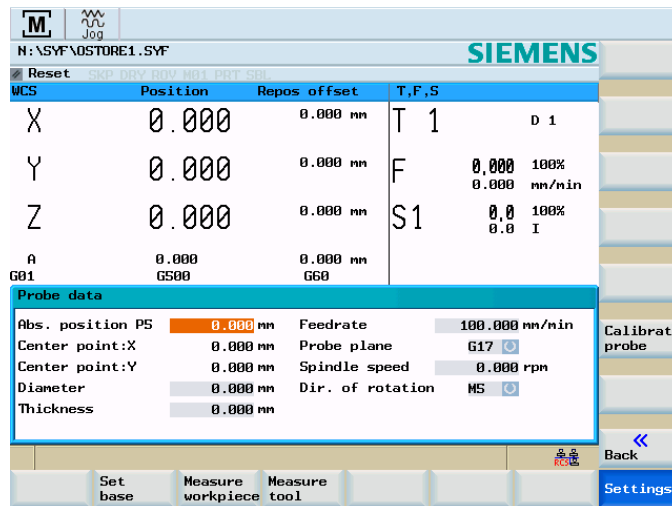


Figure 4-15 "Probe data" input screen

Table 4- 2 Meaning of the input fields

Parameter	Significance
Abs. position P5	Absolute position of the probe in the Z-direction
Center point: X Center point: Y	Calculated center point of the probe (machine coordinates)
Diameter	Diameter of the probe disk (after calibration, the calculated diameter is displayed)
Thickness	Thickness of the probe disk

Probe calibration

Comp. Mes.  
contact  
switch

The probe can be calibrated either in the "Settings" menu or in the "Measure tool" menu.

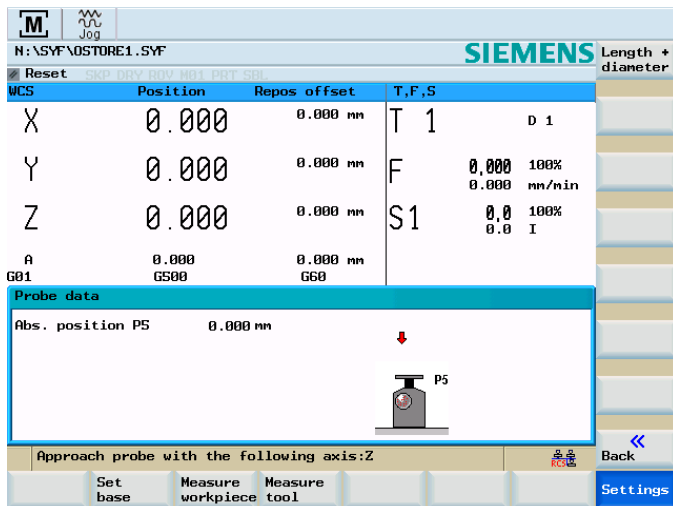


Figure 4-16 Calibrating the probe (length)...(diameter)

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. If the probe is triggered, the control system will take over the measuring process by switching to AUTOMATIC mode, activating the measuring program and starting it automatically. The operator will see an axis movement in the opposite direction for a short time.

During the measurement, the active state of the NC is symbolized by a dial gauge.

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

Note

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



## 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.

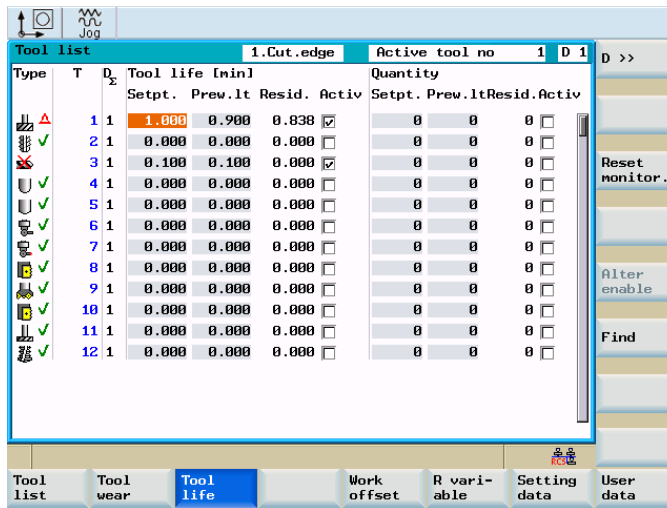


Figure 4-17 Tool monitoring

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.

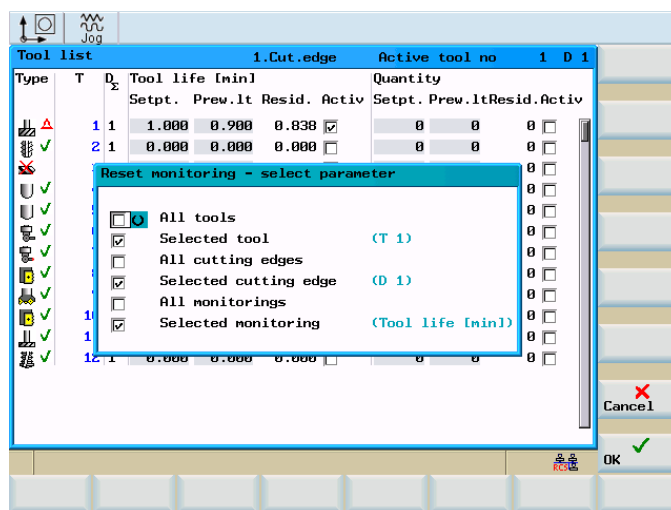


Figure 4-18 Reset monitoring

## 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. 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

Select work offset using "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.

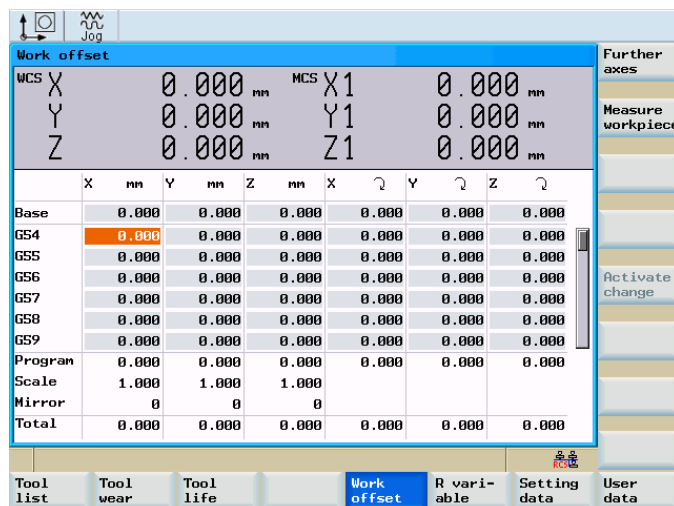


Figure 4-19 The "Work offset" window

- Position the cursor bar on the input field to be changed and
- 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 select the window with the relevant work offset (e.g. G54) and the axis you want to determine for the offset.

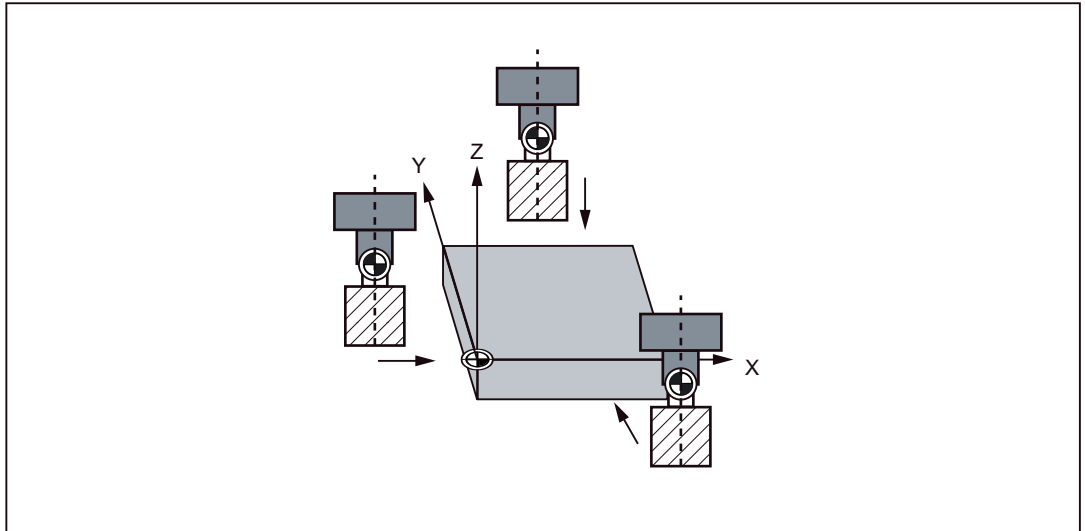


Figure 4-20 Determining the work offset

Procedure

Measure workpiece

Press the "Measure tool" softkey. The control system will switch to the <POSITION> operating area and 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.

If scratching is not possible or if you cannot reach the appropriate point with the tool (e.g. when using a spacer), enter the clearance between the tool and the workpiece surface in the "Clearance" field.

To determine the offset, the direction of movement of the tool must be taken into account for the active tool. If no tool is active, the "Radius" field is hidden.

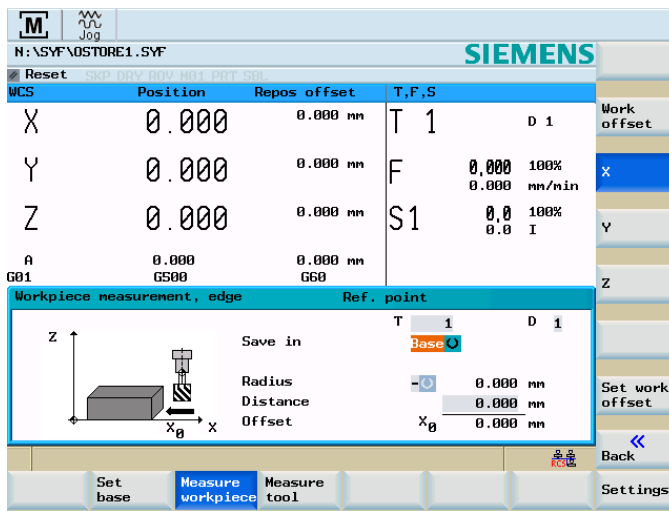


Figure 4-21 Determine work offset in X screen form

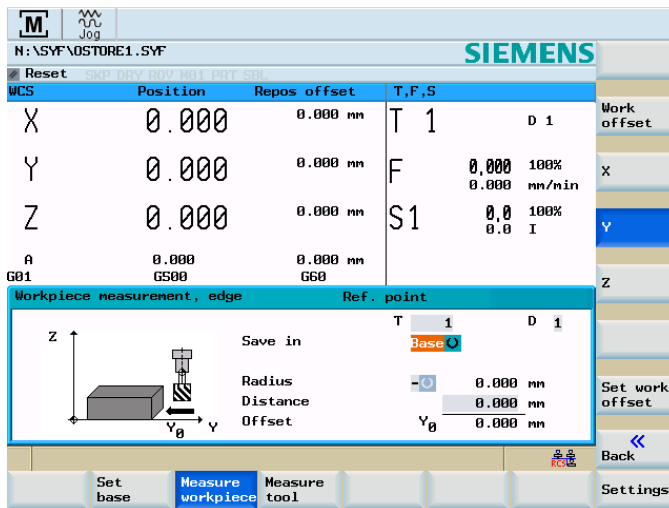


Figure 4-22 Determine work offset in Y screen form

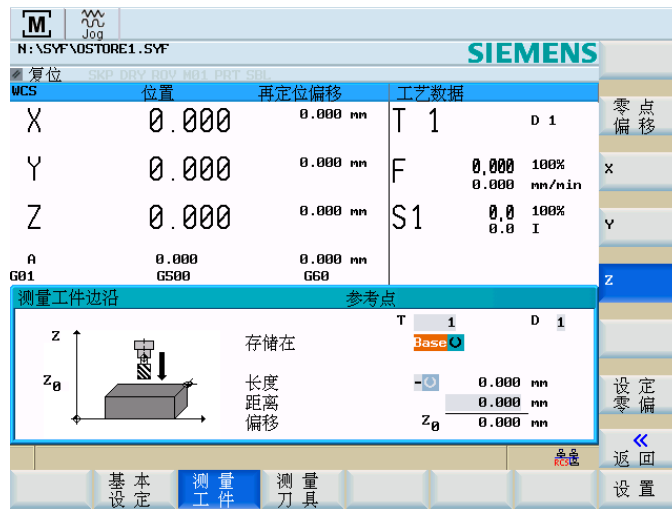


Figure 4-23 Determine work offset in Z screen form

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.

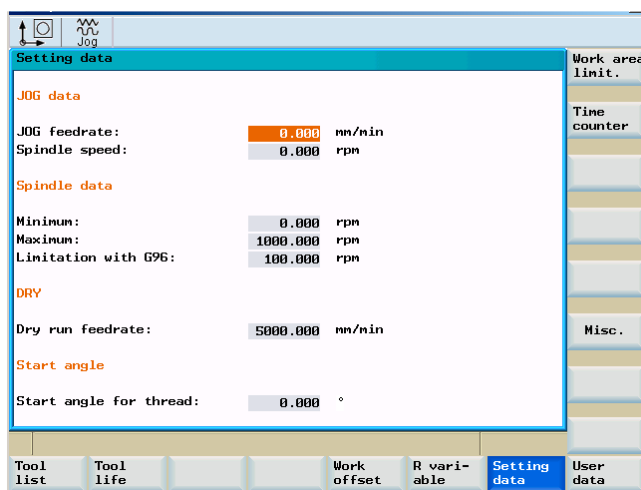


Figure 4-24 Setting data start screen

- **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)**

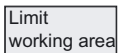
For thread cutting, a start position for the spindle is displayed as the start angle. A multiple thread can be cut by changing the angle when the thread cutting operation is repeated.

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



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

## Softkeys



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.

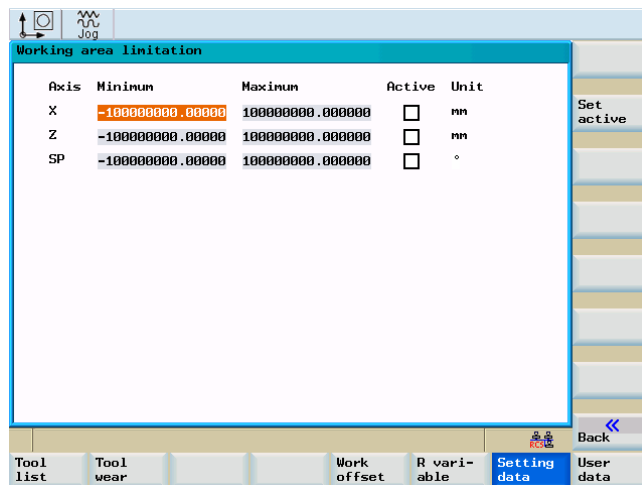


Figure 4-25 Working area limitation

Times  
Multiplier

Times Counters

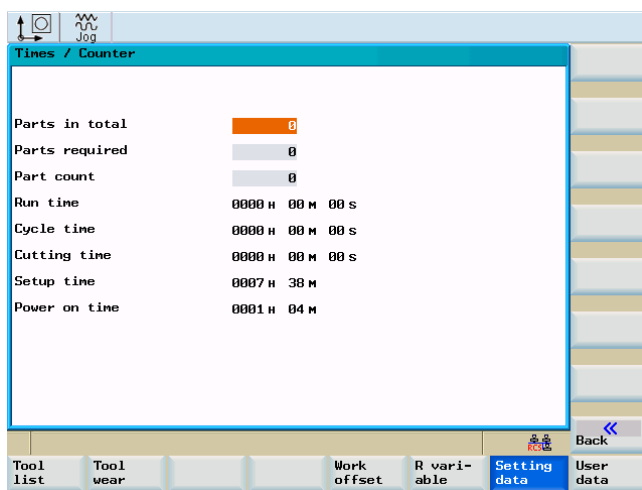


Figure 4-26 Times, Counters

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."

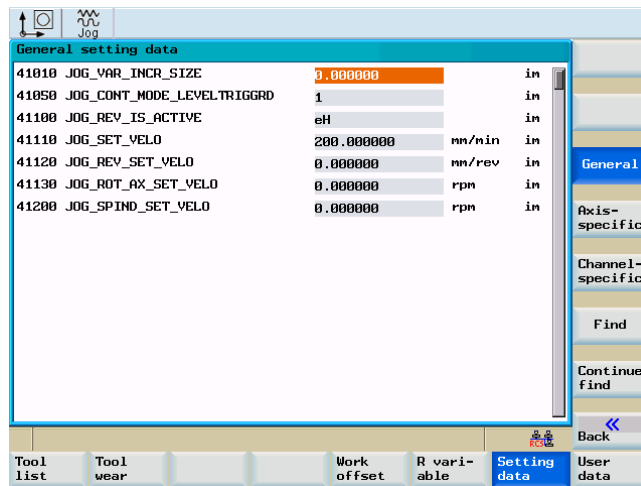


Figure 4-27 General setting data

## 4.5 Arithmetic parameter R

### 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.

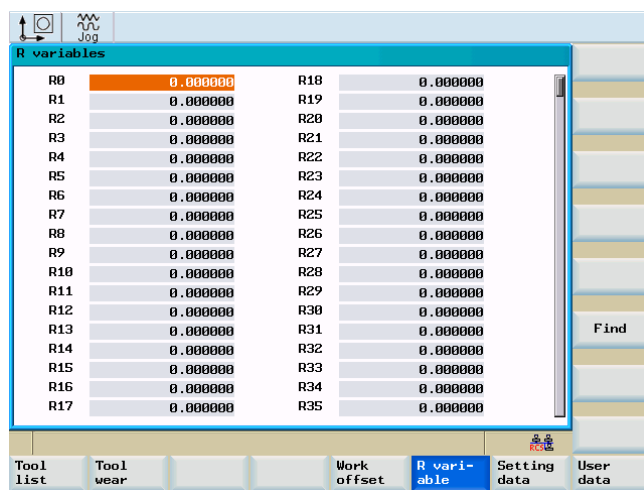


Figure 4-28 "R parameters" start screen

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

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

INPUT

Find

Searching for R variables

## Manually Controlled Mode

### 5.1 Manually Controlled Mode

#### Preliminary remark

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

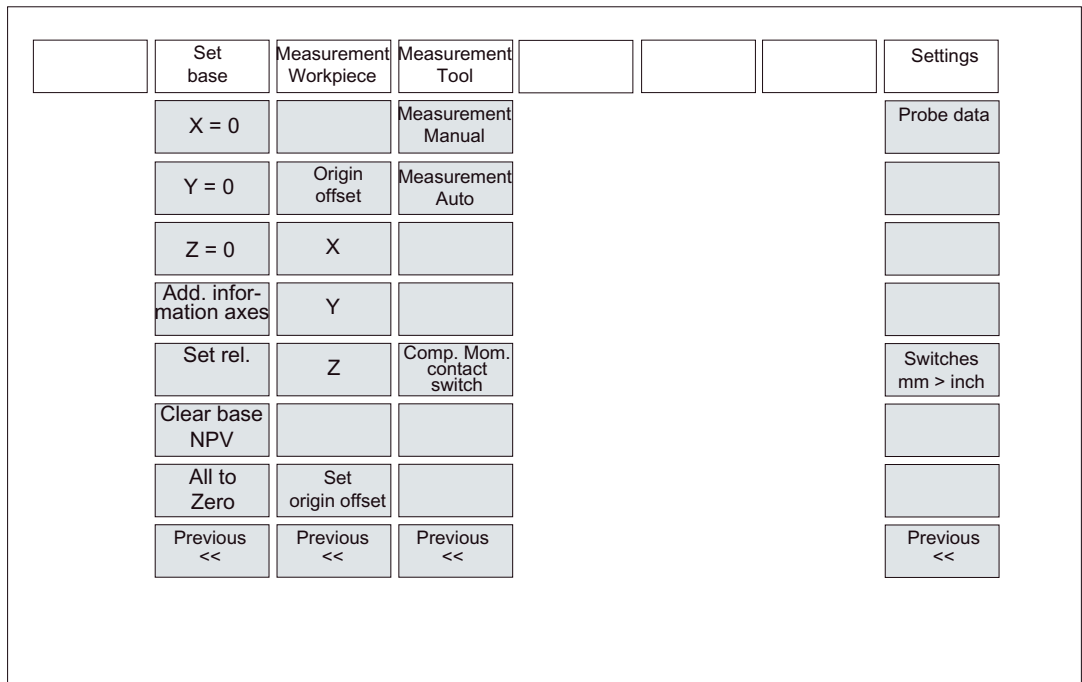


Figure 5-1 Menu tree for the JOG mode, "Position" operating area

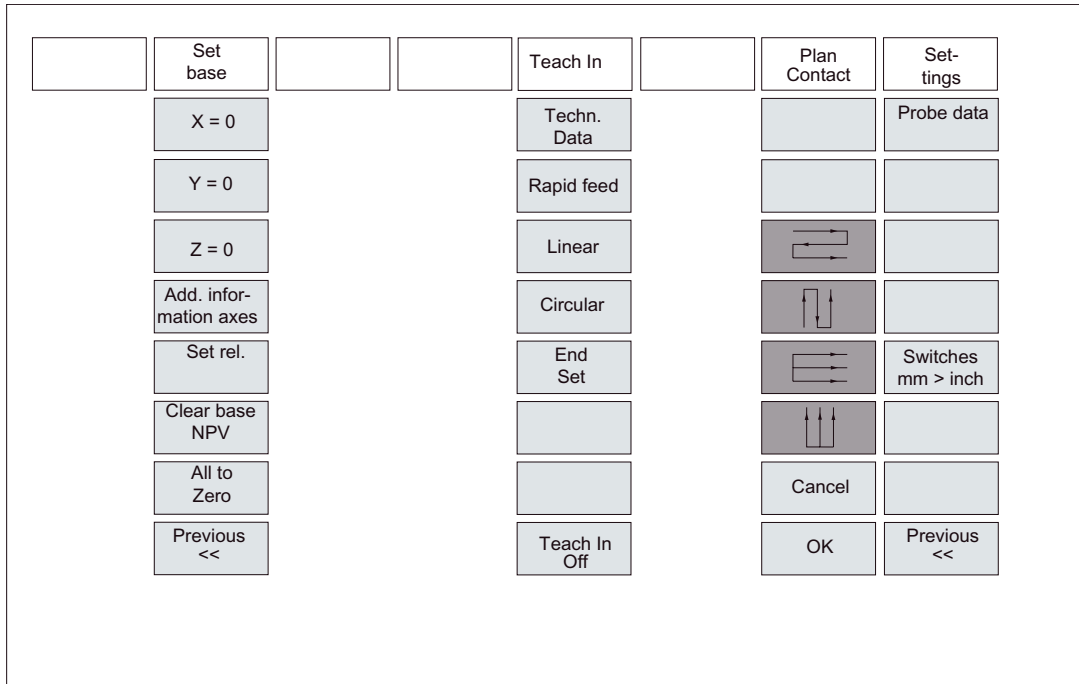


Figure 5-2 Menu tree for MDA mode, "Position" operating area

## 5.2 JOG mode - "Position" operating area

### Operating sequences



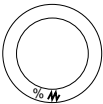
Use the <JOG> key on the machine control panel to select the Jog mode.



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



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.



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



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.

MCS	Position	Repos offset	T.F.S	G function
X1	0.000	0.000 mm	T 0	D 0
Y1	0.000	0.000 mm	F	0.000 100% 0.000 mm/min
Z1	0.000	0.000 mm	S1	0.0 110% 0.0 I
A1	0.000	0.000 mm		
G01	G500	G60		

SIEMENS

Reset

MCS / WCS REL

Handwheel

Settings

Set base Measure workpiece Measure tool

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 <i>Jog</i> mode, the distance traversed by each axis is displayed referred to the interruption point.
G function	Displays important G functions.
Spindle S rpm	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 workspindle 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:
  - in the idle state,
  - when the spindle starts,
  - if both spindles are active.
- The workspindle (small display) is displayed:
  - when starting the workspindle.

The power bar applies to the spindle currently active. With both master spindle and workspindle 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 "All to zero" softkey function overwrites the current position of the appropriate axis with zero.
- Setting individual axes to zero  
Selecting the X=0, "Y=0" or Z=0 softkey 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 entered as a reference point for the relative coordinate system.

It is useful to set a reference point "X=0" or "Y=0" and "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 (cf. "Setup" section)

Tool  
measurement

Use this softkey to measure the tool offsets (cf. "Setup" section)

Settings

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

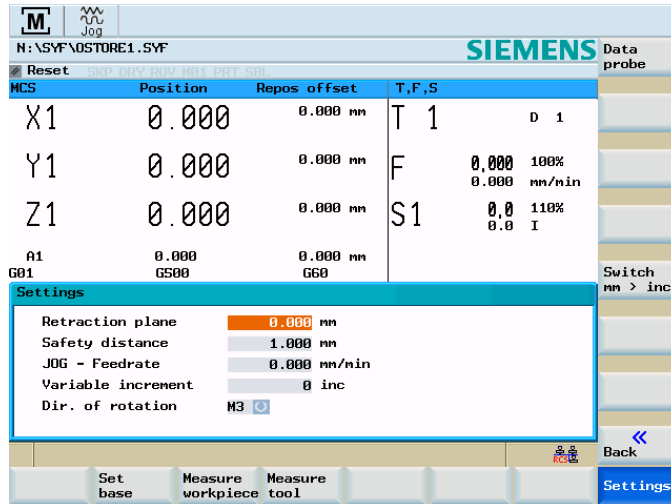


Figure 5-4 Settings

- **Retraction plane:**

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

- **Safety distance:**

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



Handwheel

Select the "Hand wheel" softkey in "Jog" mode. The "Hand wheel" 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 hand wheel using the cursor. Thereafter, press the relevant axis softkey for the desired axis for assignment or deselection

The  symbol appears in the window.

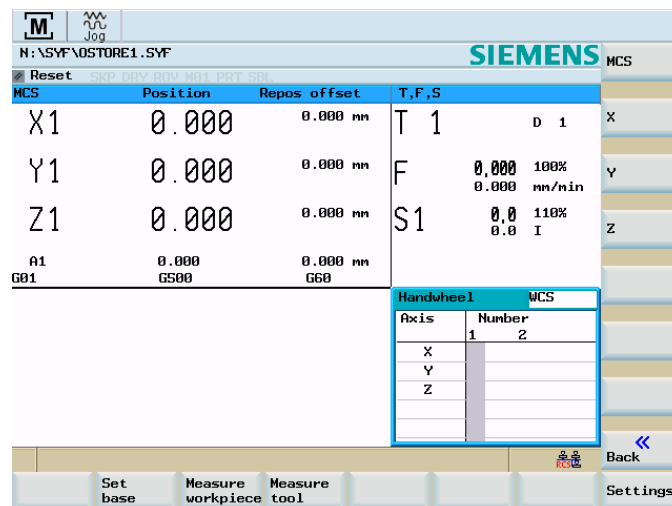


Figure 5-5 Hand wheel menu screen


MCS

Use the <MCS> softkey to select the axes from the machine or workpiece coordinate system for hand wheel 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.

 <b>CAUTION</b>
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 sequences



Select MDA mode through the machine control panel.

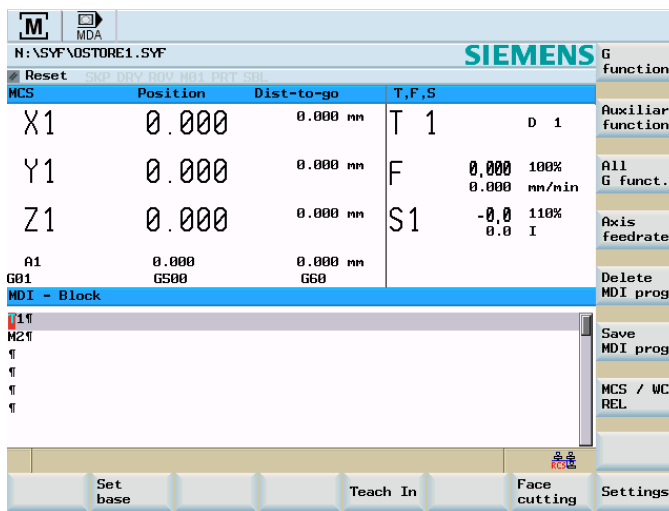


Figure 5-6 MDA start screen

Enter one or several blocks using the keyboard.



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 more.

## Parameter

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 rpm	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 workspindle 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 workspindle is displayed:
  - when starting the workspindle.

The power bar applies to the spindle currently active.

**Softkeys**

The horizontal softkeys are explained in the section titled "JOG mode - Position operating area" (Page 65).

G-function

The G function window includes G functions whereby each G function is assigned to a group and has a fixed position in the window. You can display more G functions using the keys <Page up> or <Page down>. 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.

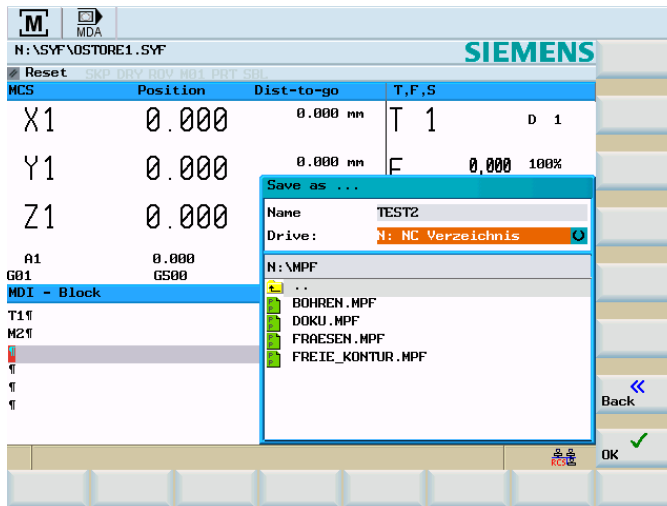


Figure 5-7 Saving an MDA program

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

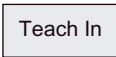
You can use the "Teach In" function to create and change simple traversing blocks. You can transfer axis position values directly into a newly generated or changed part program record.

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

#### Operating sequence



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



Press the <Teach In> softkey.

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

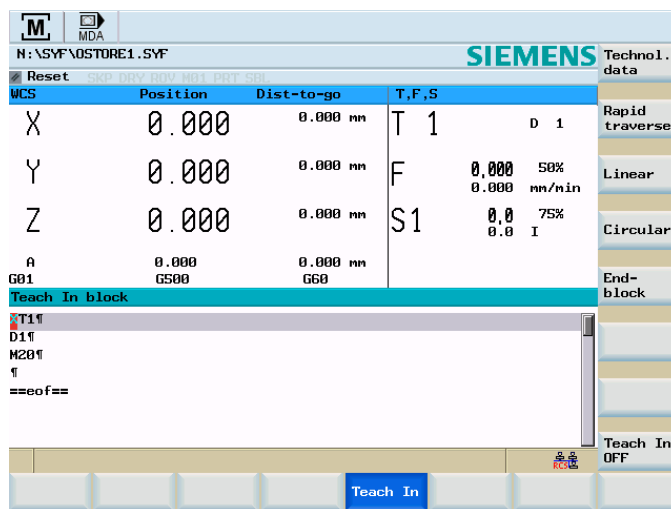


Figure 5-8 Teach In start screen

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"

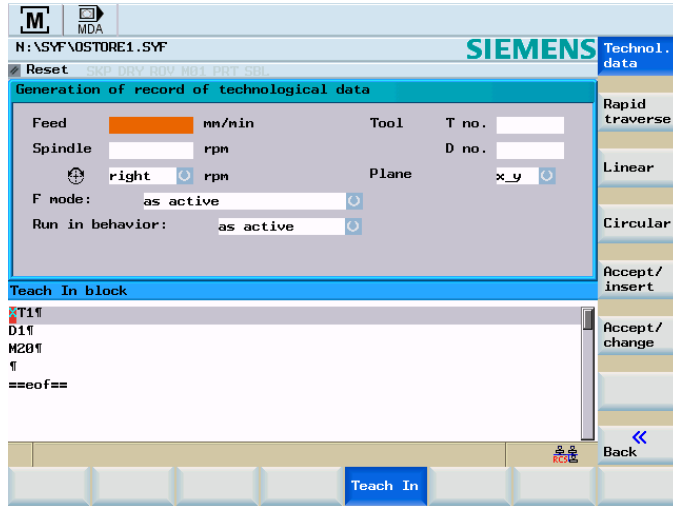


Figure 5-9 Technological data

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

Insert transfer

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

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

<< Back

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

Rapid traverse

– "Rapid feed"

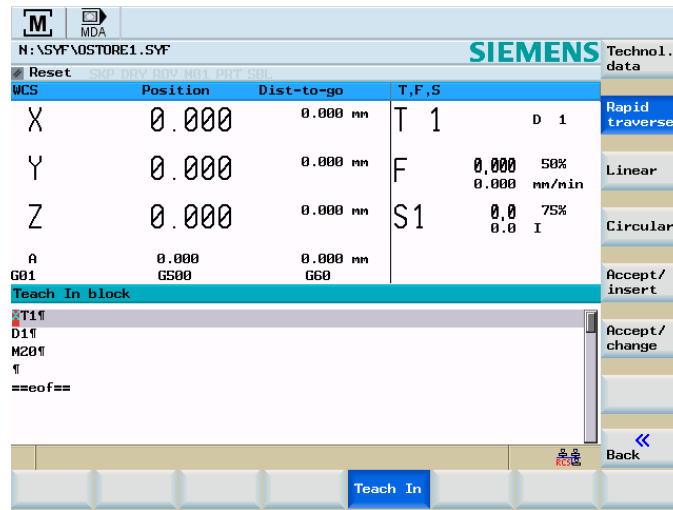


Figure 5-10 Rapid traverse

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

Linear

– "Linear"

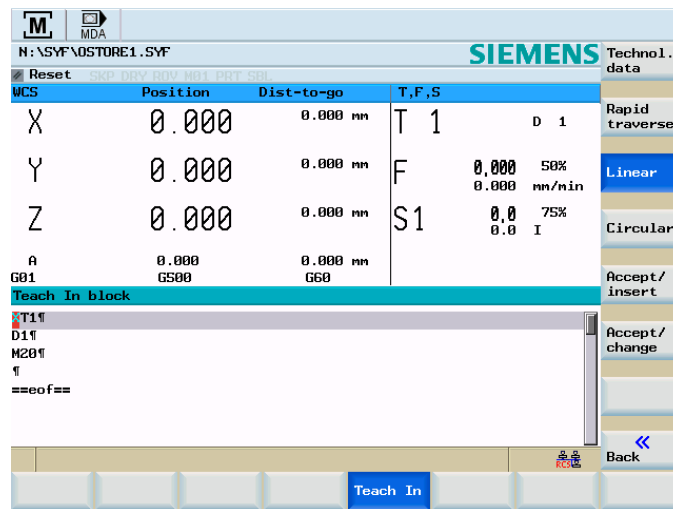


Figure 5-11 Linear

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

Circular

- "Circular"

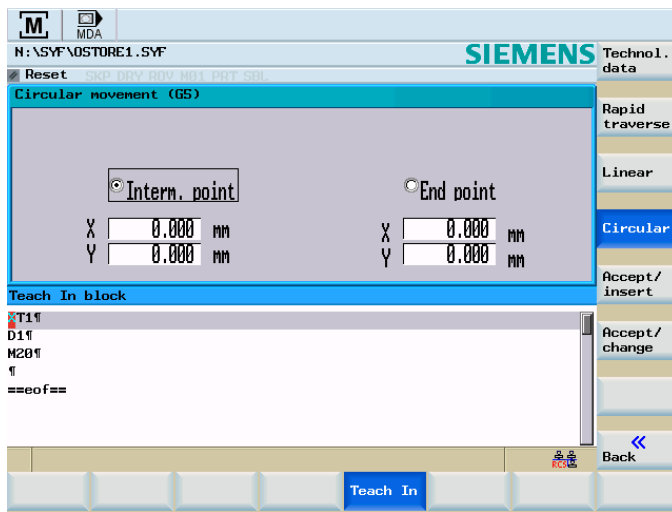


Figure 5-12 Circular

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

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

+X

+Y

Insert transfer

Change transfer << Back

Exit Teach In

1. Use the axis keys to traverse the axes to the required position that you want add/change in the part program.
  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.
  3. Click "Change transfer" to change the selected part program block.
- Use "<<Back" to return to the "Teach In" start screen.

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



## 5.3.2 Face milling

### Functionality

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

### Operating sequence



Planned machining

In MDA mode, select the Face softkey to open the input screen.

- Position the axes on the start point.
- Enter the values in the screen form.

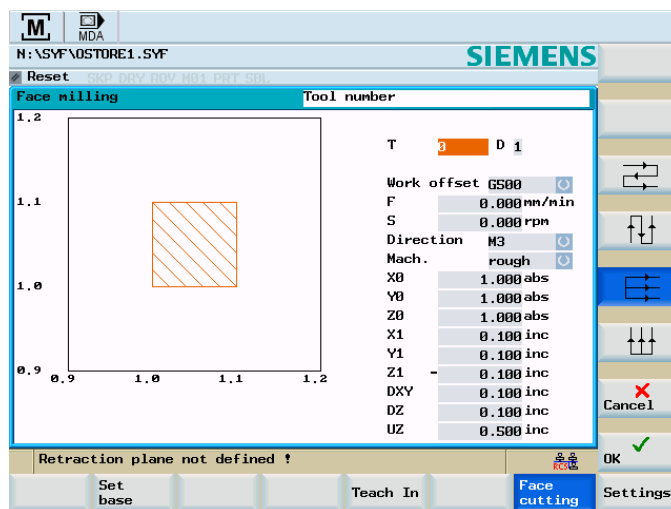


Figure 5-13 Face milling

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 "Machine" start screen will appear.

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 distance must be defined beforehand in the "Settings" menu.

Table 5- 3 Description of the parameters in the "Face milling" 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 (LL6) is provided by the machine manufacturer.
Work offset	Work offset to be selected in the program.
Feedrate F	Input of the path feedrate, in mm/min or mm/rev.
Spindle S rpm	Input of the spindle speed.
Direction	Use this softkey to select the direction of rotation of the spindle.
Contact:	Use this softkey to define the surface quality. You can select between roughing and finishing.
X0, Y0, Z0, X1, Y1 Blank dimensions	Use this softkey to enter the geometry of the workpiece.
Z1 Finished dimension	Finished dimension in Z
DXY Max. infeed	Input field for the amount of the infeed motion (X, Y)
DZ Max. infeed	Input field for the amount of the infeed motion (Z)
UZ	Input field for the stock allowance when roughing

**Softkeys to specify the machining direction**



Machining parallel to the abscissa, with alternating direction



Machining parallel to the ordinate, with alternating direction



Machining parallel to the abscissa, in one direction



Machining parallel to the ordinate, in one direction

## Automatic mode

### 6.1 AUTOMATIC mode

#### Menu tree

			Program Control	Set Search		Sim. recording	Program Compensation
			Program Test	Up Contour		Zoom Auto	
			Trial run Feedrate	Up End point		Zoom +	
			Cond. Hold	W/o calcul.		Zoom -	
			Surpress	Interrupt.		Show ...	
			Single block fine	Search		Display areas	
			ROV Effective			Deleting a screen	
						Cursor	
			<< Back	<< Back		<< Back	<< Back

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.

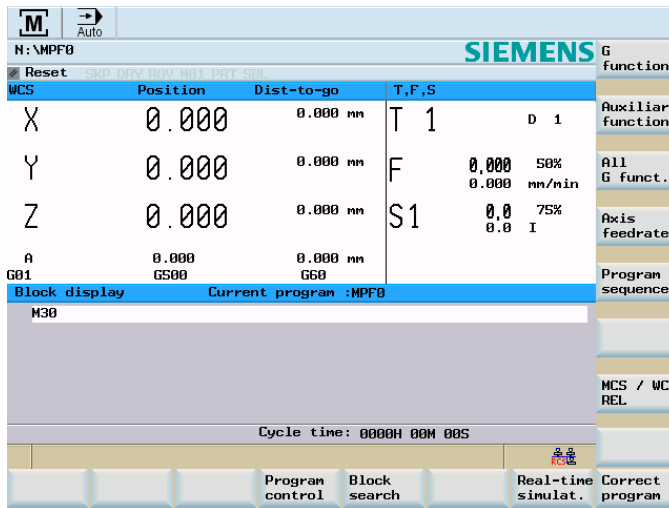


Figure 6-2 Automatic start screen

Parameter

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

Parameter	Explanation
MCS X Y Z	Displays the existing axes in the MCS or WCS.
+X Y - 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 rpm	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 workspindle 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 workspindle is displayed:
  - when starting the workspindle.

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

## Softkeys

G-  
function

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

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

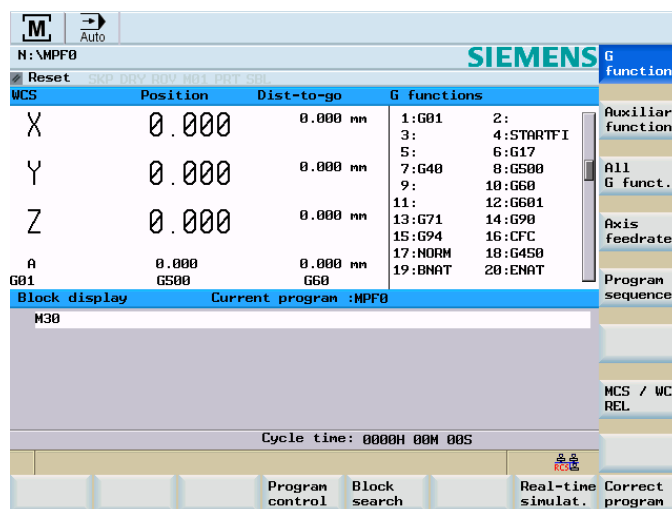


Figure 6-3 "G Functions" window

Use the <PageUp> or <PageDown> keys to display additional G functions.

Auxiliary function

The window displays the active help and M functions. Repeated pressing the softkey closes the window.

All G-functions

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

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 setpoint 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. The dry run feedrate function replaces the programmed travel commands.
- "Conditional stop": When this function is active, processing of the program is stopped at every block in which miscellaneous function M01 is programmed.
- "Skipping": Program blocks, the block number of which is preceded by a slash, are skipped during program execution (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 status.
- "ROV effective": 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

Use this softkey to correct a faulty program passage. Any changes will be stored immediately.

## 6.2 Selection and start of 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



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



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.

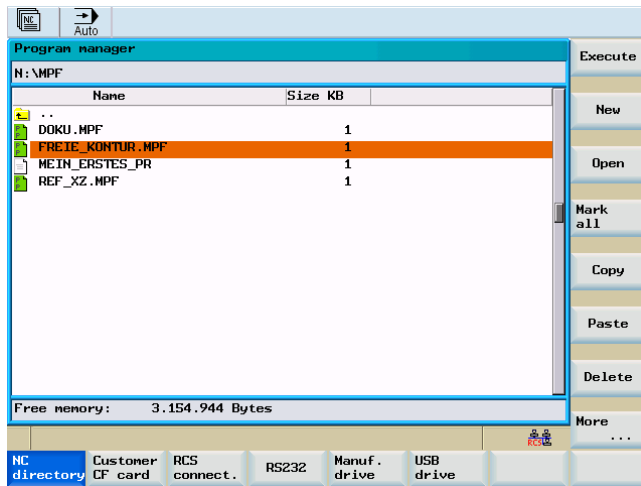
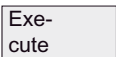


Figure 6-4 Program manager start screen

Place the cursor bar on the desired program.



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



Program control

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

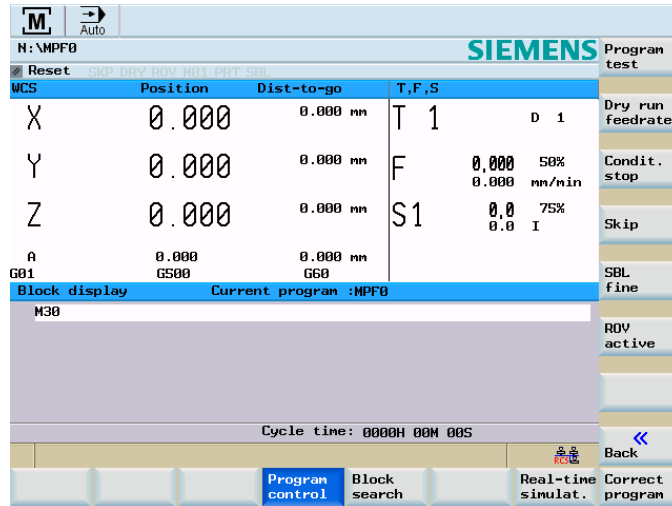


Figure 6-5 Program control



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.

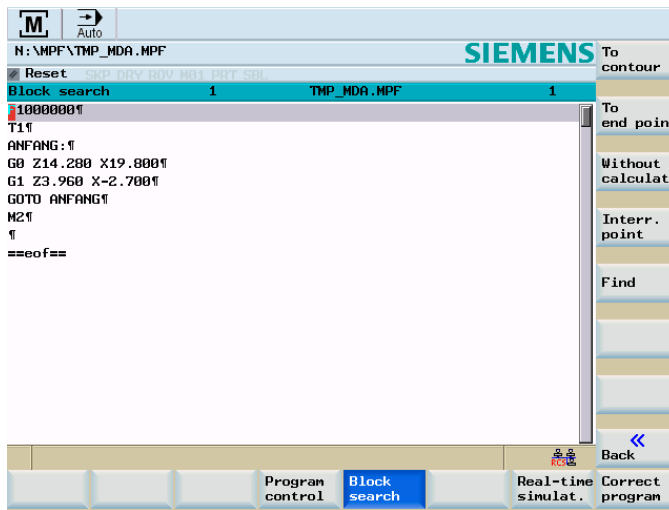


Figure 6-6 Block search

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.

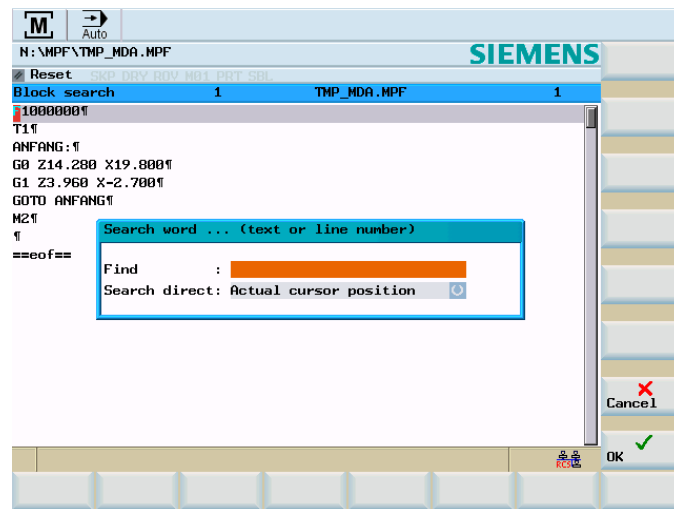


Figure 6-7 Enter search term

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



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.

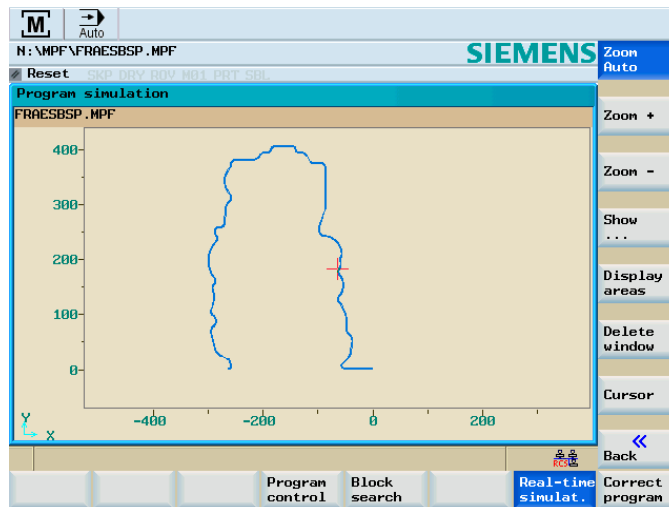


Figure 6-8 "Simultaneous recording" start screen

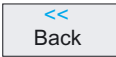
You can influence how the simultaneous recording function is displayed on the HMI using the following vertical softkeys:

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

See the following page for a description.

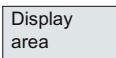
- "Delete window"
- "Cursor"
  - "Set cursor"
  - "Cursor fine", "Cursor coarse", "Cursor very coarse"

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



Exit the "Simultaneous recording" function.

### "Display areas"



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.

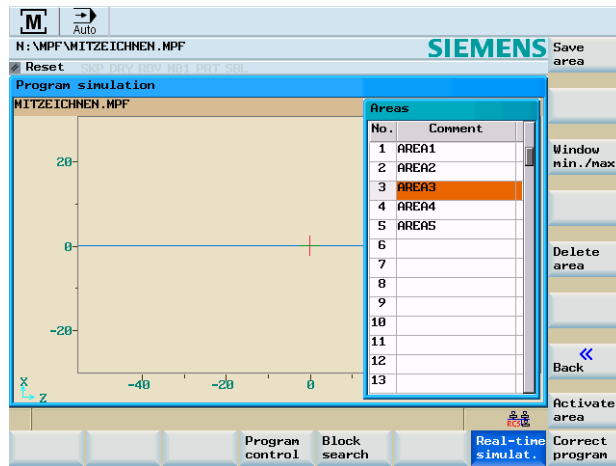


Figure 6-9 Display area "Window min"

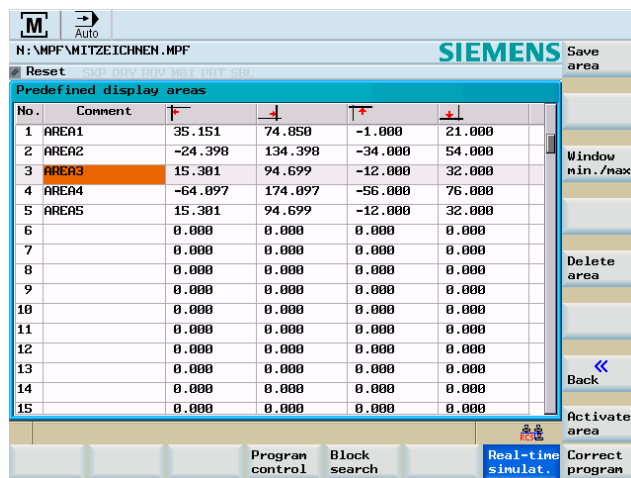


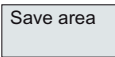
Figure 6-10 Display area "Window max"

### Operating sequence to set and save the display area

1. You have selected an area in the simulation view.
2. Press the "Display areas" function.



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.
5. Complete the entry with <Input>.



6. Press "Save area".

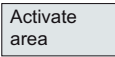
### Activating or deleting an area



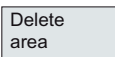
You have selected a display area.



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



Press "Activate area" or "Delete area".



## 6.5 Stop / cancel a part program

### Operating sequence



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



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

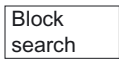
## 6.6 Reapproach after cancellation

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

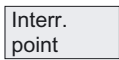
### Operating sequence



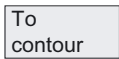
Select mode <AUTOMATIC> mode.



Opening the "Block search" window for loading the interruption point.



The interruption point is loaded.



The block search to the interruption point will start. An adjustment to the start position of the interrupted block will be carried out.



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



Select <AUTOMATIC> mode.



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



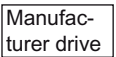
In <AUTOMATIC> mode > <PROGRAM MANAGER> operating area, the following interfaces are available for external execution of programs:



Customer CompactFlash card



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



Manufacturer's drive



USB FlashDrive

Start in the following start screen of the Program Manager:

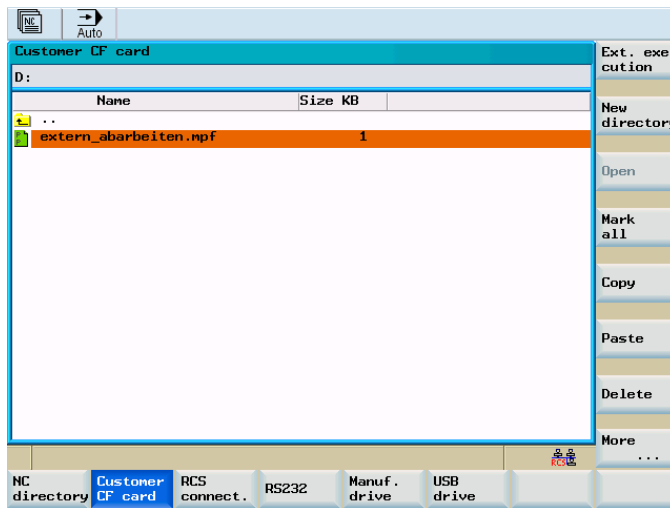


Figure 6-11 The "Program Manager" start screen

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

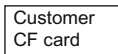
**Requirement:** The control system is in the "Reset" state.



Select the <AUTOMATIC> mode key .

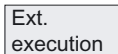


Press the <PROGRAM MANAGER> key on the machine control panel.



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.



Press "Ext. execution".

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



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 activated directory of the programming device/PC, for example: "\\123.456.789.0\External Program")

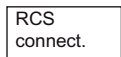
### Operating sequences for external execution via network



Select the <AUTOMATIC> mode key .

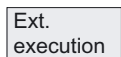


Press the <PROGRAM MANAGER> key on the machine control panel.



Press "RCS connect.". You go to the directories of the PG/PC.

Place the cursor bar on the desired program.



Press "Ext. execution".

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



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

### 7.1 Part programming overview

#### Menu tree

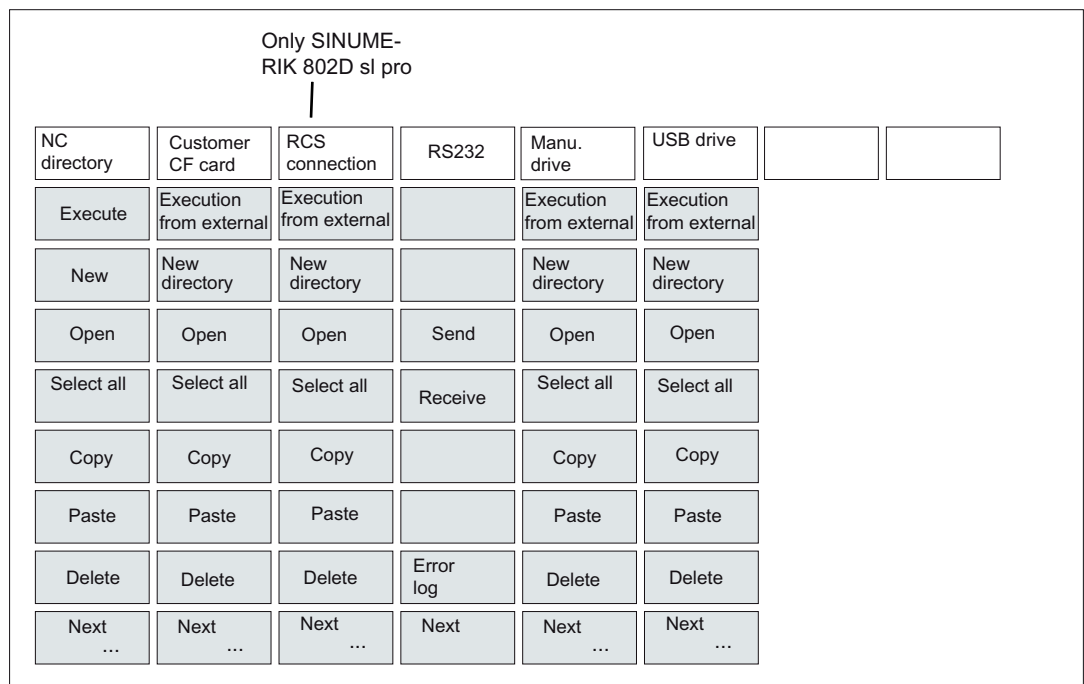


Figure 7-1 "Program Manager" menu tree

#### 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



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

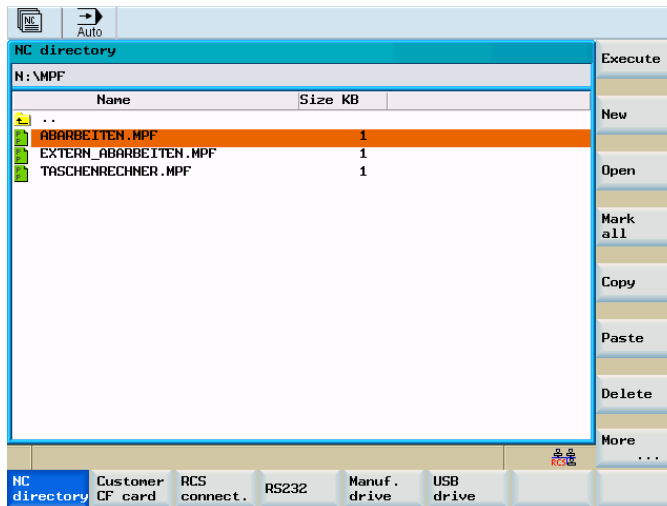
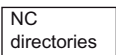


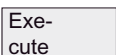
Figure 7-2 The "Program Manager" start screen

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.

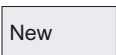
### Softkeys



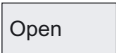
Use this softkey to display the directories of the NC.



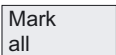
Use this softkey to select the program on which the cursor is placed for execution. The control system will switch to the position display. Use <NC START> to start this program.



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



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



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



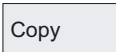
---

### 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.

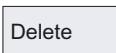
---



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

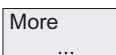


This function will paste files or directories from the clipboard to the current directory.

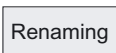


When selecting the "Delete" softkey, the file selected by the cursor is deleted after a confirmation warning. If several files have been selected, all these files will be deleted after a confirmation warning.

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

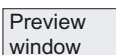


Use this softkey to branch to further functions.

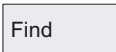


A window opens where you can rename the program you have selected beforehand using the cursor.

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

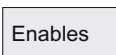


This function opens a window displaying the first seven lines of a file if the cursor has been positioned on the program name for a certain time.



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

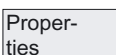
After you have entered the name, either press "OK" to confirm or "Abort" to cancel.



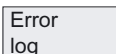
A selected directory can be released for network operation.



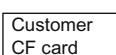
The function splits the window on the HMI. You can use the <Tab> key to switch over between windows.



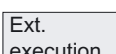
The function gives information on the properties of the memory of the selected directory and of the selected file.



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.



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.



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

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

New

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

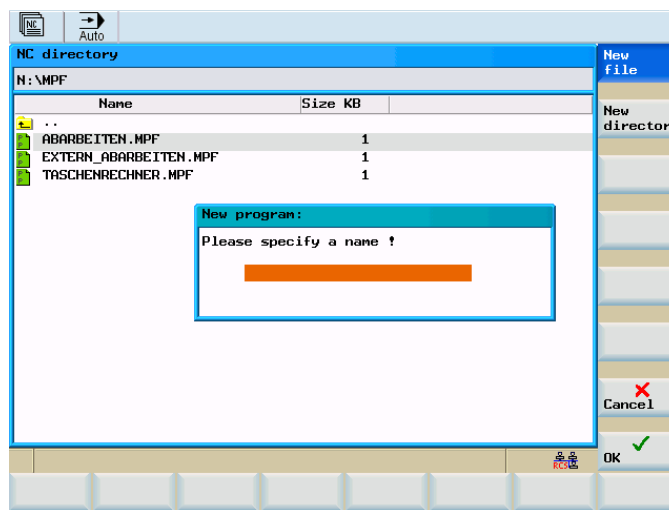


Figure 7-3 New program

New  
directory

After pressing 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 pressing 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.

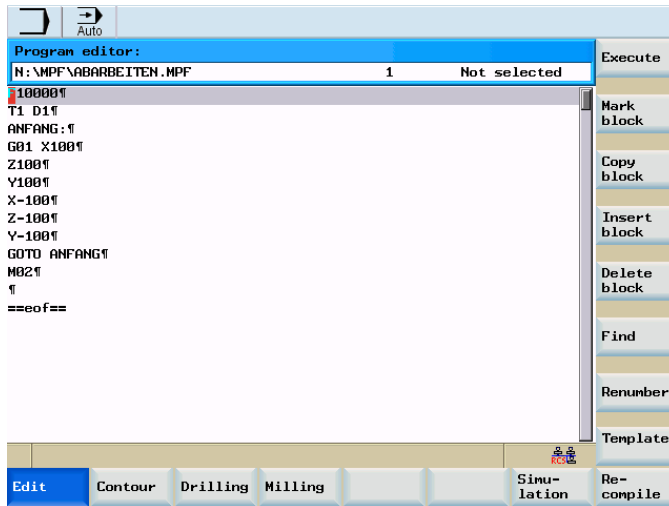


Figure 7-4 Program editor start screen

## Menu tree

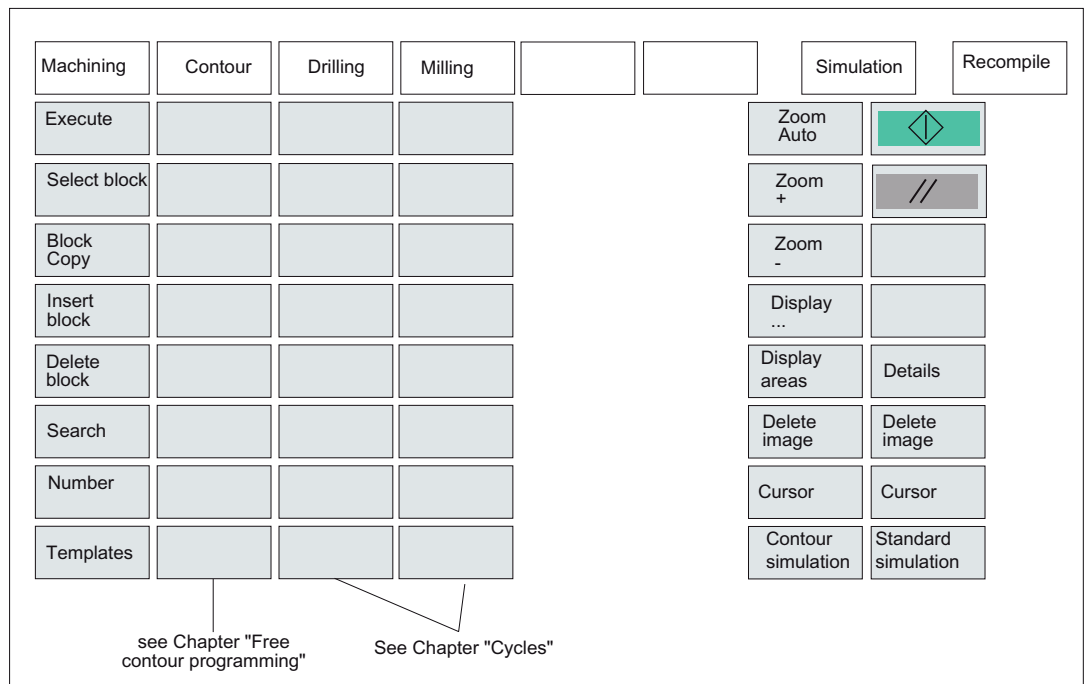
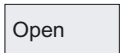


Figure 7-5 "Program" menu tree

## Operating sequence



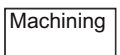
In the PROGRAM MANAGER operating area, select the program to be edited.



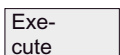
Press "Open".

The program is opened and displayed for editing. Additional softkey functions are provided. Program changes are automatically applied.

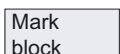
## Softkeys



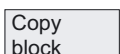
Function for editing text segments.



Use this softkey to execute the selected file.



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



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

Milling

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	Free contour programming, see "Free contour programming" section.
Bore	See Section "Cycles"
Milling	See Section "Cycles"
Simu- lation	The simulation is described in the "Simulation" section.
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. "Drilling" > "Drilling center." -> 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

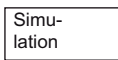
### Functionality

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

### Operating sequence



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



The start screen is opened.

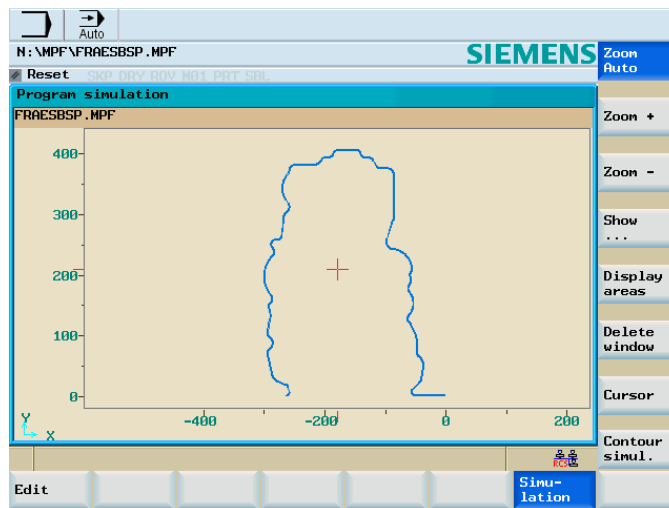


Figure 7-6 Standard simulation

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



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.

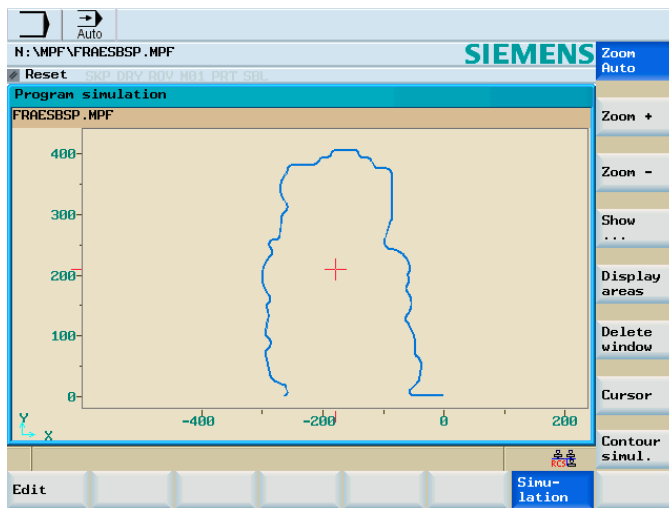


Figure 7-7 Standard simulation

### Softkeys for the standard simulation

You can influence how the standard simulation 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"
- Defines the simulation displayed on the HMI to specific areas (see Chapter "Simultaneous recording" (Page 88)).
- "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.

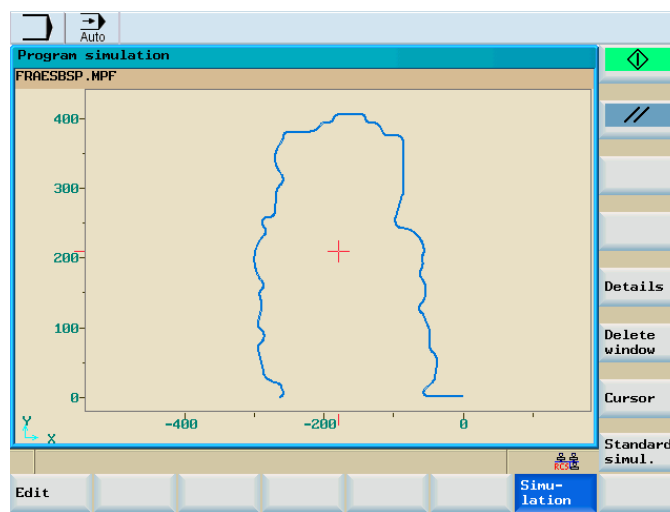


Figure 7-8 Contour simulation

## 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"

Standard simulation

When the cursor keys are pressed, the cross hair moves in small, average or large steps. Switches into the "standard simulation".

**See also**

Simultaneous recording (Page 88)



## 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.



The <SHIFT> and <=> or <CTRL> and <A> key combination activates the calculator (Page 513).

### Note

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

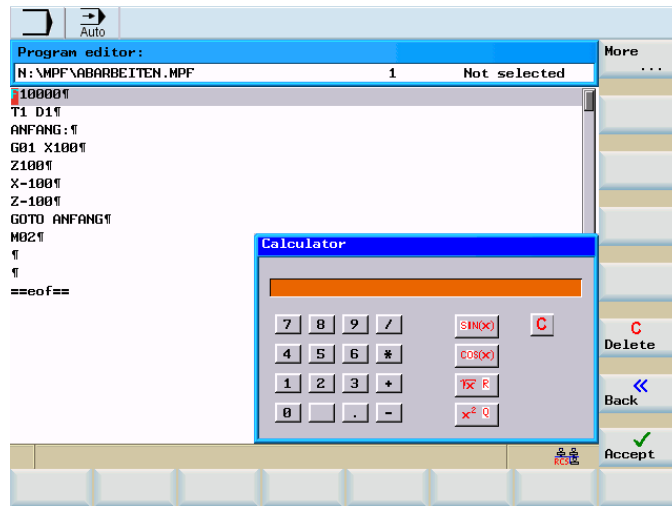


Figure 7-9 Pocket calculator

More  
...

Functions for editing contour elements are available by pressing "Next ...".

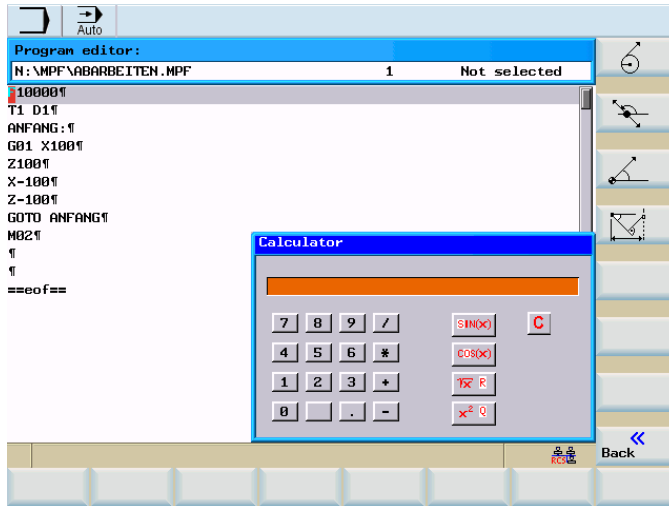


Figure 7-10 Pocket calculator > Next ...

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.

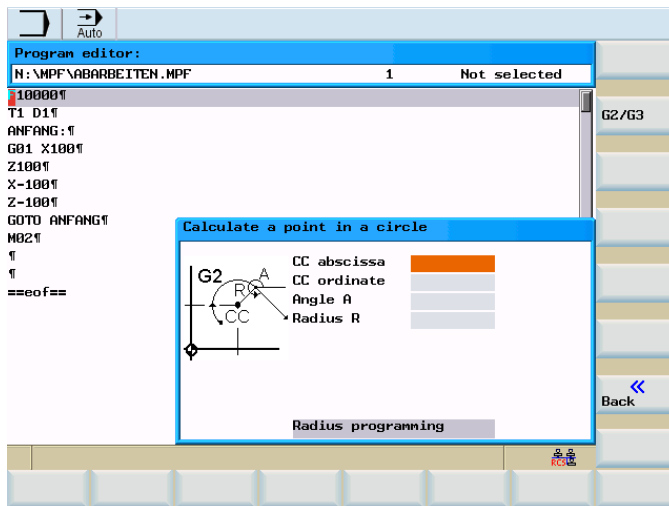


Figure 7-11 Calculate: Point on a circle

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.



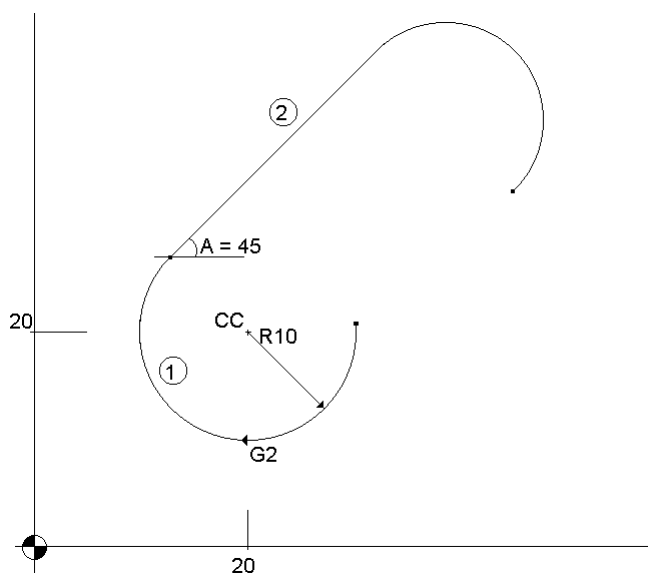
Use this softkey to calculate the abscissa and ordinate values. The abscissa is the first axis, and the ordinate is the second axis of the plane. The abscissa value is copied into the input field from which the calculator function has been called, and the value of the ordinate is copied into the next 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

*If the G17 plane is active, the abscissa is the X-axis and the ordinate is the Y-axis.*

*Calculating the point of intersection between the ① circle sector and the ② straight line.*

Existing conditions:	Radius:	10
	Circle center:	X20 Y20
	Connection angle of the straight line:	45°
	Direction of rotation:	G2



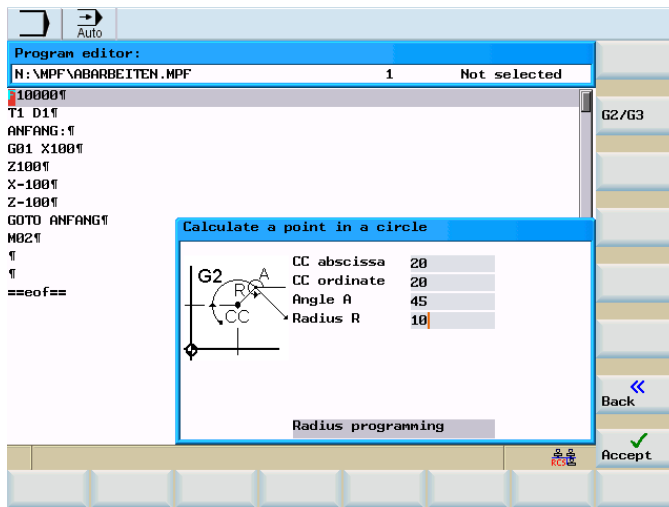
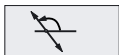


Figure 7-12 Screen form

Result: X=12.928  
Y=27.071



This function calculates the Cartesian coordinates of a point in the plane, which is to be connected to a point in the plane (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.

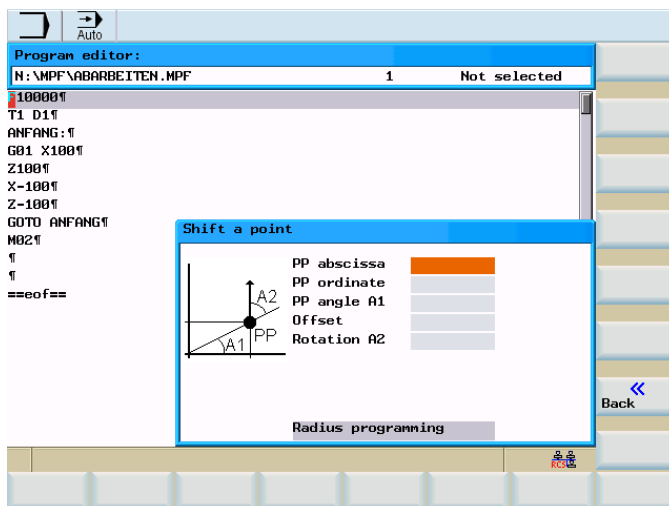


Figure 7-13 Calculate: point in the plane

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



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.

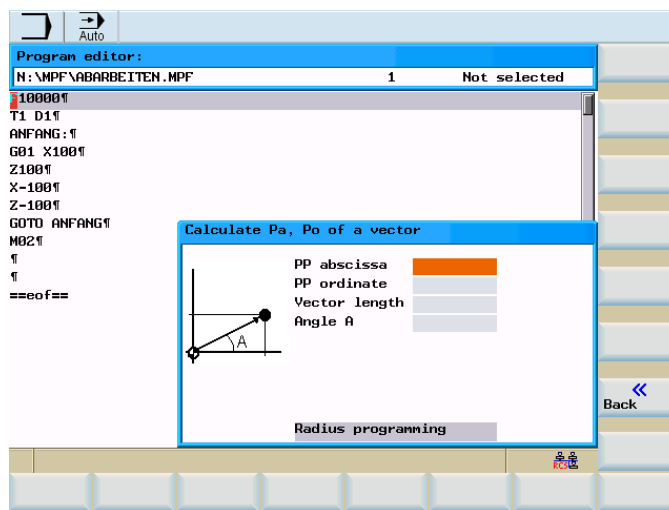


Figure 7-14 Converting polar coordinates to Cartesian coordinates

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



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.

Example

Calculating the end point of the straight line (1). The straight line is determined by the angle  $A=45$  degrees and its length.

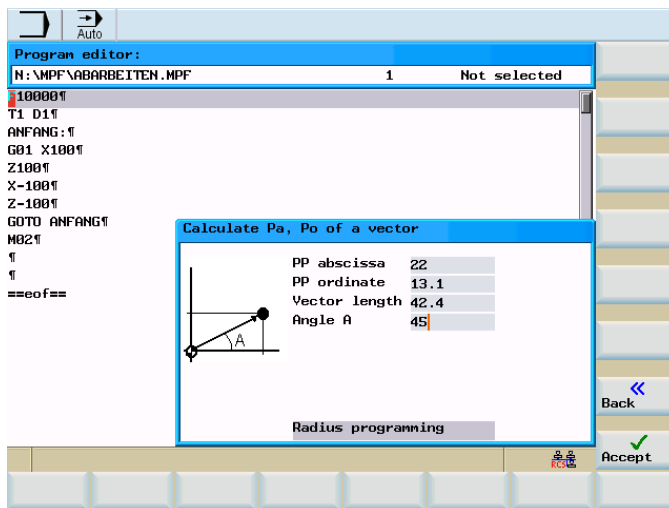
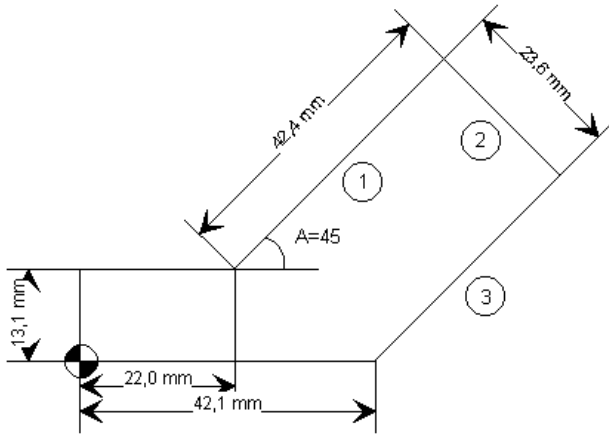


Figure 7-15 Screen form

Result: X=51.981  
Y=43.081



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: start point and gradient

straight line 2: length and one end point in the Cartesian coordinate system

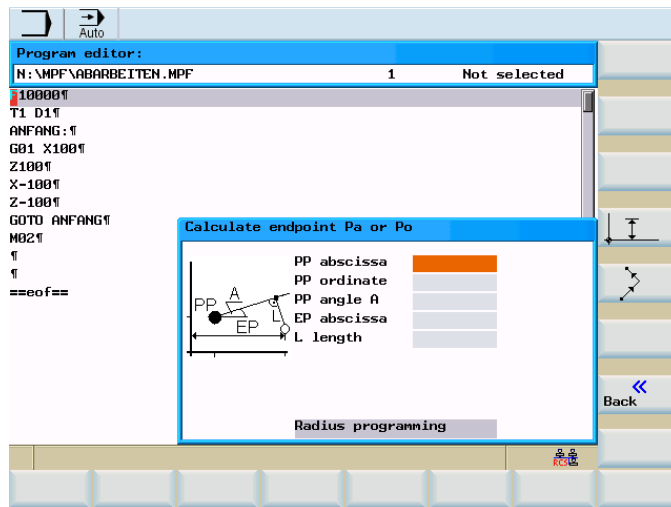
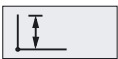


Figure 7-16 Calculate: missing end point



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 turned 90 degrees clockwise or

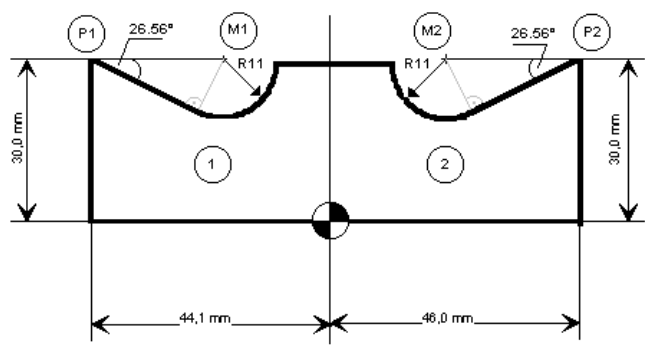


counterclockwise compared with the first straight line.

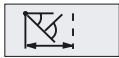


The missing end point is calculated. Copy the abscissa value into the input field from which you have called the calculator function. Copy the ordinate value into the next input field. If the function has been called from the part program editor, the coordinates are saved with the axis names of the base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

## Example



The value of the center circle must be added to the existing drawing in order to be able to calculate the intersection point between the circle sector of the straight line.



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

### Calculating M1 in section 1:

The radius in this section stands rotated in the clockwise direction on the straight line section.

The radius stands at an angle of 90° turned clockwise on the straight-line defined by the angle.



Use the softkey "Straight-line turned clockwise" to select the appropriate direction of rotation.

Use the softkey "Coordinate value or abscissa value defined" to define the given end point.

Enter the coordinates of the pole P1, the slope angle of the straight line, the ordinate angle of the end point and the circle radius as the length.

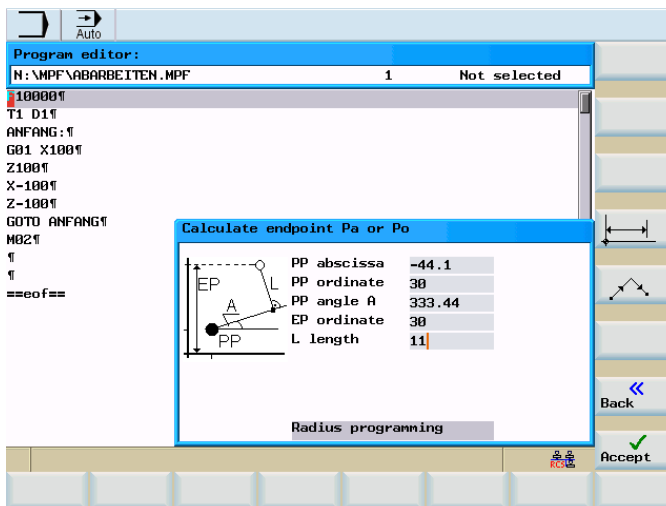


Figure 7-17 M1 input screen form

Result: X=-19.499  
Y=30

### Calculating M2 in section 2:

The radius in this section stands rotated in the clockwise direction on the straight line section.



Use the softkey "Straight-line turned clockwise" to select the appropriate direction of rotation.



Use the softkey "Coordinate value or abscissa value defined" to define the given end point.

Enter the parameters in the screen form.



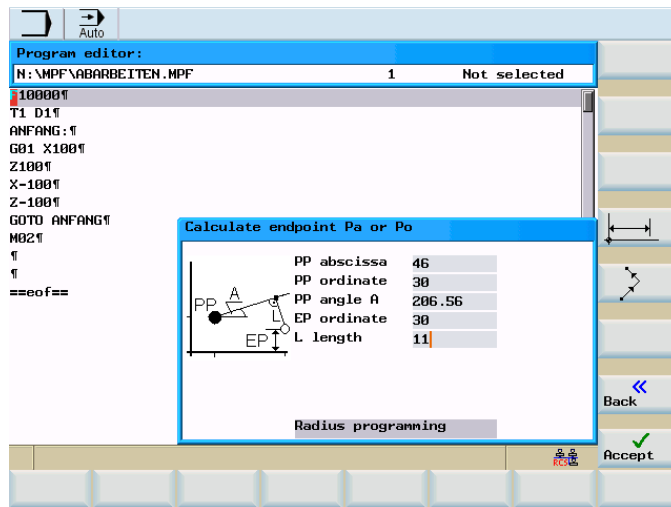


Figure 7-18 M2 input screen form

Result: X=-21.399  
Y=30

## 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 program editor (FCI) 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.

---

#### Note

The display MD 290 \$MA CTM\_POS\_COORDINATE\_SYSTEM defines the display of the coordinate system.

A value of 4 means:

Ordinate = Y axis

Abcissa = X axis

---

### Technology

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

- 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 in general, undercut)

## Start screen of the contour editor (FKE)

PROGRAM  
MANAGER

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

Contour

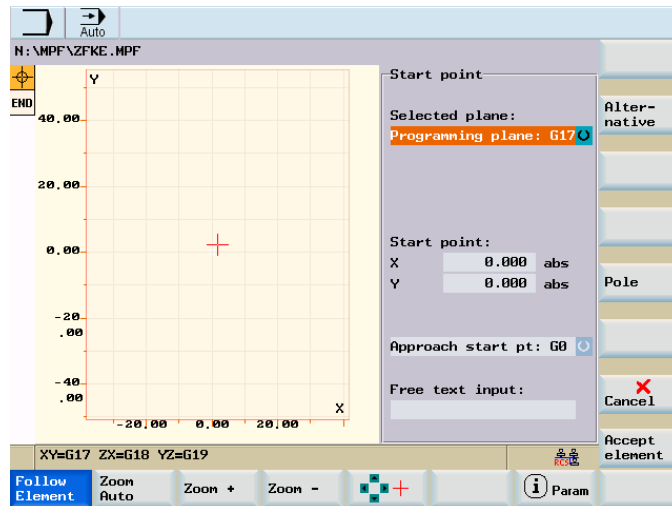


Figure 7-19 Define a start point

Initially, you define a contour starting point (see Chapter "Defining a starting point (Page 123)").

The contour is then programmed step-by-step (see Chapter "Programming example milling (Page 139)").

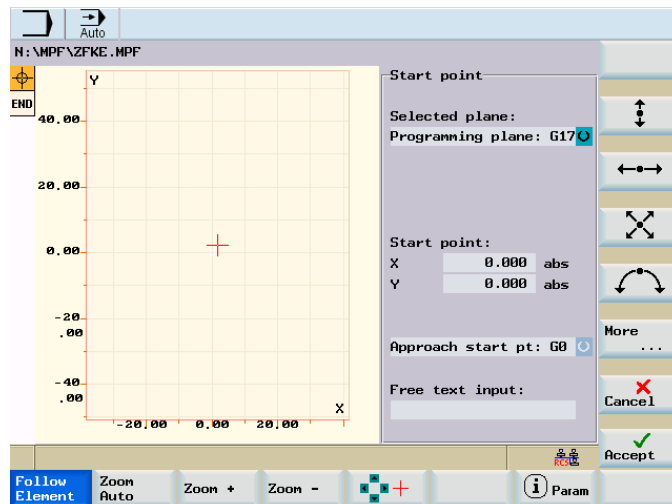


Figure 7-20 Edit contour elements

### 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).

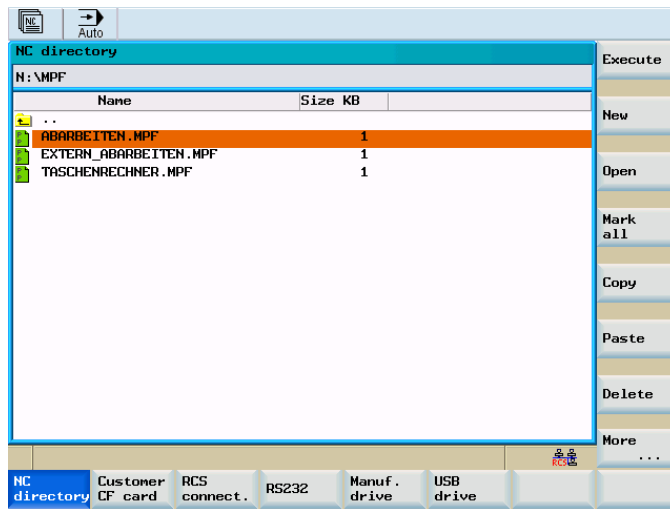


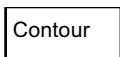
Figure 7-21 The "Program Manager" start screen

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.



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



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.

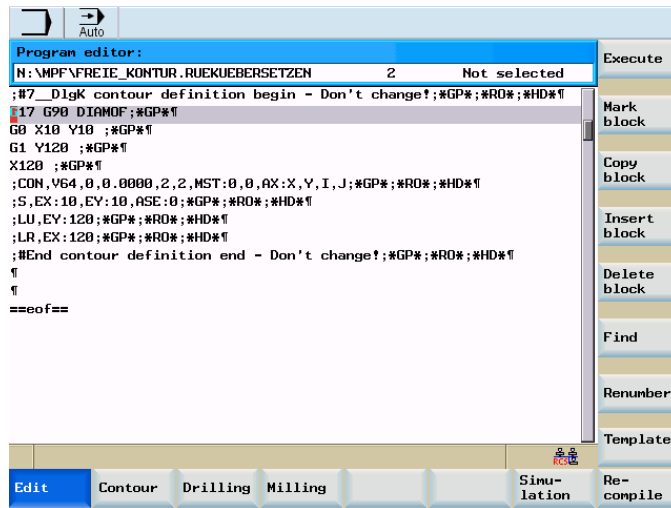


Figure 7-22 Recompile

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.

<b>NOTICE</b>
<p>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.</p> <p>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.</p>

## 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).

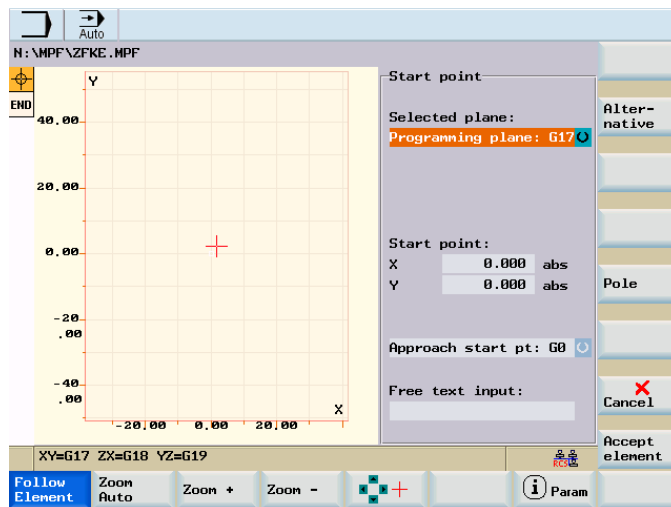


Figure 7-23 Define a start point

#### 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  $\uparrow$ ,  $\downarrow$  cursor keys. The current position in the chain is color-highlighted.

1. Select programming plane G17 for the milled 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.

**1** ... **0**

2. Enter the values for the starting point.

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

3. 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

4. Press the "Accept element" softkey.

The start point is saved.

You can add the next element using softkey commands (see next section "Define 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:

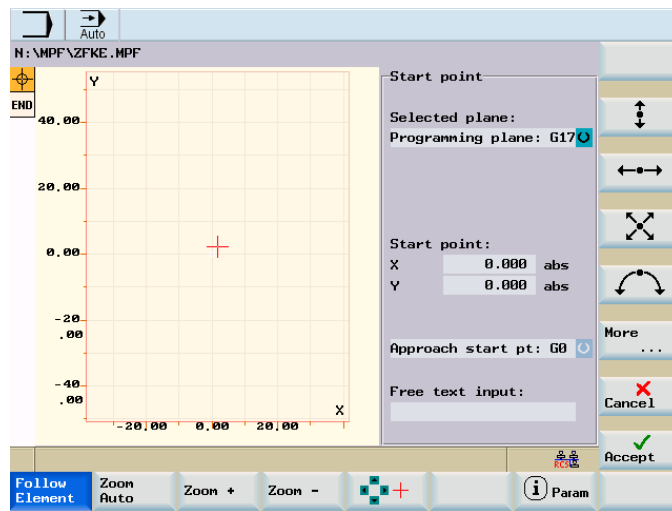


Figure 7-24 Define contour elements

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 (Y direction)



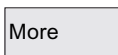
Straight line in the horizontal direction (X direction)



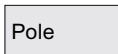
Oblique line in the Y/Z direction. Enter the end point of the line using coordinates or an angle.



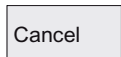
Arc with any direction of rotation.



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



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.



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



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+").

An element was selected using the cursor keys.



The "Follower element" enlarges the image section on 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.



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.

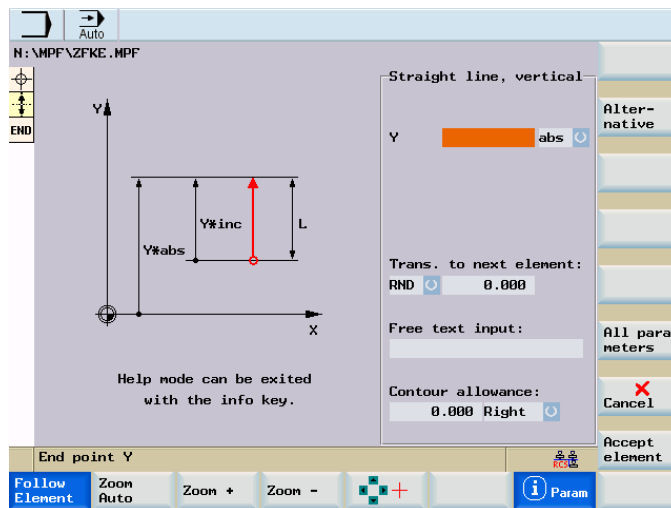


Figure 7-25 Help mode

## Parameter

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

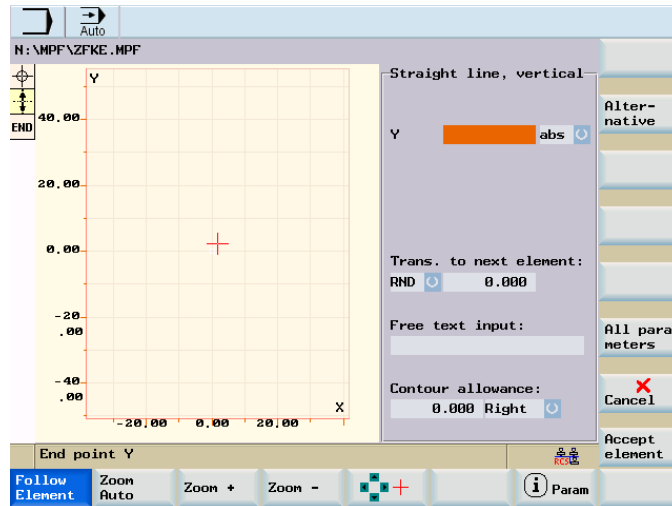


Figure 7-26 Straight line in vertical direction

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 end of a 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:

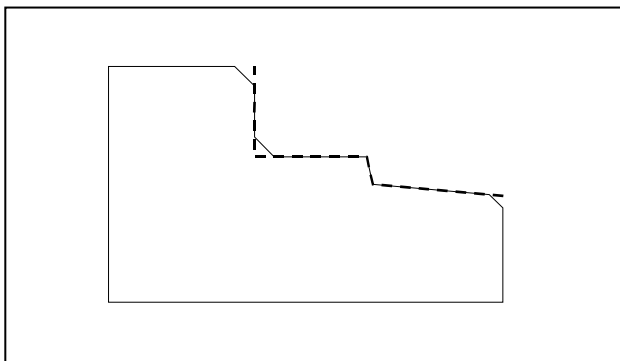


Figure 7-27 Contour with radius or chamfer

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 predecesing 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  $\uparrow$ ,  $\downarrow$  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.

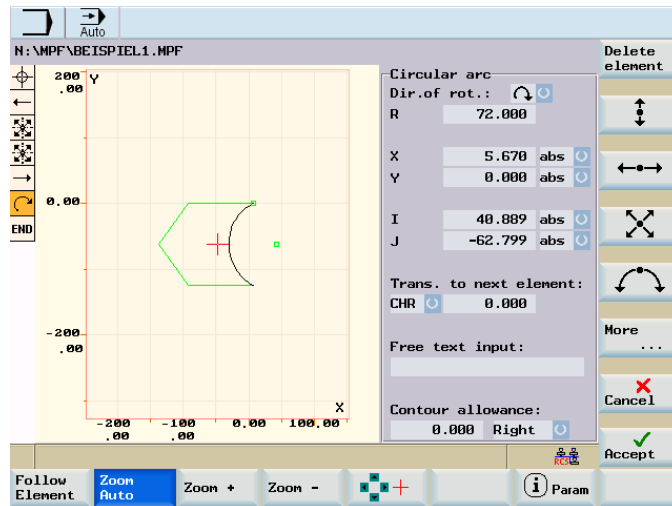


Figure 7-28 Edit contour elements



You can select an existing contour element with the <Input> key and reparameterize it again.

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 is selected in the contour chain.

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



### 7.6.4 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  $\alpha_2$  to a value of 0. The contour element has a tangential transition to the preceding element, i.e. the angle to the preceding element ( $\alpha_2$ ) 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



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



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.

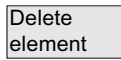
Then select the contour element to be inserted from the softkey bar.

Accept  
element

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

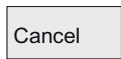
Subsequent contour elements are updated automatically according to the new contour status.

### Delete contour element



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



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



## 7.6.5 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 chapter "Program contour".

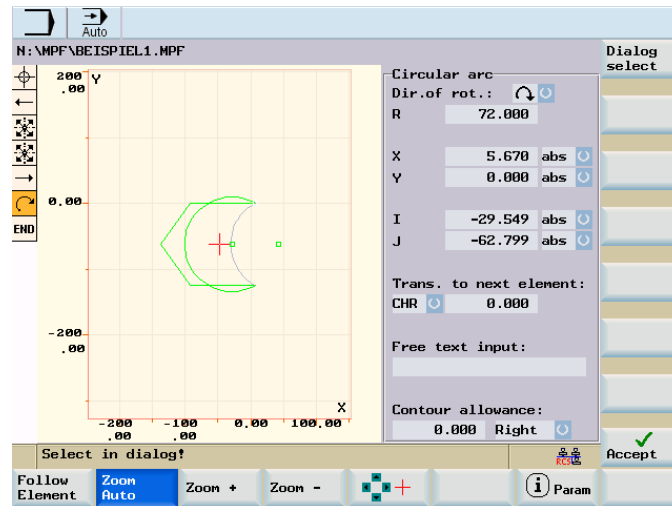


Figure 7-29 Contour with arrow

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.

An element was selected using the cursor keys.

The "Follower element" enlarges the image section on the selected element.

Follow  
Element

### Contour allowance

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

## 7.6.6 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.

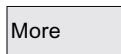
### Pole



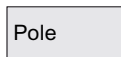
The polar coordinates are valid in the level selected with G17 to G19.

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.

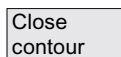
### Enter polar coordinates



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



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.



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)

For switchover between Cartesian and polar, additional toggle fields are displayed for "Straight line in any direction" and "Circular arc" in both the basic contour input view and 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  $0^\circ \dots \pm 360^\circ$ . 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

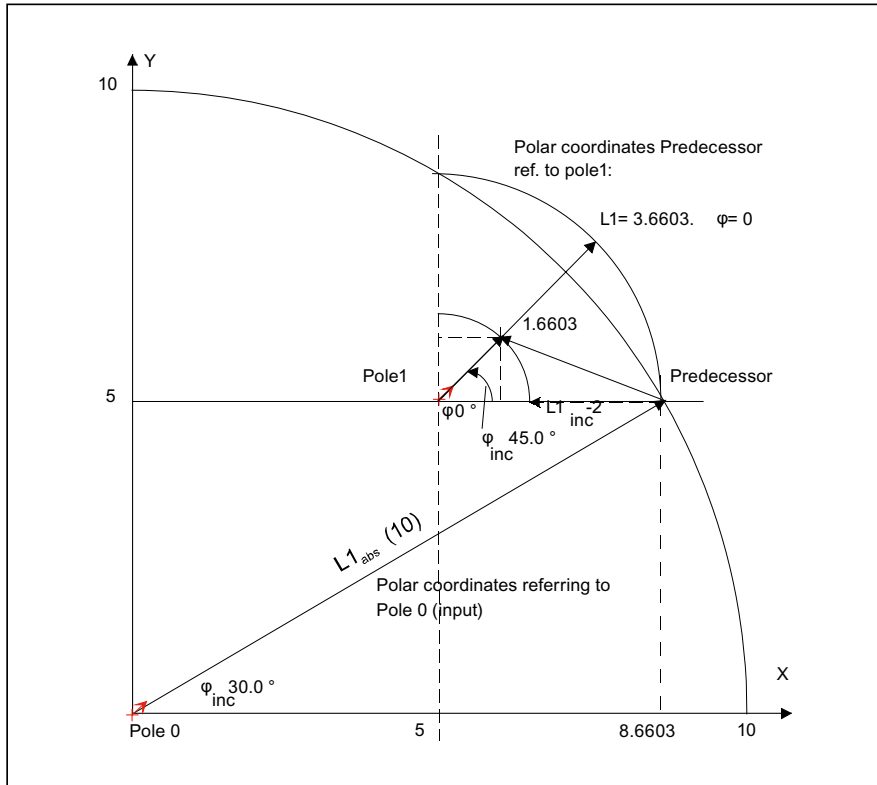


Figure 7-30 Pole change

<b>Pole:</b>	XPole = 0.0,	YPole = 0.0	(Pole 0)
End point:			
L1abs = 10.0	phiabs = 30.0°	Calculated Cart. Coordinates	
		Xabs = 8.6603	Yabs = 5.0
<b>New pole:</b>			
Xpole1 = 5.0	YPole1 = 5.0		(Pole 1)
		Calculated polar coord. Predecessor	
		L1abs = 3.6603	phiabs = 0.0°
<b>Next point:</b>			
L1inc = -2.0	phiinc = 45.0°		
		Absolute polar coordinates for current element	
		L1abs = 1.6603	phiabs = 45.0°
		Calculate Cartesian coordinates	
		Xabs = 1.1740	Yabs = 1.1740

## 7.6.7 Parameter description of straight line/circle contour elements

### Parameters for contour element "Straight line"

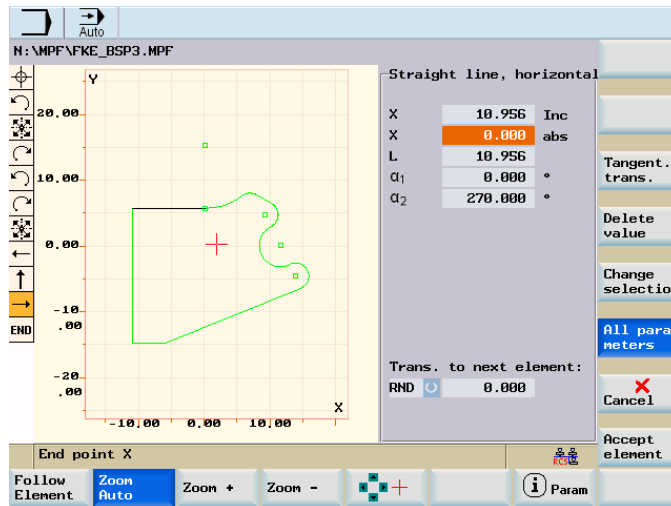


Figure 7-31 Straight horizontal line

Parameter	Contour element "Straight line"
Y inc	Incremental end position in Y direction
Y abs	Absolute end position in Y direction
X inc	Incremental end position in X direction
X abs	Absolute end position in X direction
L	Length of straight line
$\alpha_1$	Pitch angle with reference to Y axis
$\alpha_2$	Angle to preceding element; tangential transition: $\alpha_2=0$
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"

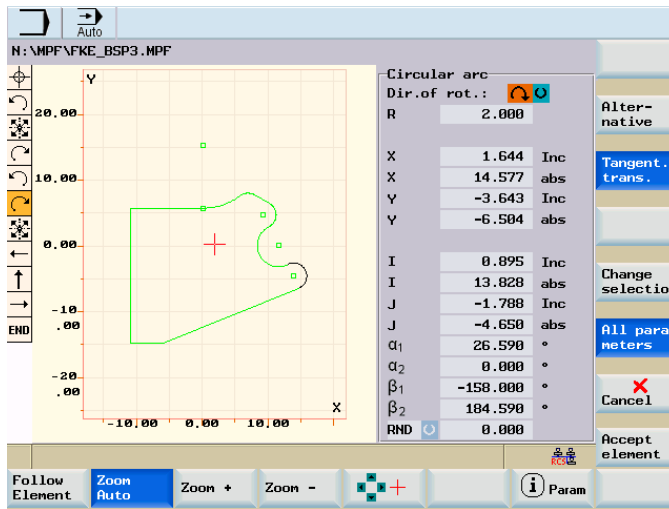


Figure 7-32 Circular arc

Parameter	Contour element "Circle"
Direction of rotation	In clockwise or counter-clockwise direction
R	Radius of circle
Y inc	Incremental end position in Y direction
Y abs	Absolute end position in Y direction
X inc	Incremental end position in X direction
X abs	Absolute end position in X direction
I	Position of circle center point in Y direction (abs. or incr.)
K	Position of circle center point in X direction (abs. or incr.)
$\alpha_1$	Start angle with reference to Y axis
$\alpha_2$	Angle to preceding element; tangential transition: $\alpha_2=0$
$\beta_1$	End angle with reference to Y axis
$\beta_2$	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 (Y or X ...) are defined in the machine data where they can also be changed.

## 7.6.8 Programming example for milling

### Example 1

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

Starting point: X=5.67 abs., Y=0 abs., machining plane G17

The contour is programmed in a counter-clockwise direction.

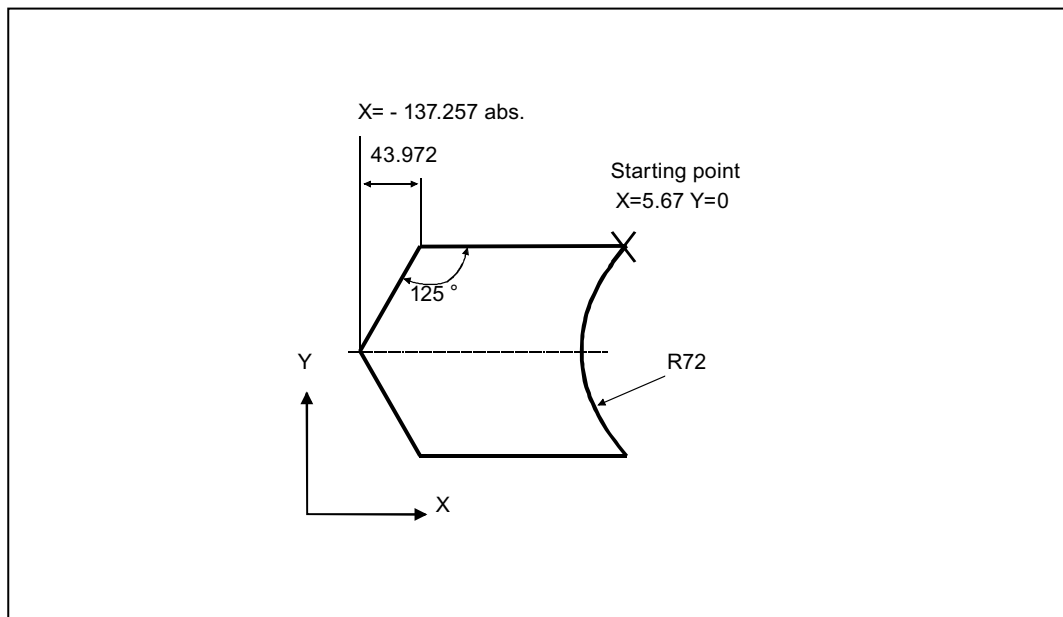


Figure 7-33 Workshop drawing of the contour, example 1

### 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.

---

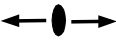


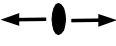

#### Note

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  $\uparrow$ ,  $\downarrow$  cursor keys. The current position in the chain is color-highlighted.

You can call each input screen with the "Input" key and enter new parameters.

---

Table 7- 1 Example 1 operating sequences

Operating step	Softkey	Parameter
1	"Contour" "Accept element"	Enter starting point: programming level G17 X=5.67 abs. Y=0
2	 "Accept element"	Enter parameters for "Straight horizontal line" element: X=-93.285 abs.
3	 "Accept element"	Enter parameters for "Straight line in any direction" element: X=-43.972 inc. $\alpha 1 = -125$ degrees
4	 "Accept element"	Enter parameters for "Straight line in any direction" element: X=43.972 inc. $\alpha 1 = -55$ degrees
5	 "Accept element"	Enter parameters for "Straight horizontal line" element: X=5.67 abs.
6	 "Select dialog" "Accept dialog" "Accept element" "Accept"	Enter parameters for "Circular arc" element: Direction of rotation: clockwise R=72, X=5.67 abs., Y=0 abs.,

The following screenshot shows the programming contour:



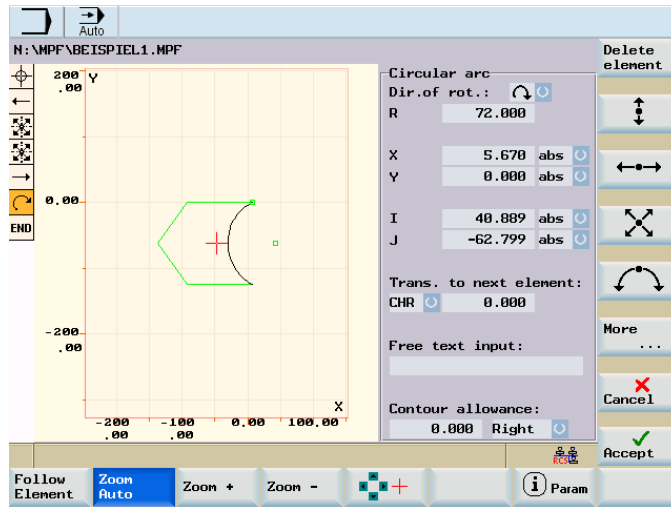


Figure 7-34 Example 1

## Example 2

Starting point: X=0 abs., Y=0 abs., machining plane G17

The contour is programmed in the clockwise direction with dialog selection.

For this contour it is advisable to display all parameters via the "All parameters" softkey.

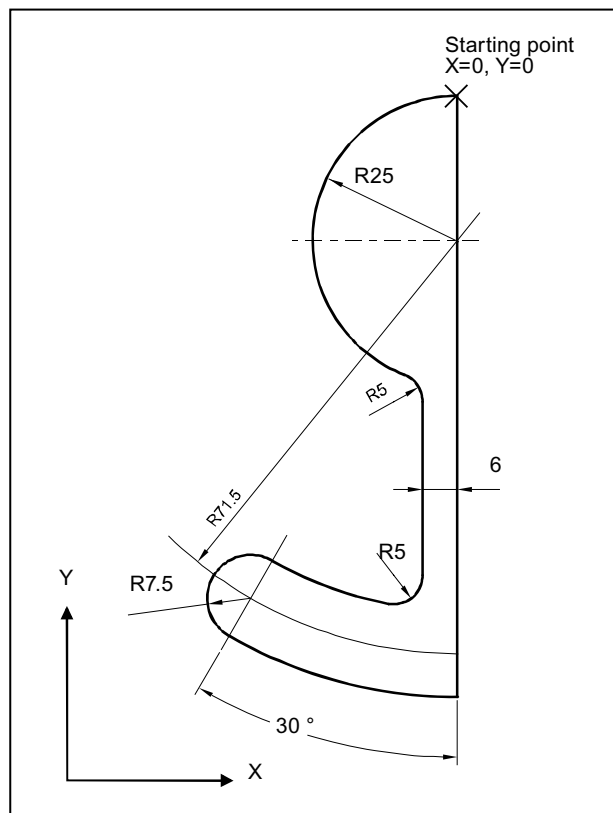








Figure 7-35 Workshop drawing of the contour, example 2

Table 7- 2 Example 2 operating sequences

Operating step	Softkey	Parameter
1		Y=-104 abs.
2		Clockwise direction of rotation, R=79, I=0 abs. Select dialog, all parameters, $\beta_2=30$ degrees
3		Clockwise direction of rotation, tangent to predecessor R=7.5, all parameters, $\beta_2=180$ degrees
4		Counter-clockwise direction of rotation, R=64, X=-6 abs., I=0 abs. Make dialog selection Transition to following element: R=5
5		All parameters, $\alpha_1=90$ degrees Transition to following element: R=5
6		Clockwise direction of rotation, R=25, X=0 abs., Y=0 abs., I=0 abs., make dialog selection

The following screenshot shows the programming contour:

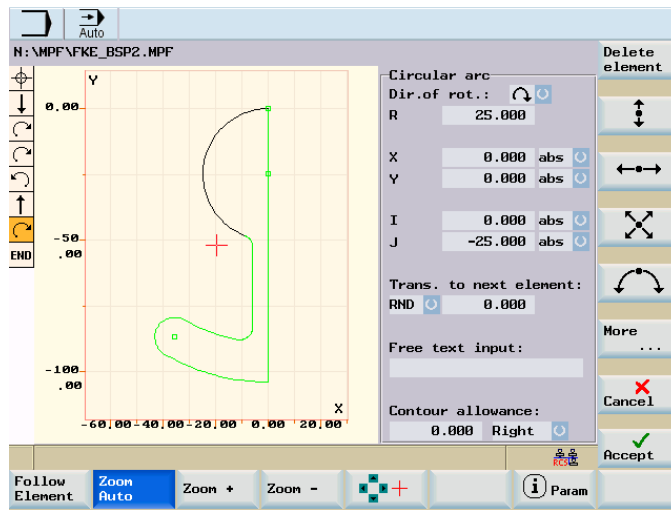


Figure 7-36 Example 2

**Example 3**

Starting point: X=0 abs., Y=5.7 abs., machining plane G17

The contour is programmed in a clockwise direction.

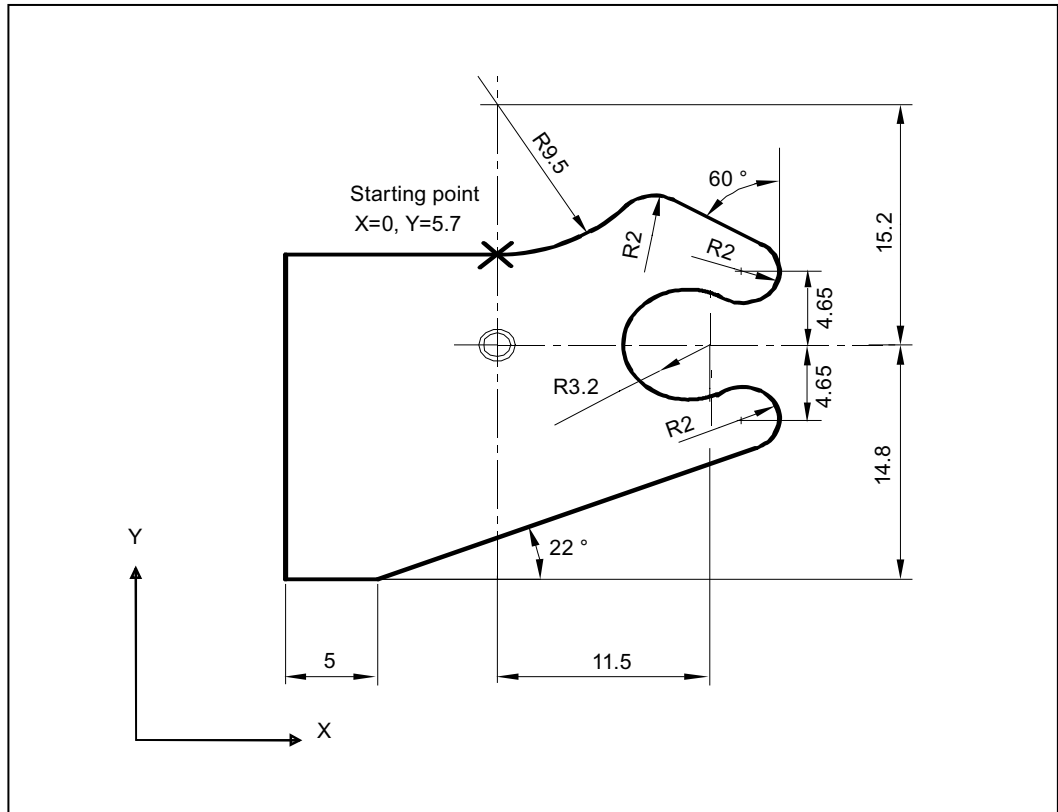


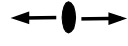

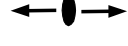


Figure 7-37 Workshop drawing of the contour, example 3

Table 7-3 Example 3 operating sequences

Operating step	Softkey	Parameter
1		Counter-clockwise direction of rotation, R=9.5, I=0 abs., make dialog selection Transition to following element: R=2
2		$\alpha 1 = -30$ degrees
3		Clockwise direction of rotation, tangent to predecessor R=2, J=4.65 abs.
4		Counter-clockwise direction of rotation, tangent to predecessor R=3.2, I=11.5 abs., J=0 abs., make dialog selection

Operating step	Softkey	Parameter
5		Clockwise direction of rotation, tangent to predecessor R=2, J=-4.65 abs., make dialog selection
6		Tangent to predecessor. $\alpha_1=-158$ degrees, Y=-14.8 abs., $\alpha_2=0$ degrees
7		All parameters, L=5, make dialog selection
8		Y=5.7 abs.
9		X=0 abs.

The following screenshot shows the programming contour:

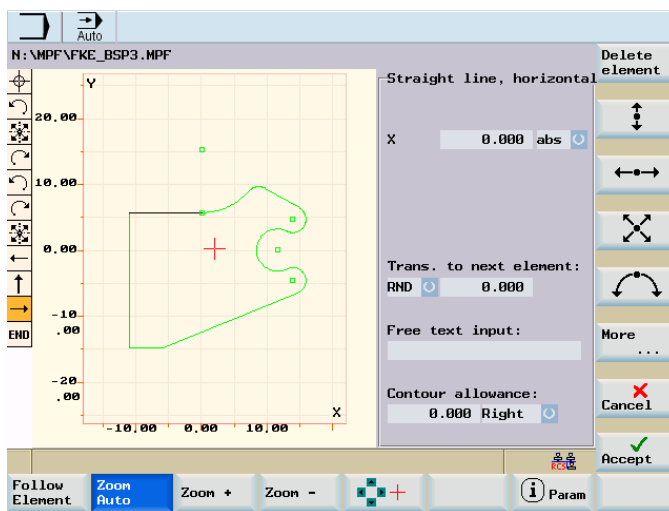


Figure 7-38 Example 3

## System

### 8.1 "System" operating area

#### Functionality

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

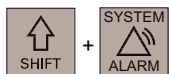
Depending on the functions selected, the horizontal and the vertical softkey bars change. The menu tree shown below **only** includes the horizontal softkeys.

#### 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



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

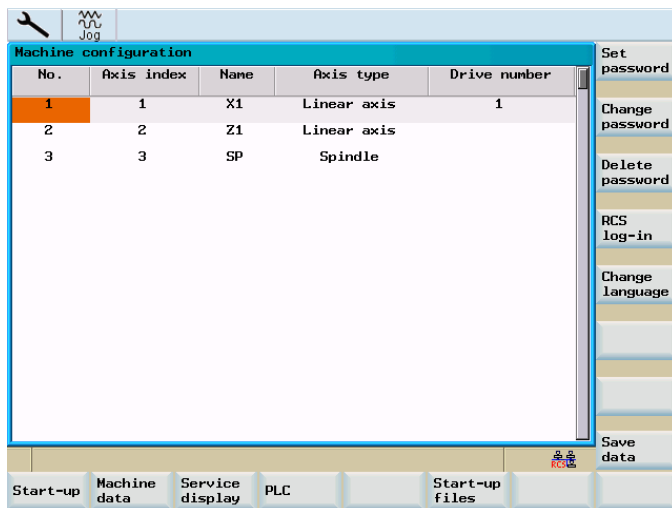
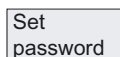


Figure 8-2 "System" operating area start screen

### Softkeys

The start screen vertical softkeys are described below.



#### "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".

---

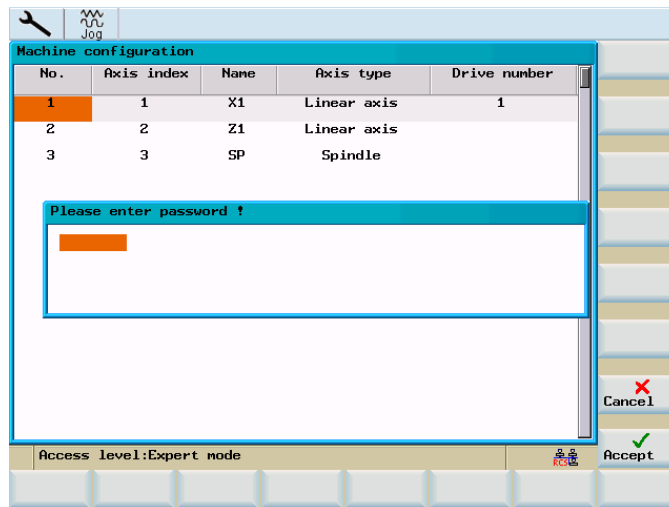


Figure 8-3 Entering the password

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"

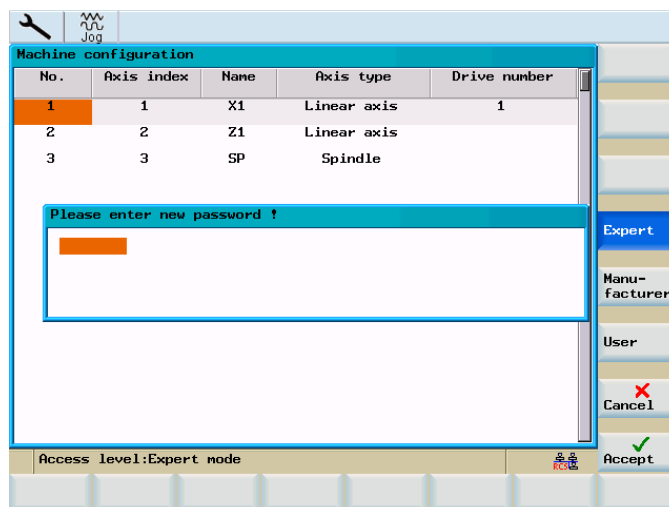


Figure 8-4 Change password

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.

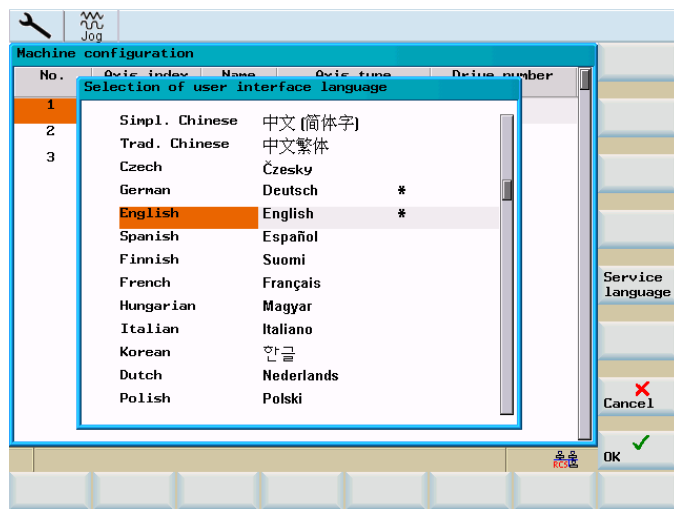


Figure 8-5 User interface language

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.

10000	REBOOT_DELAY_TIME	0.200000	s	so
1	2	3	4	5

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.

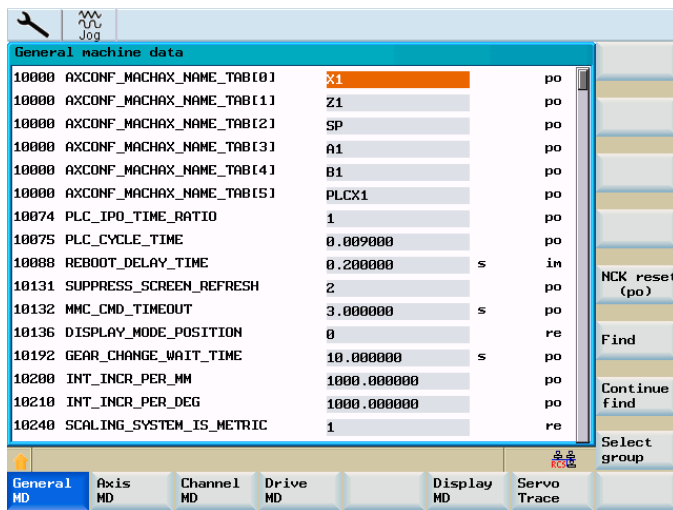


Figure 8-7 General machine data

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.

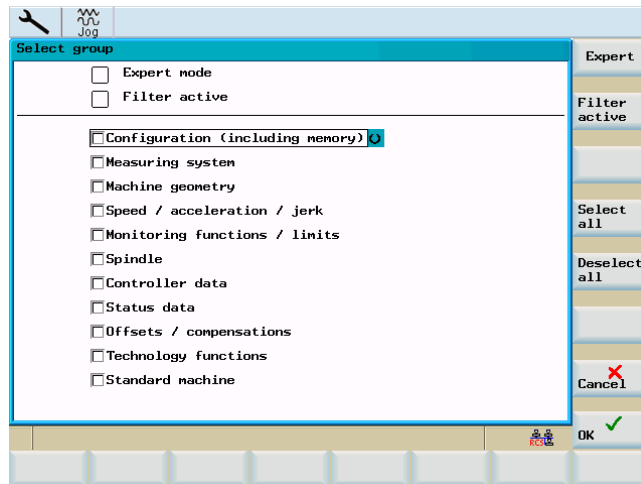


Figure 8-8 Display filter

## 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 -".

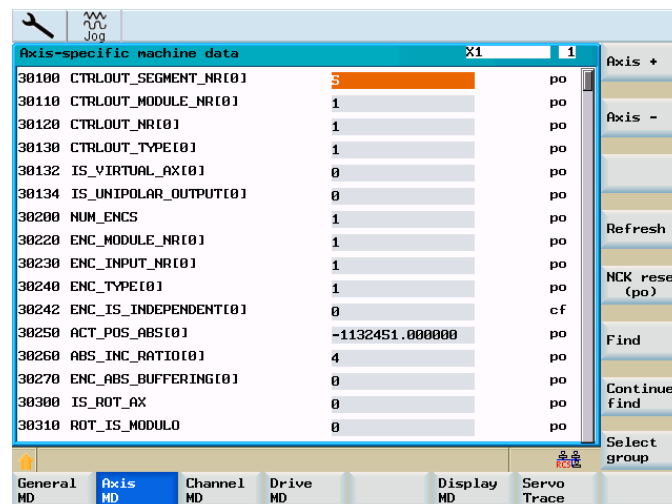


Figure 8-9 Axis-specific machine data

The data of axis 1 are displayed.

8.3 SYSTEM - "Machine data" softkeys

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.

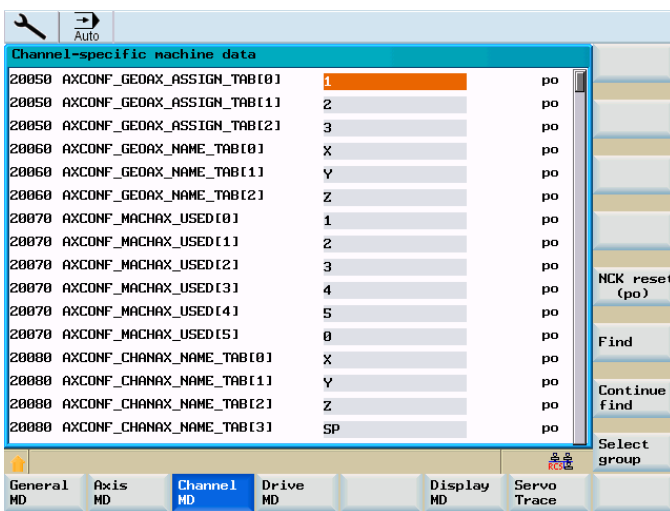


Figure 8-10 Channel-specific machine data

## 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.

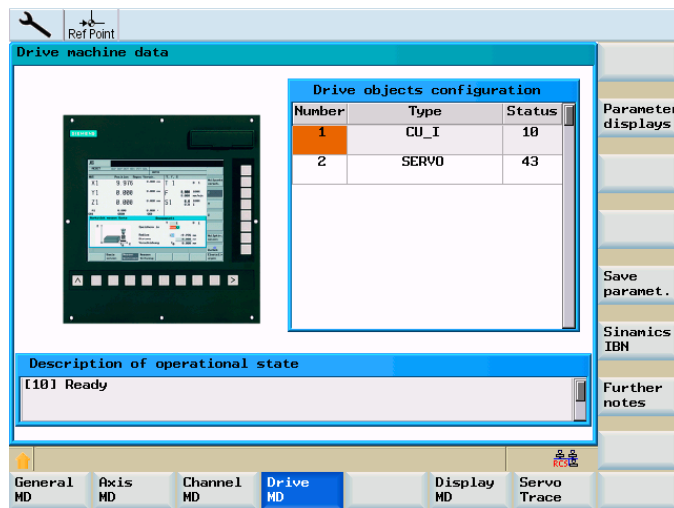


Figure 8-11 Drive machine data

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.

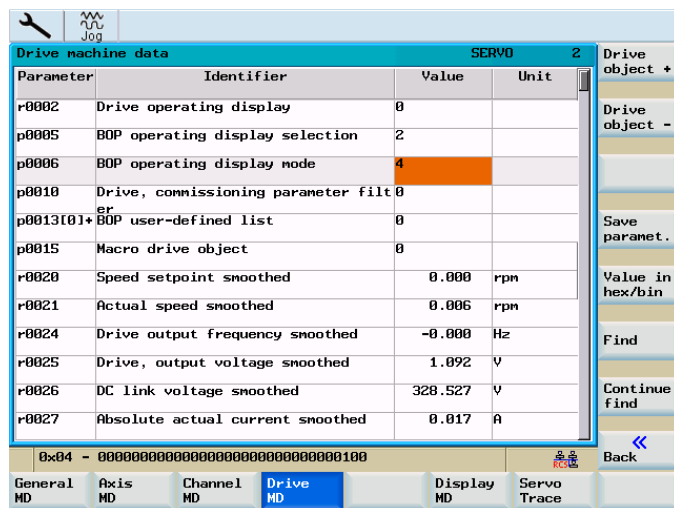


Figure 8-12 Parameter list

Drive  
object +

Switch to the respective drive objects.

Drive  
object -

**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.

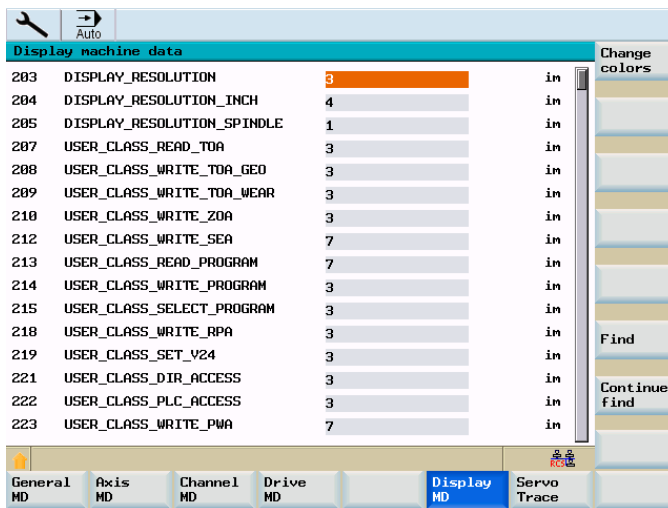


Figure 8-13 Display of machine data

**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.

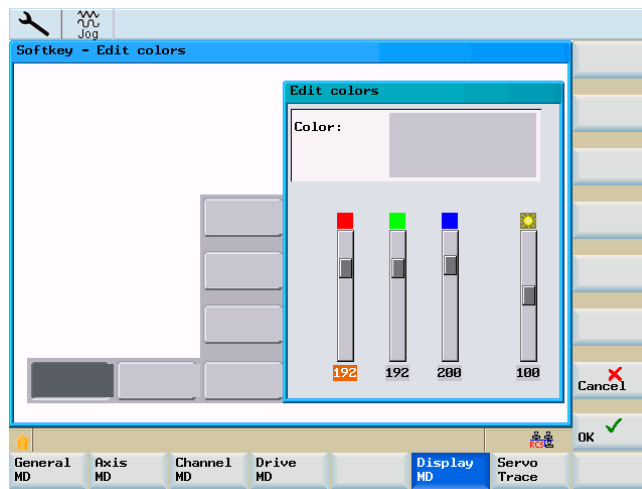


Figure 8-14 Edit softkey color.

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.

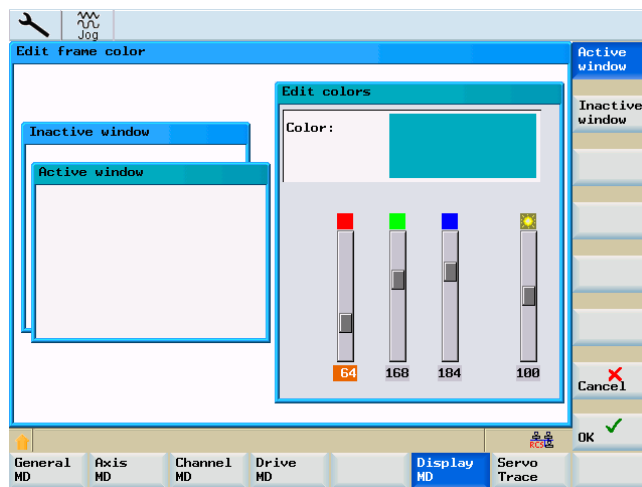


Figure 8-15 Edit frame color.

## 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.

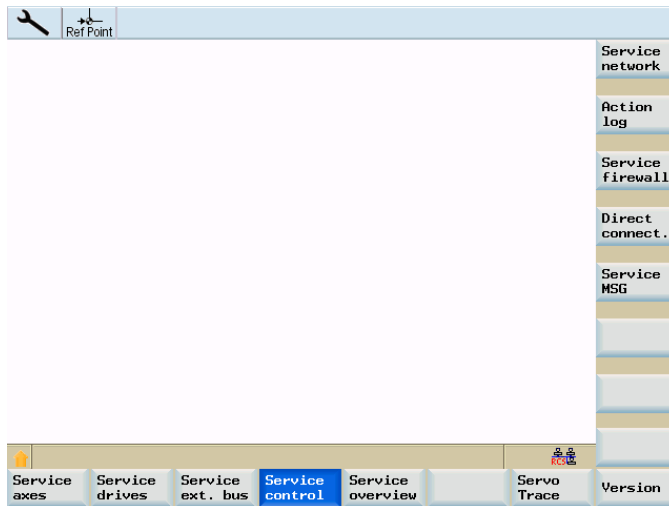


Figure 8-16 The "Service control" start screen

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.

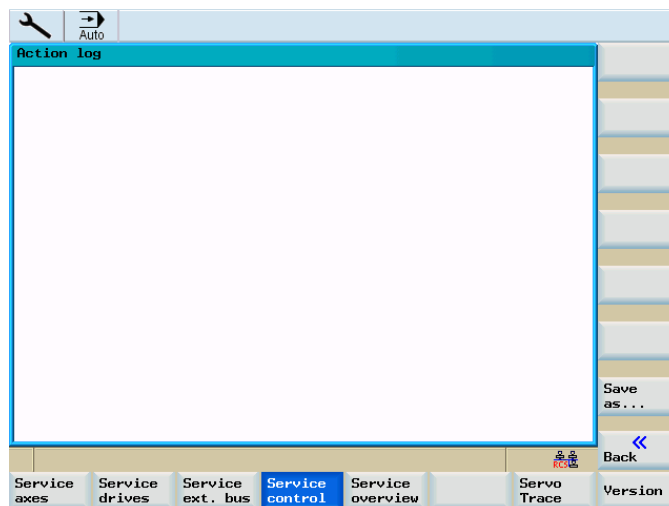


Figure 8-17 Action log

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.

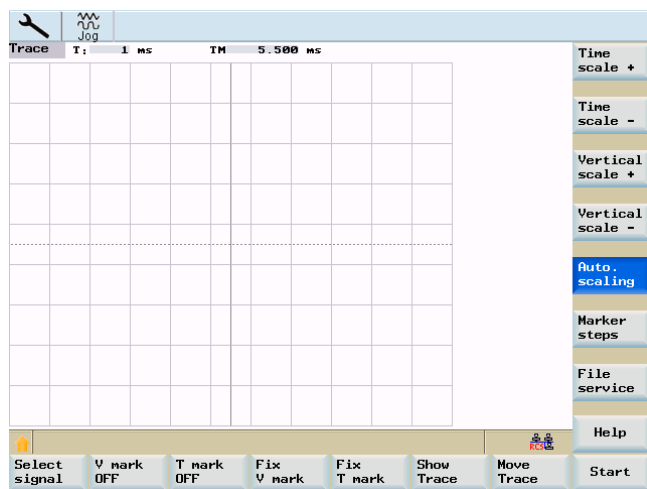
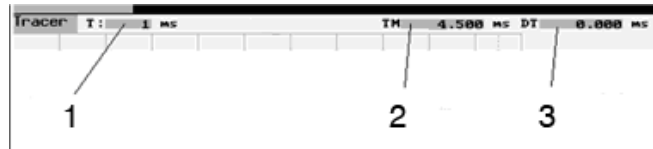


Figure 8-18 Servo trace start screen

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.



- 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.

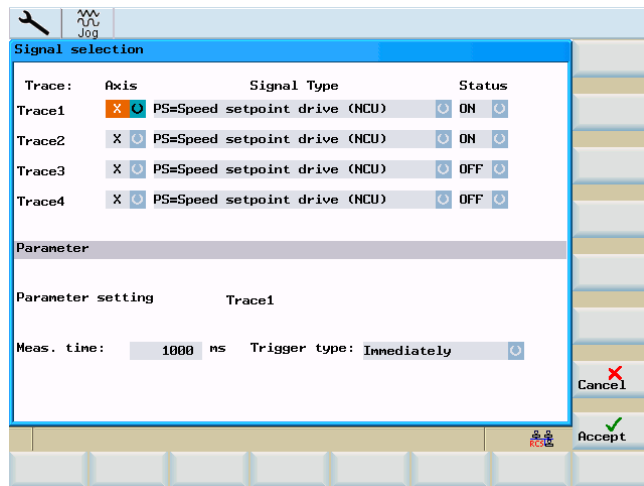


Figure 8-20 Select signal

- **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

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.

T mark off

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.

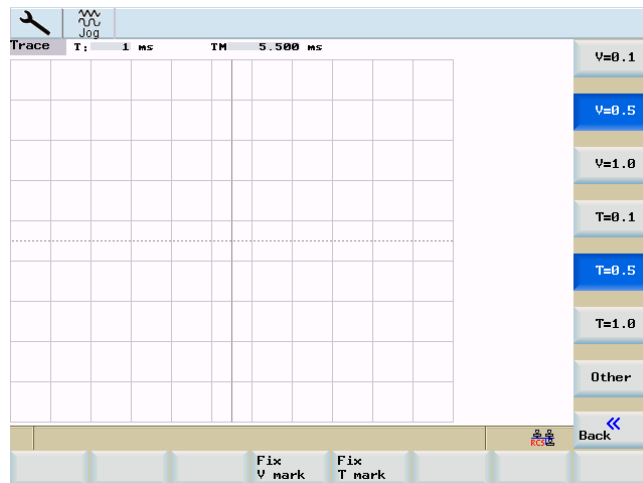


Figure 8-21 Marker steps

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.

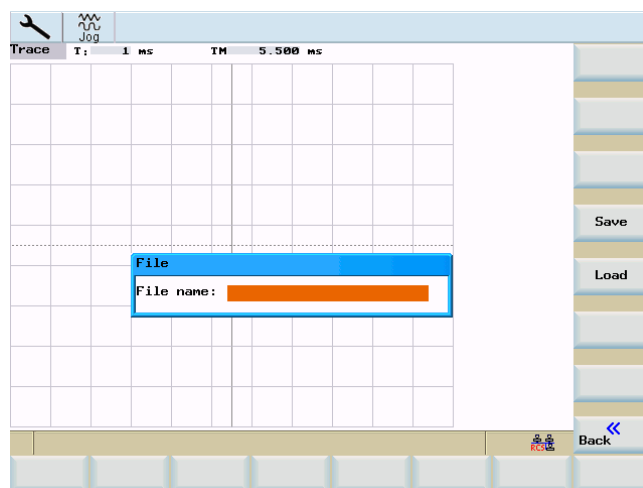


Figure 8-22 Trace data

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.

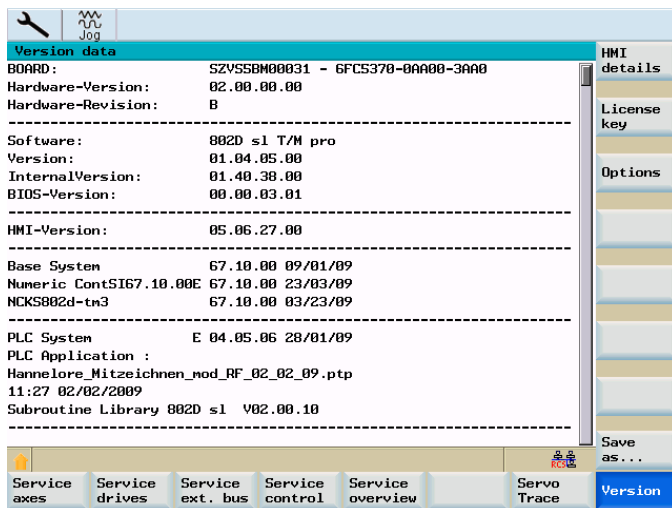


Figure 8-23 Version

#### Note

The version releases shown in the version screen shot are for example only.

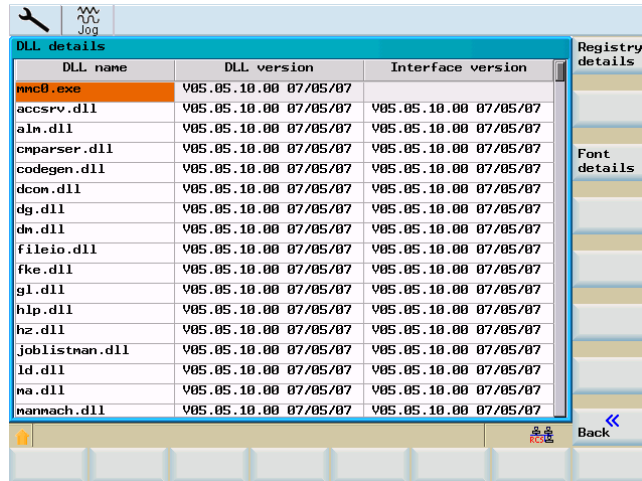
Save under

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.

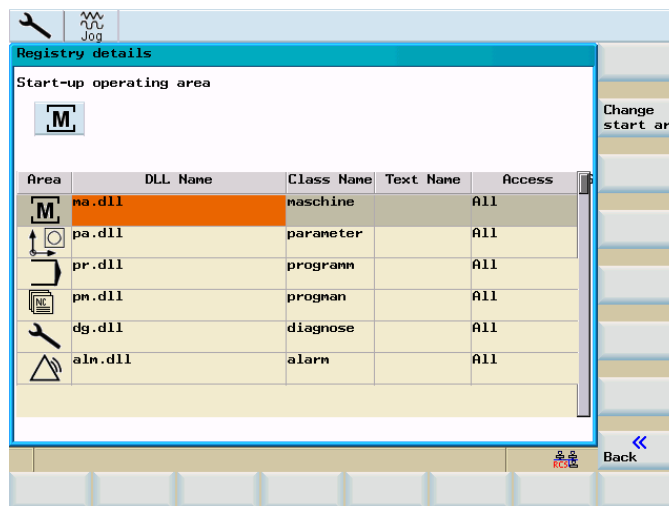


DLL name	DLL version	Interface version
mnc0.exe	V05.05.10.00 07/05/07	
accsrv.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
aln.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
cmparser.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
codegen.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
dcon.dll	V05.05.10.00 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
dn.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
na.dll	V05.05.10.00 07/05/07	V05.05.10.00 07/05/07
nannach.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.



Area	DLL Name	Class Name	Text Name	Access
M	na.dll	maschine		All
↑	pa.dll	parameter		All
→	pr.dll	programm		All
☰	pn.dll	progman		All
🔧	dg.dll	diagnose		All
🚨	aln.dll	alarm		All

Figure 8-25 Registry details



**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.

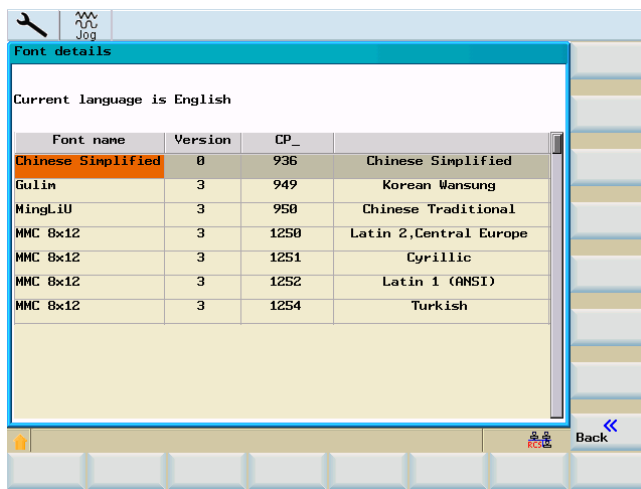


Figure 8-26 Font details

license  
key

Entering the license key.

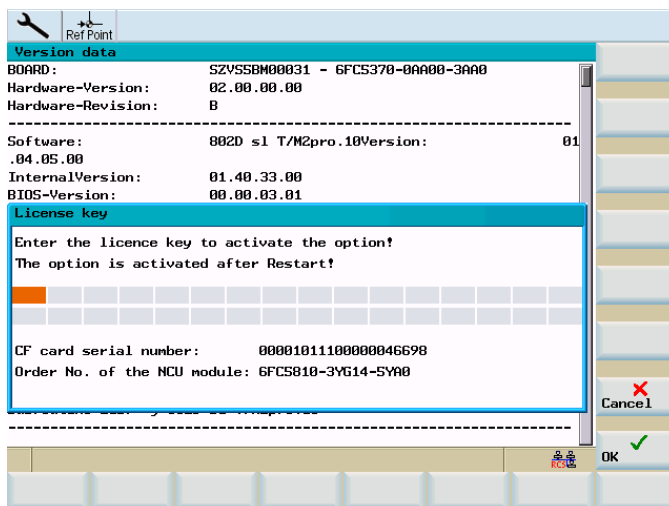


Figure 8-27 License key

## References

SINUMERIK 802D sl Operating Instructions for Turning, Milling, Grinding, Nibbling; Licensing in SINUMERIK 802D sl

Options

Setting the licensed options.

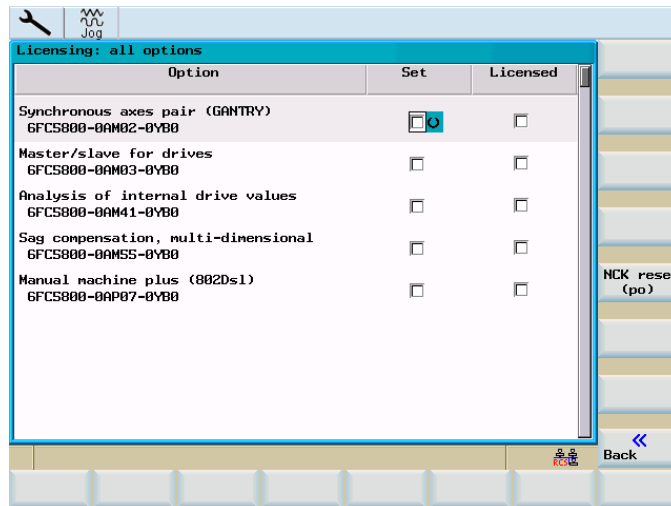


Figure 8-28 Options

## 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 (" <b>&lt;interface&gt;</b> : 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".

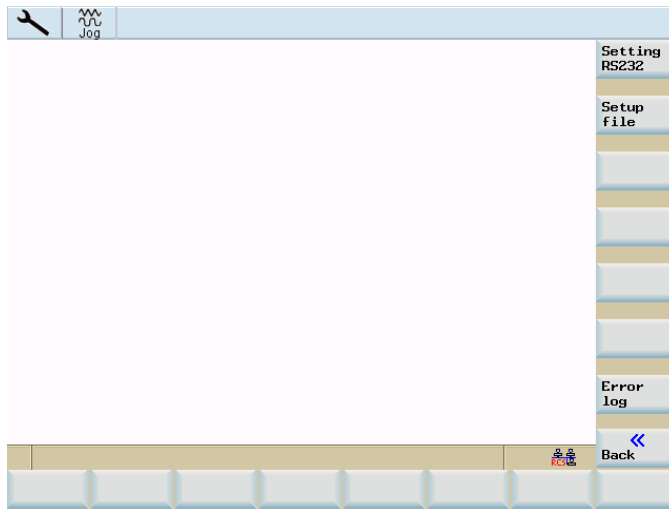


Figure 8-29 Dialog box, Service MSG

## Settings for output via the RS232 interface

Setting  
RS232

Settings of the RS232 output interface.

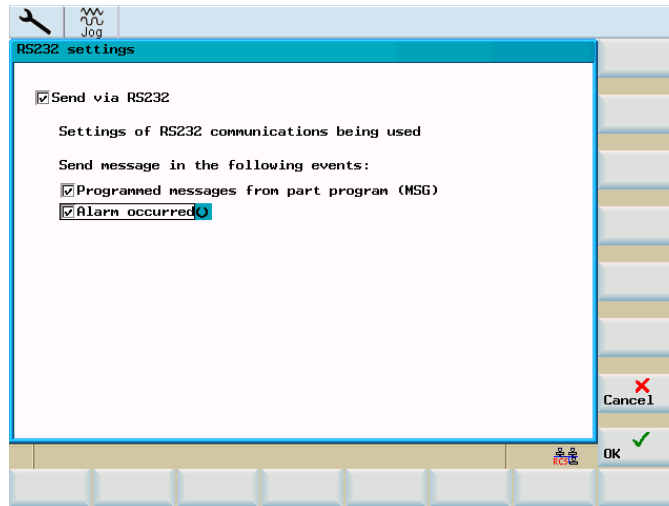


Figure 8-30 Dialog box, RS232 interface settings

"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.

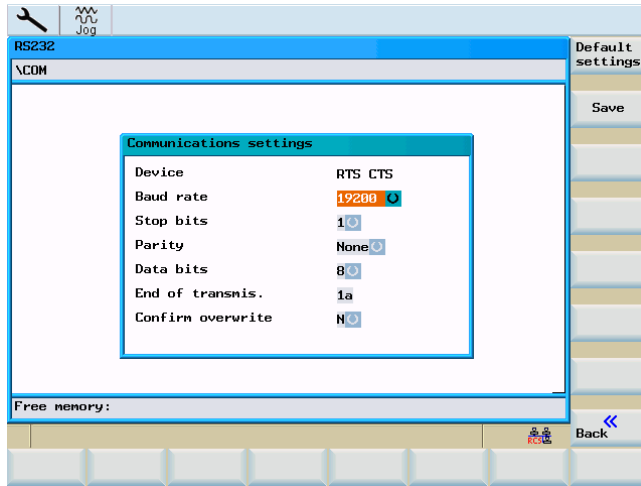


Figure 8-31 Parameters of the RS232 interface

---

**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.

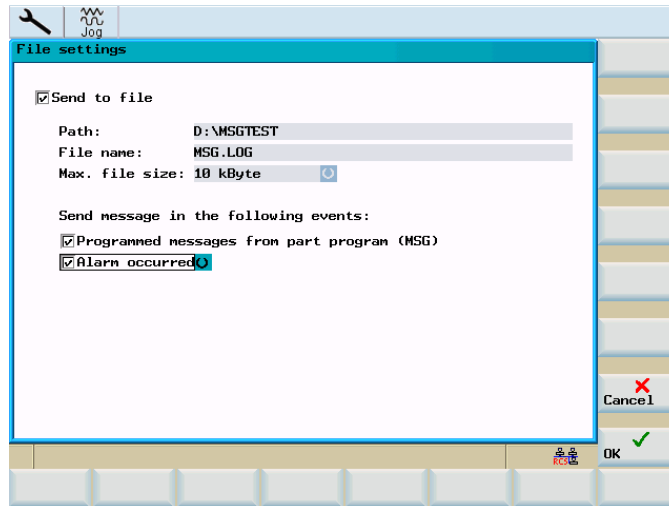


Figure 8-32 Dialog box, file settings

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.

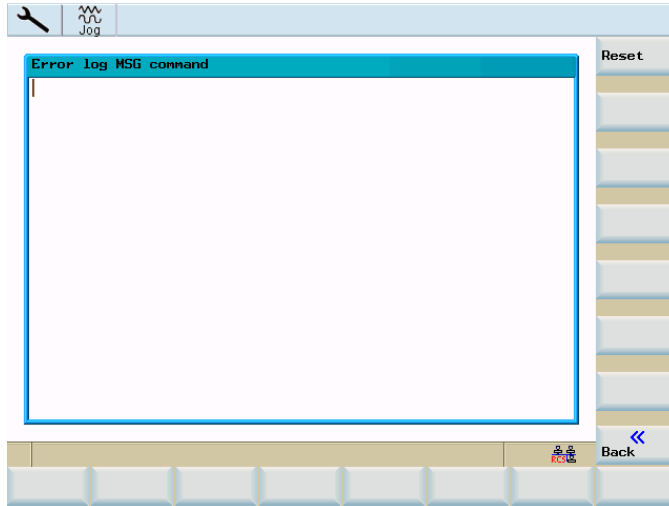


Figure 8-33 Dialog box, error log

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... Z ...	
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... Z ...	
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.

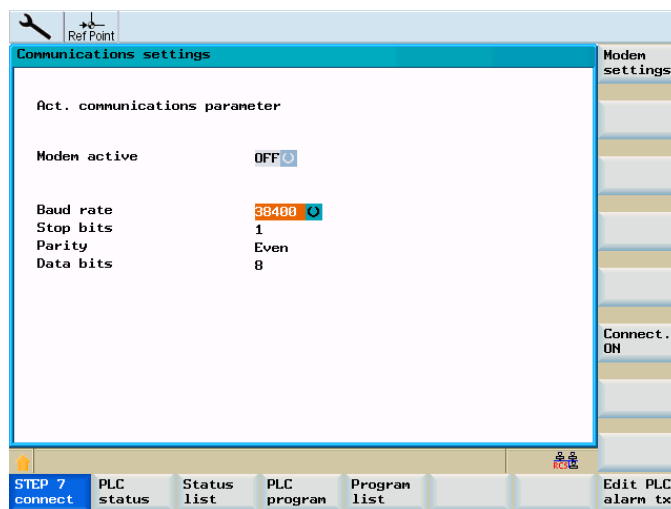


Figure 8-34 Communication settings

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:

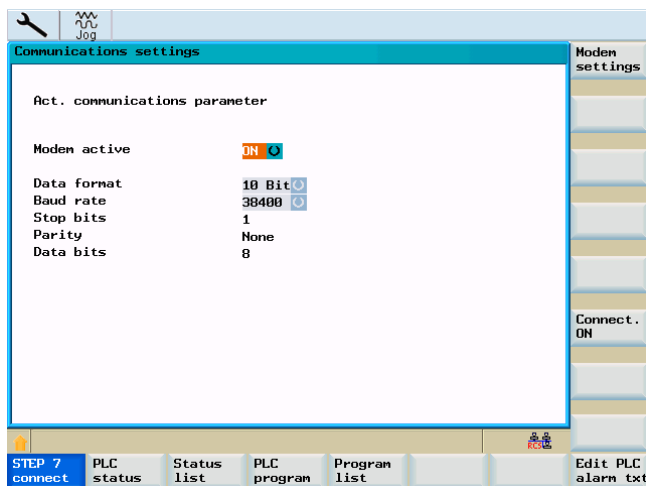


Figure 8-35 Initialize the modem

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:

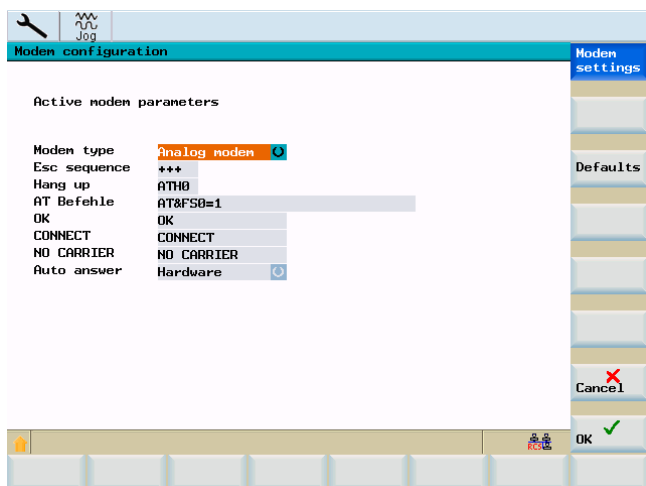


Figure 8-36 Modem settings

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.



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

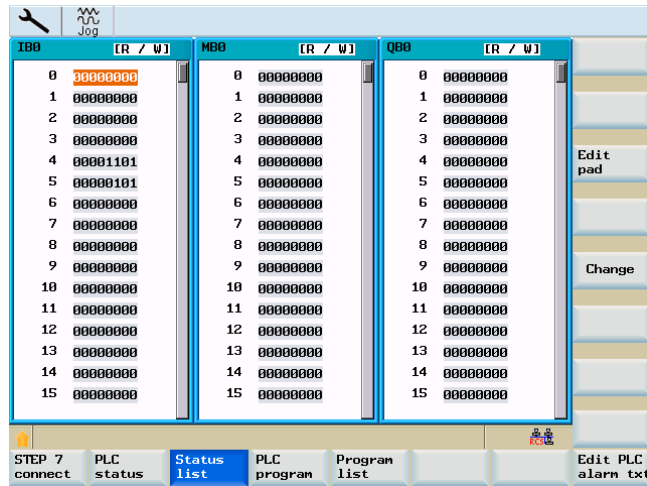


Figure 8-38 PLC status list

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.



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.

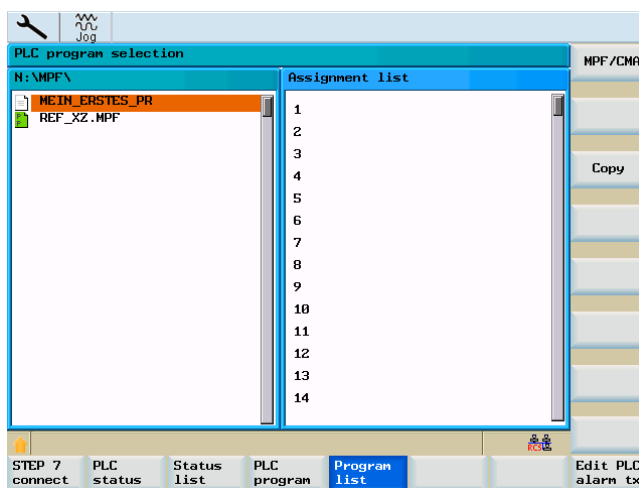


Figure 8-40 PLC program list



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.

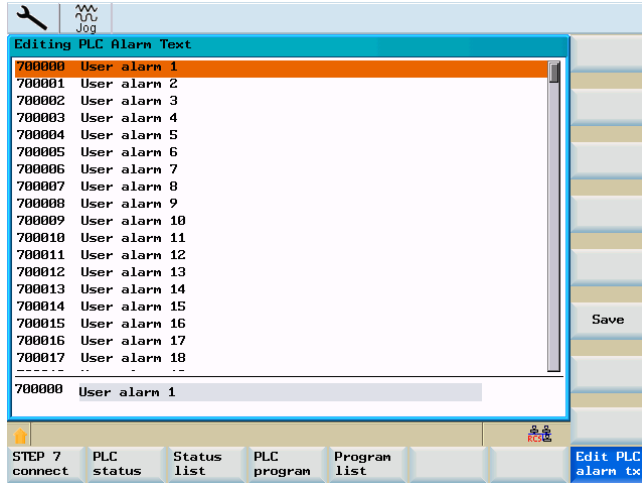


Figure 8-41 Editing the PLC alarm text

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.

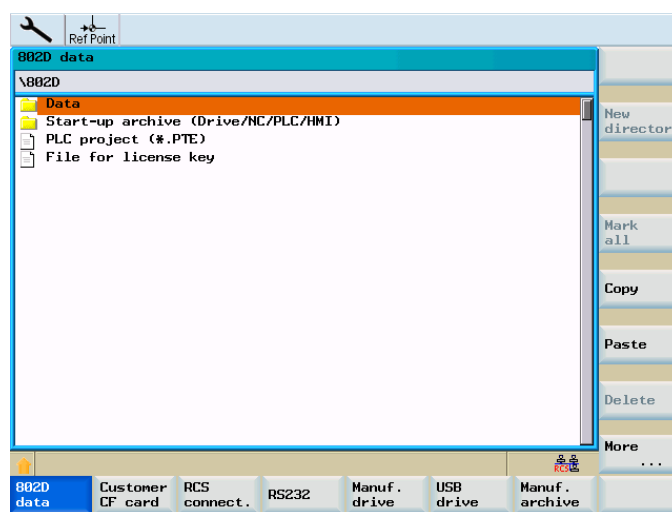


Figure 8-42 Start-up files

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.

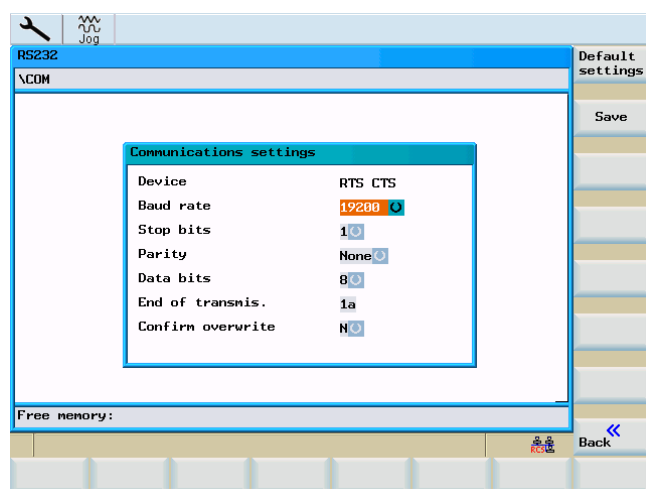


Figure 8-43 Parameters of the RS232 interface

Interface parameters

Table 8- 8 Interface parameters

Parameter	Description
Device type	<b>RTS CTS</b> 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.

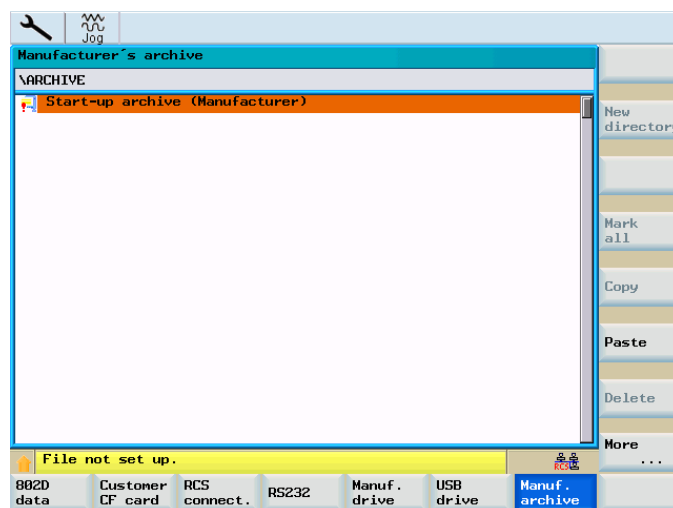


Figure 8-44 Manufacturers` archive, archive file not yet created

## 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



The alarm window is opened. You can sort the NC alarms using softkeys; PLC alarms will **not** be sorted.

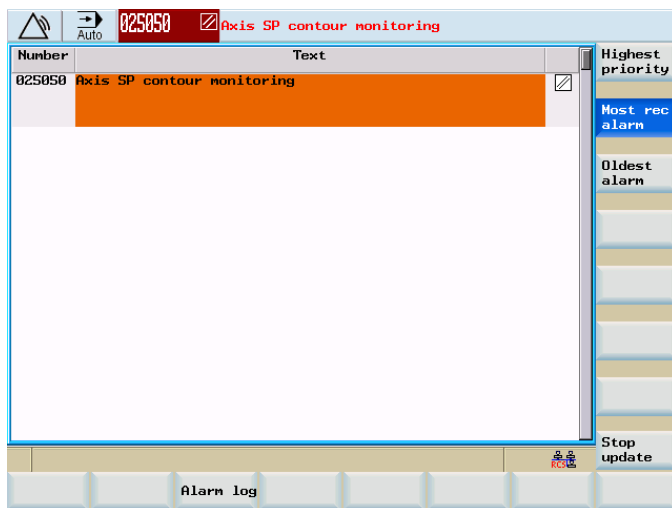
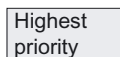
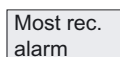


Figure 8-45 Alarm display window

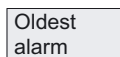
### Softkeys



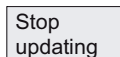
Use this softkey to display all alarms sorted by their priorities. The highest priority alarm is at the beginning of the list.



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.



Use this softkey to display the alarms sorted by the time of their occurrence. The oldest alarm stands at the beginning of the list.



Updating of pending alarms is stopped / started.



Alarm log

All alarms are logged.

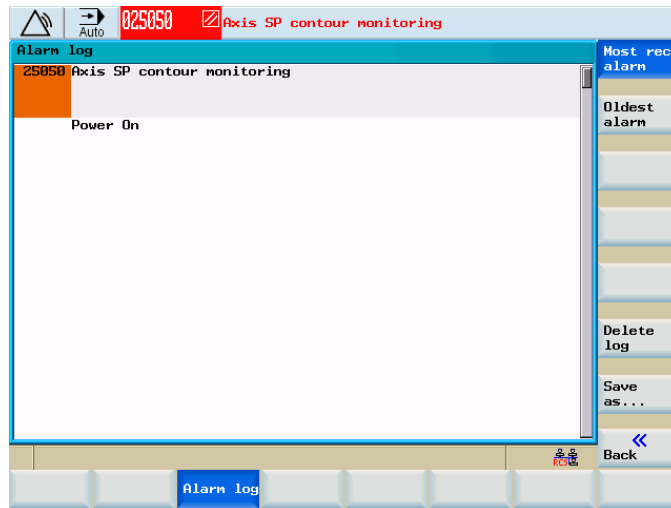


Figure 8-46 Alarm log

Save under

The log is deleted using softkey "Delete log".

The file is output using softkey "Save under..." on a CF card or on the USB FlashDrive.



## Programming

### 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: **WORKPIECE**

#### 9.1.2 Program structure

##### Structure and content

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 characters:** generally a letter and a
- **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.

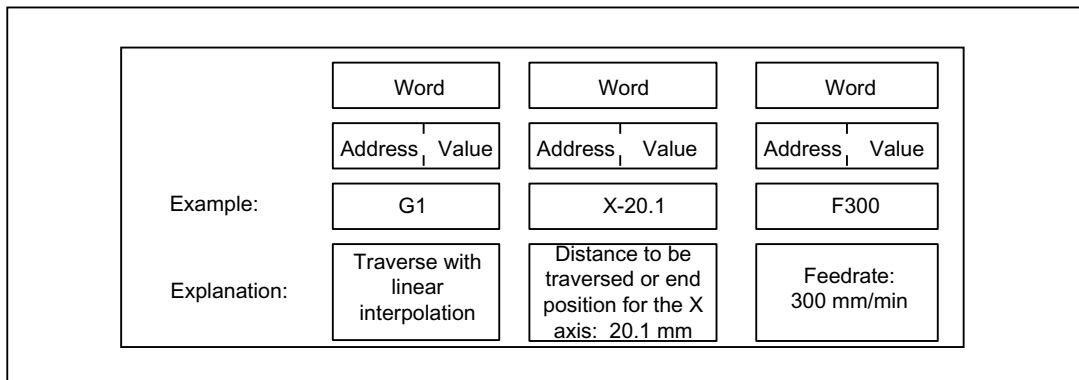


Figure 9-1 Word structure (example)

#### 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 "List of instructions").

Example: **SCALE** ; Enable scaling factor

#### Extended address

For the following addresses, the address is extended by 1 to 4 digits to obtain a higher number of addresses.

R: Arithmetic parameters

H: H function

I, J, K: Interpolation parameters/intermediate point

In this case, the value must be assigned using an equality sign "=" (see also Section "List of instructions").

Example: **R10=6.234 H5=12.1 I1=32.67**

## 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.

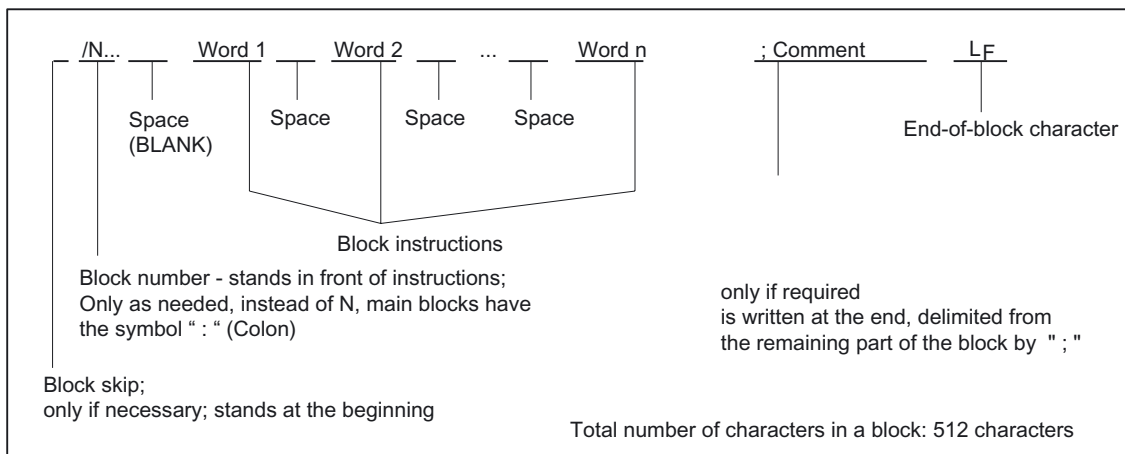


Figure 9-2 Block structure diagram

### 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 Fonts

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 upper and lower case 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

Tabulator: Reserved; do not use

9.1.6 List of instructions

Functions marked with \*\* are not available on SINUMERIK 802D sl value.

The functions marked with \* take effect at the start of the program (control version for "Milling" technology, if not otherwise programmed and provided the default settings have been retained by the machine manufacturer).

Address	Significance	Value assignments	Information	Program- ming
D	Tool offset number	0 ... 9, only integer, no sign	Contains compensation data for a particular tool T... ; D0->compensation values= 0, maximum 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 in block with G4	0.001 ... 99 999.999	Dwell time in seconds	G4 F...; separate block
G	G function (preparatory function)	Only integer, specified values	The G functions are divided into G groups. Only one G function from one group can be written in one block. A G function can either be modal (until canceled by another function from the same group), or non-modal (only effective for the block it is written in).  <b>G group:</b>	G... or symbolic name, e.g.: CIP
G0	Linear interpolation at rapid traverse rate		1: Motion commands (type of interpolation) modally effective	G0 X... Y... Z... ; Cartesian in polar coordinates: G0 AP=... RP=... or with additional axis: G0 AP=... RP=... Z... ; e.g.: with G17 axis Z
G1 *	Linear interpolation at feedrate			G1 X... Y... Z... F... in polar coordinates: G1 AP=... RP=... F... or with additional axis: G1 AP=... RP=... Z... F... ; e.g.: with G17 axis Z
G2	Circular interpolation in clockwise direction (in conjunction with a third axis and TURN=... also helix interpolation -> see under TURN)			G2 X... Y... I... J... F... ; Center and end points G2 X... Y... CR=... F... ; Radius and end point G2 AR=... I... J... F... ; Aperture angle and center point G2 AR=... X... Y... F... ; Aperture angle and end point in polar coordinates: G2 AP=... RP=... F... or with additional axis: G2 AP=... RP=... Z... F... ; e.g.: with G17 axis Z



Address	Significance	Value assignments	Information	Program- ming
G3	Circular interpolation in counter-clockwise direction  (in conjunction with a third axis and TURN=... also helix interpolation -> see under TURN)			G3 ... ; otherwise as for G2
CIP	Circular interpolation through intermediate point			CIP X... Y... Z... I1=... J1=... K1=... F...
CT	Circular interpolation; tangential transition			N10 ... N20 CT X... Y... F... ; Circle; tangential transition to the previous path segment
G33	Thread cutting, tapping with constant pitch			S... M... ;Spindle speed, direction G33 Z... K... ;Thread drilling <b>with</b> compensating chuck, e.g. in Z axis
G331	Thread interpolation			N10 SPOS=... ;Spindle in position control N20 G331 Z... K... S... ;tapping <b>without</b> compensating chuck, e.g. in Z axis ; RH or LH thread is defined via the sign of the pitch (e.g. K+): + : as for M3 - : as with M4
G332	Thread interpolation - retraction			G332 Z... K... ; <b>Rigid tapping</b> , e.g. in Z axis, <b>retraction motion</b> ; sign of pitch as for G331
G4	Dwell time		2: Special motions, non-modal	G4 F...;separate block, F: Time in seconds or G4 S.... ;separate block, S: in spindle revolutions
G63	Tapping with compensating chuck			G63 Z... F... S... M...
G74	Reference point approach			G74 X1=0 Y1=0 Z1=0; Separate block, (machine axis identifier!)
G75	Fixed point approach			G75 X1=0 Y1=0 Z1=0; Separate block, (machine axis identifier!)
G147	SAR - Approach with a straight line			G147 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G148	SAR - Retract with a straight line			G148 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...

Address	Significance	Value assignments	Information	Program- ming
G247	SAR - Approach with a quadrant			G247 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G248	SAR - Retract with a quadrant			G248 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
G347	SAR - Approach with a semicircle			G347 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G348	SAR - Retract with a semicircle			G348 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
TRANS	Translation, programmable		3: Write memory non-modal	TRANS X... Y... Z... ; separate block
ROT	Rotation, programmable			ROT RPL=... ; Rotation in the current plane G17 to G19, separate block
SCALE	Programmable scaling factor			SCALE X... Y... Z... ; Scaling factor in the direction of the specified axis, separate block
MIRROR	Programmable mirroring			MIRROR X0 ; Coordinate axis whose direction is changed, separate block
ATRANS	Additive translation, programming			ATRANS X... Y... Z... ; separate block
AROT	Additive programmable rotation			AROT RPL=... ; Rotation in the current plane G17 to G19, separate block
ASCALE	Additive programmable scaling factor			ASCALE X... Y... Z... ; Scaling factor in the direction of the specified axis, 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... Y... Z... ; separate block
G26	Upper spindle speed limitation or upper working area limitation		G26 S... ; separate block G26 X... Y... Z... ; separate block	
G110	Pole specification relative to the last programmed setpoint position		G110 X... Y... ; Pole specification, Cartesian, e.g.: for G17 G110 RP=... AP=... ; Pole specification, polar, separate block	

Address	Significance	Value assignments	Information	Program- ming
G111	Pole specification relative to origin of current workpiece coordinate system			G111 X... Y... ; Pole specification, Cartesian, e.g.: for G17 G111 RP=... AP=... ; Pole specification, polar, separate block
G112	Pole specification, relative to the last valid POLE			G112 X... Y... ; Pole specification, Cartesian, e.g.: for G17 G112 RP=... AP=... ; Pole specification, polar, separate block
G17 *	X/Y plane		6: Plane selection modally effective	G17 .... ; Vertical axis on this
G18	Z/X plane			Plane is tool length
G19	Y/Z plane			Compensation axis
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			
G53	Non-modal skipping of the settable work offset		9: Suppression of settable work offset non-modal	
G153	Non-modal skipping of the settable work offset including base frame			
G60 *	Exact stop		10: Approach behavior modally effective	
G64	Continuous-path mode			
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: Exact stop, non-modal non-modal	
G601 *	Exact stop window, fine, with G60, G9		12: Exact stop window modally effective	
G602	Exact stop window, coarse, with G60, G9			
G621	Corner deceleration at all corners.		Only in conjunction with continuous-path mode.	G621 AIDS=...
G70	Inch dimension data input		13: Inch / metr.dimension input	
G71 *	Metric dimension data input		modally effective	

9.1 Fundamental Principles of NC Programming

Address	Significance	Value assignments	Information	Program- ming
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 data input		modally effective	
G94 *	Feed F in mm/min		15: Feedrate / spindle	
G95	Feedrate F in mm/spindle revolutions		modally effective	
CFC *	Feedrate override with circle ON		16: Feed override	
CFTCP	Feedrate override OFF		modally effective	
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	; Applies to all axes activated via setting data; values set via G25, G26
WALIMOF *	Working area limitation OFF		modally effective	
COMPOF *	Compressor OFF		30: Compressor	
COMPCAD	Compressor for surface quality ON		modally effective Functions available only with SINUMERIK 802D sl pro.	
EXTCALL	Execute external subprogram			Reload program from HMI in "Execution from external source" mode.
G340 *	Approach and retraction in space (SAR)		44: Path segmentation with SAR	
G341	Approach and retraction in the plane (SAR)		modally effective	
G290 *	SIEMENS mode		47: External NC languages	
G291	External mode		modally effective	
The functions marked with * act at program start (CNC variant for "Milling" technology unless otherwise programmed and provided that the machine manufacturer default settings have not been changed).				
H H0= to H9999=	H function	± 0.0000001 ... 9999 9999 (8 decimal places) or with specification of an exponent: ± (10 <sup>-300</sup> ... 10 <sup>+300</sup> )	Value transfer to the PLC; meaning defined by the machine manufacturer.	H0=... H9999=... e.g.: H7=23.456

Address	Significance	Value assignments	Information	Program- ming
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, G331, G332 -> thread pitch	See G2, G3, G33, G331, and G332
J	Interpolation parameters	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	Belonging to the Y axis; otherwise, as with I	See G2, G3, G33, G331, and G332
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, G33, G331, and G332
I1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the X axis; specification for circular interpolation with CIP	See CIP
J1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belonging to the Y 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.	L781; 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	
M3	Spindle clockwise			
M4	Spindle counter-clockwise			

Address	Significance	Value assignments	Information	Program- ming
M5	Spindle stop			
M6	Tool change		Only if activated with M6 via the machine control panel; otherwise, change directly using the T command.	
M40	Automatic gear stage changeover			
M41 to M45	Gear stage 1 to gear stage 5			
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 in 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.	N10 L781 P... ; separate block N10 L871 P3; three cycles
R0 to R299	Arithmetic parameters	$\pm 0.0000001 \dots 9999 9999$ (8 decimal places) or with specification of an exponent: $\pm (10^{-300} \dots 10^{+300})$		R1=7.9431 R2=4  with specification of an exponent: R1=-1.9876EX9; R1=-1 987 600 000
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: ... Degr.
ATAN2( , )	Arctangent2		The angle of the sum vector is calculated from two vectors standing vertically one on another. The second 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

Address	Significance	Value assignments	Information	Program- ming
SQRT()	Square root			R6=SQRT(R7)
POT()	Square			R12=POT(R13)
ABS()	Absolute value			R8=ABS(R9)
TRUNC()	Truncate to integer			R10=TRUNC(R11)
LN()	Natural logarithm			R12=LN(R9)
EXP()	Exponential function			R13=EXP(R1)
RET	Subroutine end		Used instead of M2 - to maintain the continuous-path control mode	RET ;separate block
S	Spindle speed	0.001 ... 99 999.999	Unit of measurement of the spindle speed rpm.	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	Y...
Z	Axis	±0.001 ... 99 999.999	Positional data	Z...
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[ <i>axis</i> ]	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

Address	Significance	Value assignments	Information	Program- ming
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 G17 X... Y... N11 X... <b>ANG=...</b> or contour over several blocks: N10 G1 G17 X... Y... N11 <b>ANG=...</b> N12 X... Y... <b>ANG=...</b>
AP	Polar angle	0 ... ±359.99999	Specification in degrees, traversing in polar coordinates, definition of the pole; in addition: Polar radius RP	see G0, G1, G2; G3 G110, G111, G112
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 <b>chamfer length</b> between two contour blocks	N10 X... Y... <b>CHF=...</b> N11 X... Y...
CHR	Chamfer; in the contour definition	0.001 ... 99 999.999	Inserts a chamfer of the specified <b>side length</b> between two contour blocks	N10 X... Y... <b>CHR=...</b> N11 X... Y...
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
CYCLE... HOLES... POCKET.. SLOT...	Machining cycle	Only specified values	Call of machining cycles requires a separate block, the provided transfer parameters must be assigned values, special cycle calls are possible with additional MCALL or CALL	
CYCLE81	Drilling, centering			N5 RTP=110 RFP=100 ... ; Assign with values N10 CYCLE81(RTP, RFP, ...); separate 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 <b>with</b> compensating chuck			N10 CYCLE840(...); separate block



Address	Significance	Value assignments	Information	Program- ming
CYCLE85	Reaming 1			N10 CYCLE85(...); separate block
CYCLE86	Boring			N10 CYCLE86(...); separate block
CYCLE87	Drilling with stop 1			N10 CYCLE87(...); separate block
CYCLE88	Drilling with stop 2			N10 CYCLE88(...); separate block
CYCLE89	Reaming 2			N10 CYCLE89(...); separate block
CYCLE90	Thread milling			N10 CYCLE90(...); separate block
HOLES1	Row of holes			N10 HOLES1(...); separate block
HOLES2	Circle of holes			N10 HOLES2(...); separate block
SLOT1	Mill slot			N10 SLOT1(...); separate block
SLOT2	Mill a circumferential slot			N10 SLOT2(...); separate block
POCKET3	Rectangular pocket			N10 POCKET3(...); separate block
POCKET4	Circular pocket			N10 POCKET4(...); separate block
CYCLE71	Face milling			N10 CYCLE71(...); separate block
CYCLE72	Contour milling			N10 CYCLE72(...); separate block
CYCLE76	Milling the rectangular spigot			N10 CYCLE76(...) ;separate block
CYCLE77	Circular spigot milling			N10 CYCLE77(...) ;separate block
LONG- HOLE	Elongated hole			N10 LONGHOLE(...); separate 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 user variable of the BOOL, CHAR, INT, REAL, STRING[n] type; define directly at the beginning of the program	DEF INT VARI1=24, VARI2; two variables of the INT type; name defined by the user DEF STRING[12] VARS3="HELLO" ; max. 12 characters

Address	Significance	Value assignments	Information	Program- ming
DISCL	Approach / retraction distance of infeed motion for machining plane (SAR)	-	Safety clearance for speed switchover for infeed motion; note: G340, G341	See G147, G148 , G247, G248 , G347, G348
DISR	Approach / retraction distance or radius (SAR)	-	G147/G148: Distance of the cutter edge from the starting or end point of the contour G247, G347/G248, G348: Radius of the tool center point path	See G147, G148 , G247, G248 , G347, G348
FAD	Velocity for the infeed (SAR)	-	Speed takes effect after the safety clearance is reached for the infeed; note: G340, G341	See G147, G148 , G247, G248 , G347, G348
FRC	Non-modal feedrate for chamfer/rounding	0, >0	In case 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	If FRCM=0, feedrate F will go into effect,	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
IF	Jump condition	-	If the jump condition is fulfilled, the jump to the block with <i>Label: is performed</i> ; otherwise, next instruction/block; several IF instructions per block are possible  <b>Relational operators:</b> = = 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: ...

Address	Significance	Value assignments	Information	Program- ming
MEAS**	Measuring <b>with</b> deletion of the distance to go	+1 -1	=+1: Measuring input 1, rising edge =-1: Measuring input 1, falling edge	N10 <b>MEAS=-1</b> G1 X... Y... Z...F...
MEAW**	Measuring <b>without</b> deletion of the distance to go	+1 -1	=+1: Measuring input 1, rising edge =-1: Measuring input 1, falling edge	N10 <b>MEAW=-1</b> G1 X... Y... 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_MONIF ACT**	Factor for tool life monitoring	> 0.0	Initialization value: 1.0	N10 \$A_MONIFACT=5.0; Tool life elapsed five 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 <b>machine coordinate system</b>	-	Axis: Identifier of an axis (X, Y, Z ...) traversing when measuring	N10 R1=\$AA_MM[X]
\$AA_MW [axis]**	Measurement result for an axis in the <b>workpiece coordinate system</b>	-	Axis: Identifier of an axis (X, Y, Z ...) traversing when measuring	N10 R2=\$AA_MW[X]
\$A..... TIME **	Timer for runtime: \$AN_SETUP_ TIME \$AN_POWERON_ TIME \$AC_OPERATING_ TIME \$AC_CYCLE_ TIME \$AC_CUTTING_ TIME	0.0 ... 10 <sup>+300</sup> 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==1 5 ....
\$AC_MEA [1] **	Measuring task status	-	Default condition: 0: Default condition, probe did not switch 1: Probe switched	N10 IF \$AC_MEA[1]==1 GOTOF .... ; Continue program when probe has switched ...

Address	Significance	Value assignments	Information	Program- ming
\$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 ....
\$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_MOP1 1[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_MOP1 3[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
MCALL	Modal subprogram call	-	The subroutine in the block containing MCALL is called automatically after each successive block containing a path motion. The call acts until the next MCALL is called. Application example: Drilling a hole pattern	N10 MCALL CYCLE82(...); Separate block, drilling cycle N20 HOLES1(...); Row of holes N30 MCALL; Separate block, modal call of CYCLE82(...) completed
MSG ()	Signal	max. 65 characters	Message text in inverted commas	N10 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, special FRC= ... feed possible	N10 X... Y... <b>RND=4.5</b> N11 X... Y...

Address	Significance	Value assignments	Information	Program- ming
RNDM	Modal rounding	0.010 ... 99 999.999  0	<ul style="list-style-type: none"> <li>Inserts roundings with the specified radius value tangentially at the following contour corners; special feedrate possible: FRCM= ...</li> <li>modal rounding OFF</li> </ul>	N10 X... Y... <b>RNDM=7.3</b> ; modal rounding ON N11 X... Y... .... N100 <b>RNDM=0</b> ; modal rounding OFF
RP	Polar radius	0.001 ... 99 999.999	Traversing in polar coordinates, pole specification; in addition: Polar angle AP	see G0, G1, G2; G3 G110, G111, G112
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
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 (not applicable for tapping)	See G33
SPI(n)	Converts the spindle number n into the axis identifier		n =1 or =2, axis identifier: e.g. "SP1" or "C"	
SPOS	Spindle position	0.0000 ... 359.9999  with incremental specification (IC): ±0.001 ... 99 999.999	specified in degrees; the spindle stops at the specified position (to achieve this, the spindle must provide the appropriate technical prerequisites: position control)	N10 SPOS=... N10 SPOS=ACP(...) N10 SPOS=ACN(...) N10 SPOS=IC(...) N10 SPOS=DC(...)
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...
START-FIFO	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
TANG(Fo, Le1,Le2,...)	Tangential control, definition	-	Fo: Name of the following axis (rotary axis) Le1: Name of leading axis 1 Le2: Name of leading axis 2 Further parameters optionally Function only available with SINUMERIK 802D sl pro!	TANG(C,X,Y); Separate block  TANG(C,X,Y,1"W","P"); Max. number of parameters

Address	Significance	Value assignments	Information	Program- ming
TANGON (Fo,...)	Activate tangential control	-	Fo: Name of the following axis (rotary axis) This function is only available with SINUMERIK 802D sl pro!	TANGON(C); Separate block TANGON(C,Angle,Dist,Angleletol) ; Max. number of parameters
TANGOF (Fo)	Tangential control, deactivation	-	Fo: Name of the following axis (rotary axis) This function is only available with SINUMERIK 802D sl pro!	TANGOF(C); Separate block
TANGDEL (Fo)	Tangential control, clear definition	-	Fo: Name of the following axis (rotary axis) This function is only available with SINUMERIK 802D sl pro!	TANGDEL(C); Separate block
TLIFT(Fo)	Tangential control, insert intermediate block	-	Fo: Name of the following axis (rotary axis) This function is only available with SINUMERIK 802D sl pro!	TLIFT(C); Separate block
TRACYL (d) **	Milling of the peripheral surface	d: 1.000 ... 99 999.999	Kinematic transformation	TRACYL(20.4); Separate block; Cylinder diameter: 20.4 mm TRACYL(20.4,1) ; also possible
TRAFOOF **	Disable TRACYL	-	Disables all kinematic transformations	TRAFOOF ; separate block
TURN **	Number of additional circle passes with helix interpolation	0 ... 999	In conjunction with circular interpolation G2/G3 in a plane G17 to G19 and infeed motion of the axis vertical to this plane	N10 G0 G17 X20 Y5 Z3 N20 G1 Z-5 F50 N30 G3 X20 Y5 Z-20 I0 J7.5 TURN=2 ; total of three full circles
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>)
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,... , )

Address	Significance	Value assignments	Information	Program- ming
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 Plane selection: G17 to G19

### Functionality

To assign, for example, **tool radius and tool length compensations**, a plane with two axes is selected from the three axes X, Y and Z. In this plane, you can activate a tool radius compensation.

For drill and cutter, the length compensation (length1) is assigned to the axis standing vertically on the selected plane (see Section "Tool and tool offsets"). It is also possible to use a 3-dimensional length compensation for special cases.

Another influence of plane selection is described with the appropriate functions (e.g. Section "Support for the contour definition programming").

The individual planes are also used to define the **direction of rotation of the circle for the circular interpolation** CW or CCW. In the plane in which the circle is traversed, the abscissa and the ordinate are designed and thus also the direction of rotation of the circle. Circles can also be traversed in a plane other than that of the currently active G17 to G19 plane (see Section "Axis Movements").

The following plane and axis assignments are possible:

Table 9- 2 Plane and axis assignments

G function	Plane (abscissa/ordinate)	Vertical axis on plane (length compensation axis when drilling/milling)
G17	X/Y	Z
G18	Z/X	Y
G19	Y/Z	X

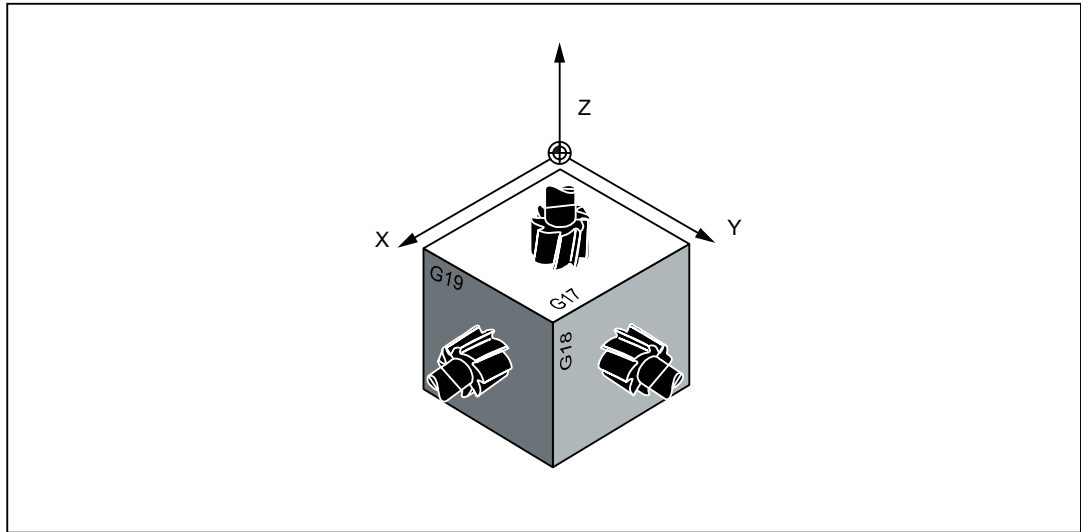


Figure 9-3 Planes and axes when drilling/milling

### Programming example

```
N10 G17 T... D... M... ; X/Y plane selected  
N20 ... X... Y... Z... ; tool length compensation (length1) in Z axis
```

## 9.2.3 Absolute/incremental dimensioning: G90, G91, AC, IC

### Functionality

With the instructions G90/G91, the written positional data X, Y, Z, ... are evaluated as a coordinate point (G90) or as an axis position to traverse to (G91). G90/G91 applies to 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
X=AC(...)	; Absolute dimensioning for a certain axis (here: X axis), non-modal
X=IC(...)	; Absolute dimensioning for a certain axis (here: X axis), non-modal

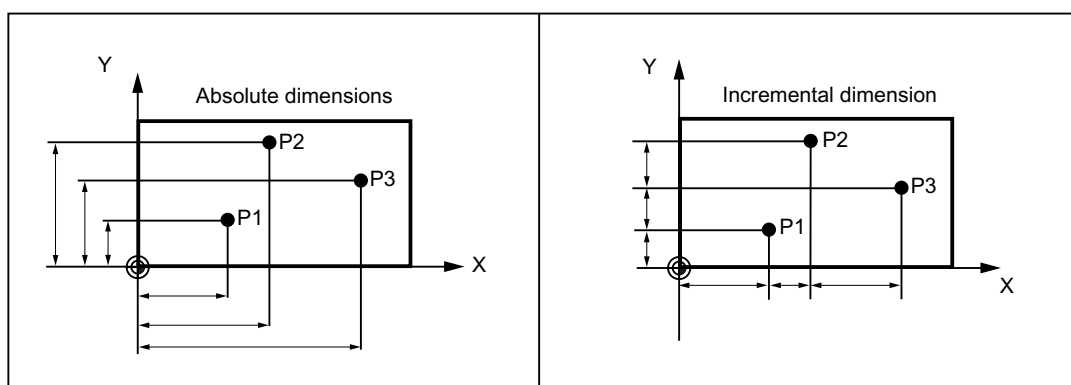


Figure 9-4 Different dimensioning types in the drawing

### 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 specified in round brackets.

Absolute dimensions are also possible for circle 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 Z20         ; Switch-over to incremental dimensioning
N190 X-12 Z=AC(17)       ; X-remains incremental dimensioning, Z-absolute
```

## 9.2.4 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 display are also to be understood as geometrical values; this also applies to the feedrate F in mm/min or inch/min. The default setting can be set via machine data.

All examples listed in this manual are based on a **metric default setting**.

G70 or G71 evaluates all geometric parameters that directly refer to the **workpiece**, either as inches or metric units, for example:

- Positional data X, Y, Z, ... for G0,G1,G2,G3,G33, CIP, CT
- Interpolation parameters I, J, K (also thread pitch)
- Circle radius CR
- **Programmable** work offset (TRANS, ATRANS)
- Polar radius RP

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.).

### 9.2.5 Polar coordinates, pole definition: G110, G111, G112

#### Functionality

In addition to the common specification in Cartesian coordinates (X, Y, Z), the points of a workpiece can also be specified using polar coordinates.

Polar coordinates are also helpful if a workpiece or a part of it is dimensioned from a central point (pole) with specification of the radius and the angle.

#### Plane

The polar coordinates refer to the plane activated with G17 to G19. In addition, the third axis standing vertically on this plane can be specified. When doing so, spatial specifications can be programmed as cylinder coordinates.

#### Polar radius RP=...

The polar radius specifies the distance of the point to the pole. It is stored and must only be written in blocks in which it changes, after changing the pole or when switching the plane.

#### Polar angle AP=...

The angle is always referred to the horizontal axis (abscissa) of the plane (for example, with G17: X axis). Positive or negative angle specifications are possible.

The polar angle remains stored and must only be written in blocks in which it changes, after changing the pole or when switching the plane.

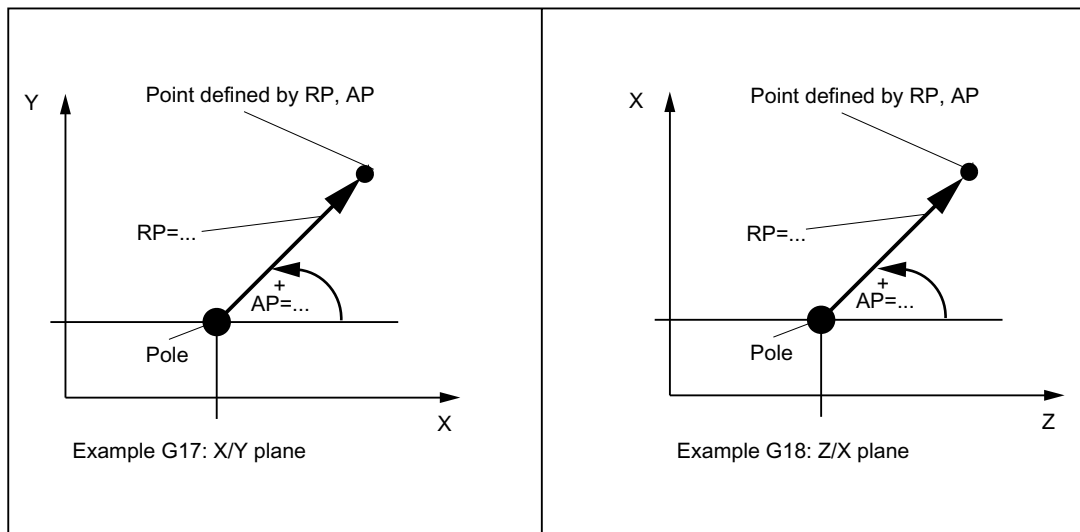


Figure 9-5 Polar radius and polar angle with definition of the positive direction in different planes

## Pole definition, programming

G110	Pole specification relative to the setpoint position last programmed (in the plane, e.g. with G17: X/Y)
G111	; Pole specification relative to the origin of the current workpiece coordinate system (in the plane, e.g. with G17: X/Y)
G112	; Pole specification, relative to the last valid pole; preserve plane

---

### Note

#### Pole specifications

- Pole definitions can also be performed using polar coordinates. This makes sense if a pole already exists.
  - If no pole is defined, the origin of the current workpiece coordinate system will act as the pole.
- 

## Programming example

```
N10 G17                                ; X/Y plane
N20 G111 X17 Y36                       ; Pole coordinates in the current workpiece
                                         coordinate system
...
N80 G112 AP=45 RP=27.8                 ; New pole, relative to the last pole as a polar
                                         coordinate
N90 ... AP=12.5 RP=47.679              ; Polar coordinate
N100 ... AP=26.3 RP=7.344 Z4           ; Polar coordinate and Z axis (= cylinder
                                         coordinate)
```

## Traversing with polar coordinates

The positions programmed using polar coordinates can also be traversed as positions specified with Cartesian coordinates as follows:

- G0 - linear interpolation with rapid traverse
- G1 - linear interpolation with feedrate
- G2 - circular interpolation CW
- G3 - circular interpolation CCW

(see also Section "Axis movements").

## 9.2.6 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.

### Programming

TRANS X... Y... Z...	; programmable offset, deletes old instructions for offsetting, rotation, scaling factor, mirroring
ATRANS X... Y... 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.

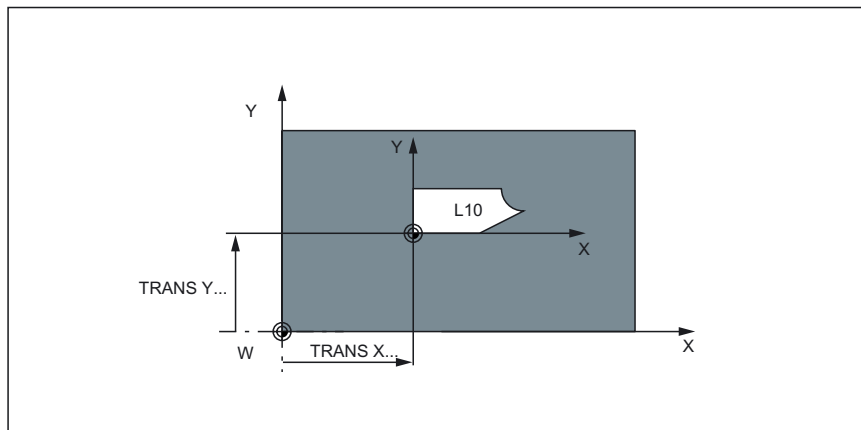


Figure 9-6 Programmable offset (example)

### Programming example

```
N20 TRANS X20 Y15      ; Programmable translation
N30 L10                ; Subroutine call; contains the geometry to be offset
...
N70 TRANS              ; Offset cleared
```

Subroutine call - see Section "Subroutine technique"



## 9.2.7 Programmable rotation: ROT, AROT

### Functionality

The rotation is performed in the current plane G17 or G18 or G19 using the value of RPL=... specified in degrees.

### Programming

ROT RPL=... ; Programmable rotation, deletes old instructions for offsetting, rotation, scaling factor, mirroring

AROT RPL=... ; Programmable rotation, additive to existing instructions

ROT ; Without values: clears old instructions for offset, rotation, scaling factor, mirroring

The instructions that contain ROT or AROT each require a separate block.

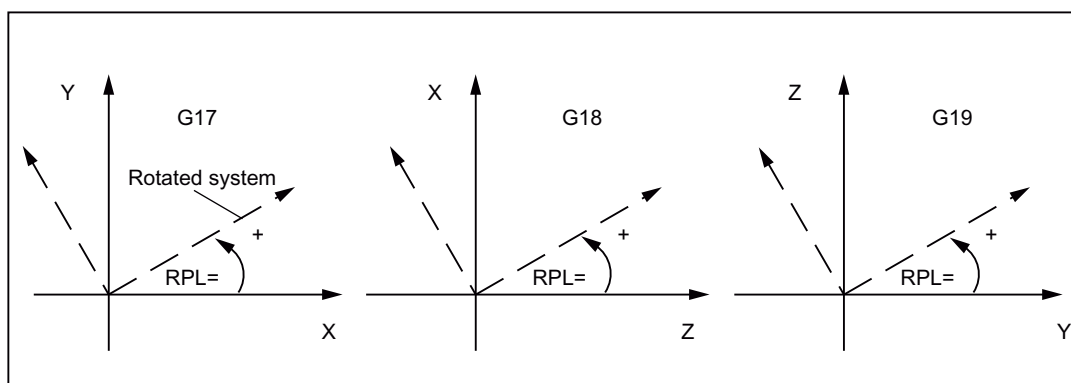


Figure 9-7 Definition of the positive direction of the angle of rotation in the individual planes

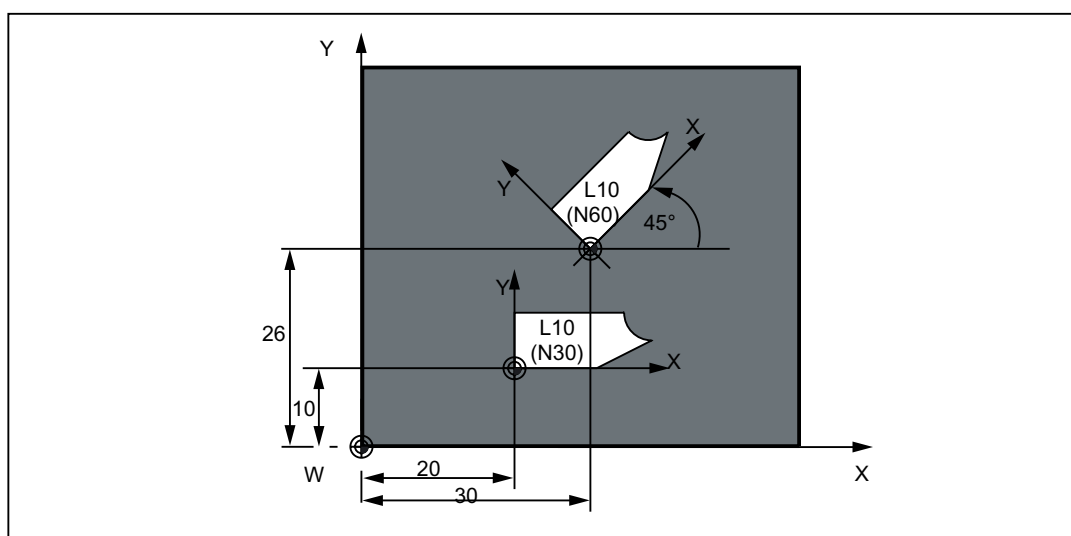


Figure 9-8 Programming example for programmable offset and rotation

### Programming example

```
N10 G17 ...           ; X/Y plane
N20 TRANS X20 Y10     ; Programmable translation
N30 L10               ; Subroutine call; contains the geometry to be offset
N40 TRANS X30 Y26     ; New offset
N50 AROT RPL=45       ; Additive 45 degree rotation
N60 L10               ; Subroutine call
N70 TRANS             ; Offset and rotation cleared
...
```

Subroutine call - see section "Subroutine technique"

### 9.2.8 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... Y... Z... ; Programmable scaling factor, clears old instructions for offset,
                    ; rotation, scaling factor, mirroring
ASCALE X... Y... 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.

---

#### Note

For circles, the same factor should be used in both axes.

If ATRANS is programmed with SCALE/ASCALE active, these offset values are also scaled.

---

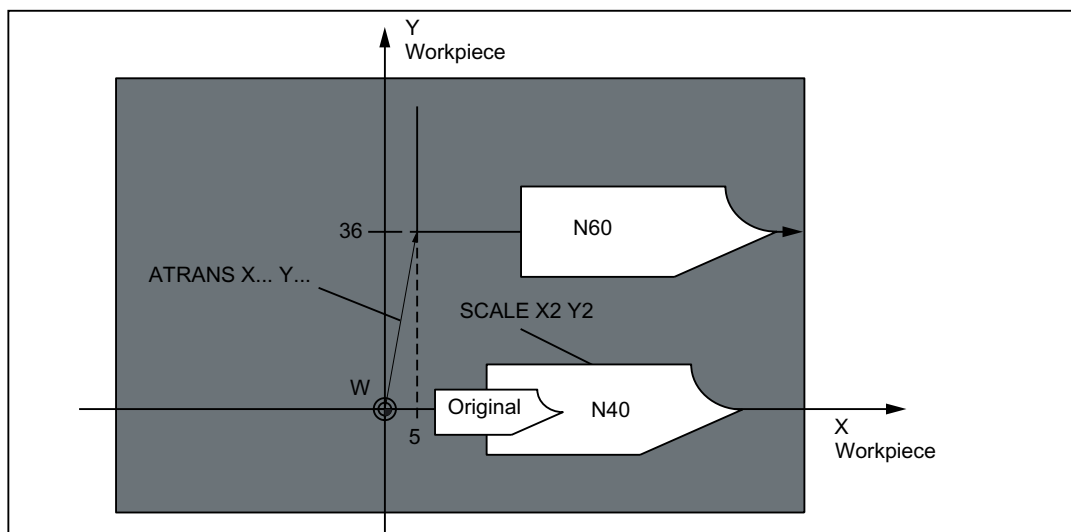


Figure 9-9 Example for scaling and offset

### Programming example

```

N10 G17 ; X/Y plane
N20 L10 ; Programmed contour original
N30 SCALE X2 Y2 ; Contour in X and Z enlarged two times
N40 L10
N50 ATRANS X2.5 Y18 ; Values are also scaled!
N60 L10 ; Contour enlarged and offset

```

Subroutine call - see section "Subroutine technique "

### 9.2.9 Programmable mirroring: MIRROR, AMIRROR

#### Functionality

MIRROR and AMIRROR can be used to mirror workpiece shapes on coordinate axes. All traversing motions of axes for which mirroring is programmed are reversed in their direction.

#### Programming

- MIRROR X0 Y0 Z0 ; Programmable mirroring, clears old instructions for offset, rotation, scaling factor, mirroring
- AMIRROR X0 Y0 Z0 ; Programmable mirroring, additive to existing instructions
- MIRROR ; Without values: clears old instructions for offset, rotation, scaling factor, mirroring

The instructions that contain MIRROR or AMIRROR each require a separate block. The axis value has no influence. A value, however, must be specified.

#### Note

Any active tool radius compensation (G41/G42) is reversed automatically when mirroring. The direction of rotation of the circle G2/G3 is also reversed automatically when mirroring.

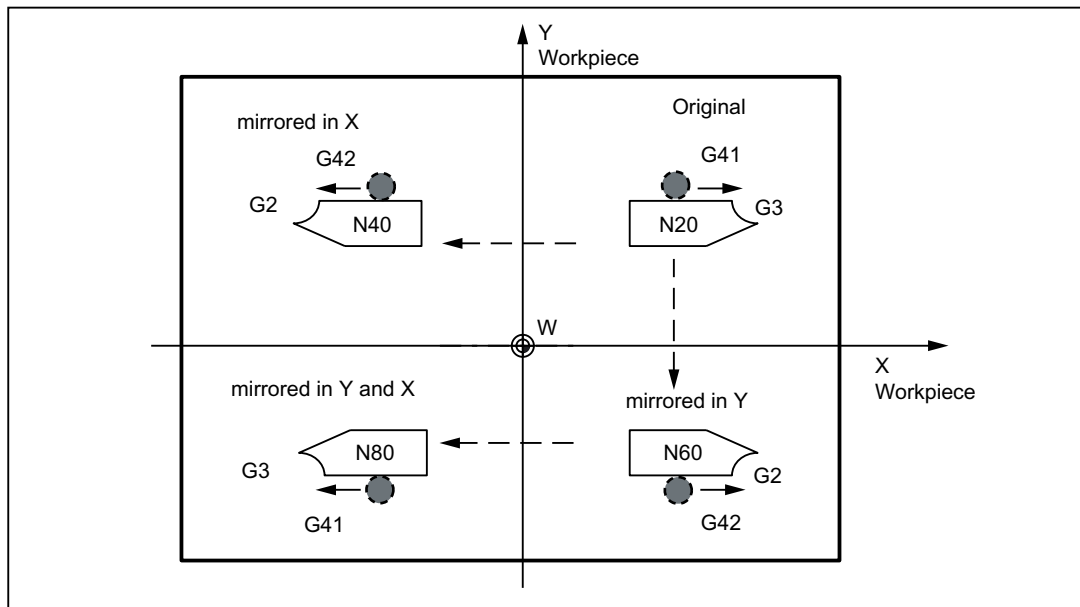


Figure 9-10 Example for mirroring with the tool position shown

## Programming example

Mirroring in different coordinate axes with influence on an active tool radius compensation and G2/G3:

```
...  
N10 G17                ; X/Y plane, Z standing vertically on it  
N20 L10                ; Programmed contour with G41  
N30 MIRROR X0         ; Direction changed in X  
N40 L10                ; Mirrored contour  
N50 MIRROR Y0         ; Direction changed in Y  
N60 L10  
N70 AMIRROR X0        ; Mirroring once more, but now in X  
N80 L10                ; Twice-mirrored contour  
N90 MIRROR            ; Mirroring off  
...
```

Subroutine call - see Section "Subroutine technique "

### 9.2.10 Workpiece clamping - settable work offset: G54 to G59, G500, G53, G153

#### Functionality

The settable work offset specifies the position of the **workpiece zero** on the machine (offset of the workpiece zero with respect to the machine zero). 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.

---

#### Note

Workpiece clamping at an angle is possible by entering the angles of rotation around the machine axes. These rotation portions are activated with the offset G54 to G59.

---

For information on operation, see Section "Setting/changing the work offset"

#### Programming

G54 to G59	; 1. to 6th settable work offset
G507 to G554	; 7th 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

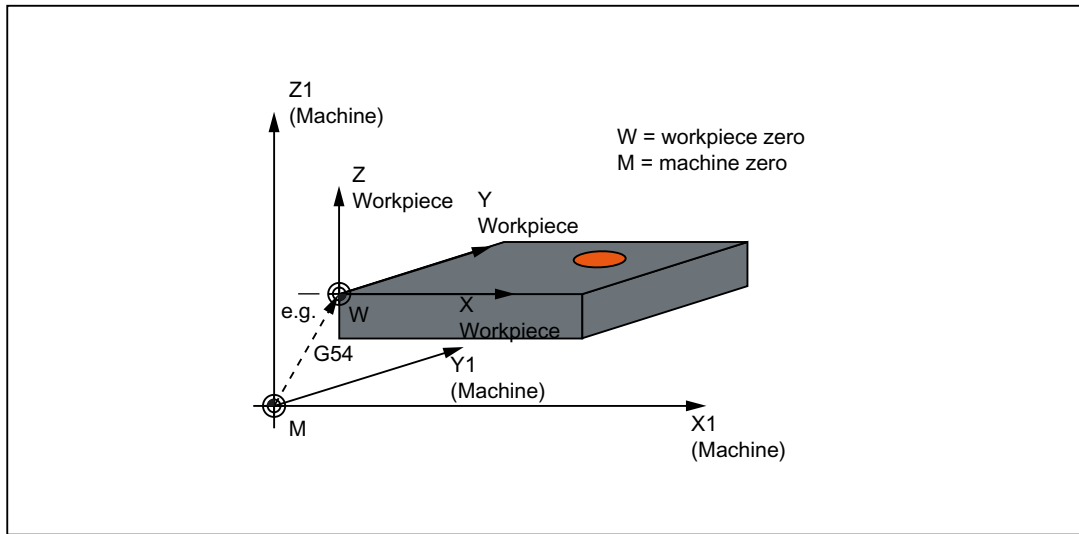


Figure 9-11 Settable work offset

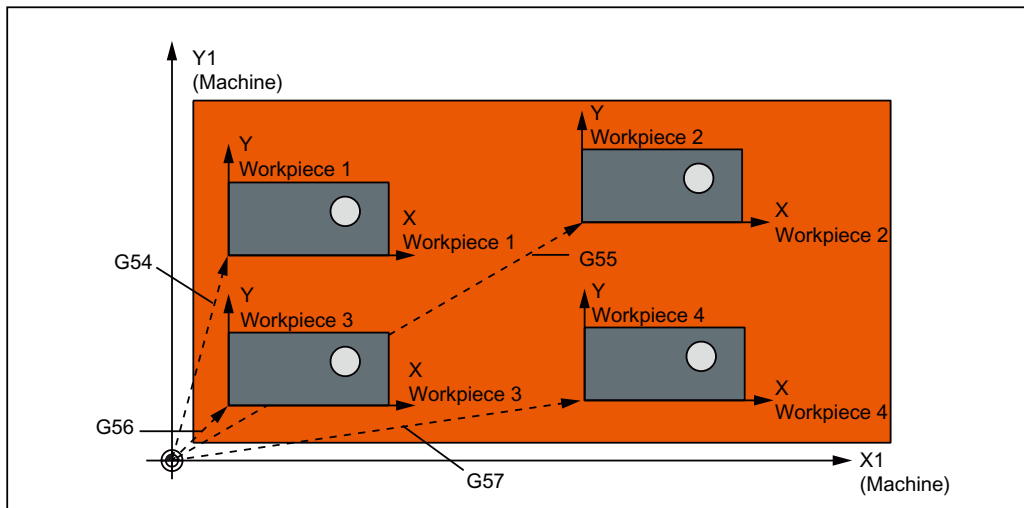


Figure 9-12 Various workpiece clamping positions when drilling/milling

**Programming example**

```

N10 G54 ...           ; Call first settable work offset
N20 L47              ; Machining of workpiece 1, here using L47
N30 G55 ...         ; Call second settable work offset
N40 L47              ; Machining of workpiece 2, here using L47
N50 G56 ...         ; Call third settable work offset
N60 L47              ; Machining of workpiece 3, here using L47
N70 G57 ...         ; Call fourth settable work offset
N80 L47              ; Machining of workpiece 4, here using L47
N90 G500 G0 X...    ; Deactivate settable work offset
    
```

Subroutine call - see section "Subroutine technique"

## 9.2.11 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 "Offset param" > "Setting data" > "Working area limit".

This makes the working area limitation effective in JOG mode as well.

- Programming with G25/G26

The values for the individual axes can be changed in the part program. The values that were input in the input screen ("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... Y... Z...           ; Lower working area limitation
G26 X... Y... Z...           ; Upper working area limitation
WALIMON                       ; Working area limitation ON
WALIMOF                       ; Working area limitation OFF
```

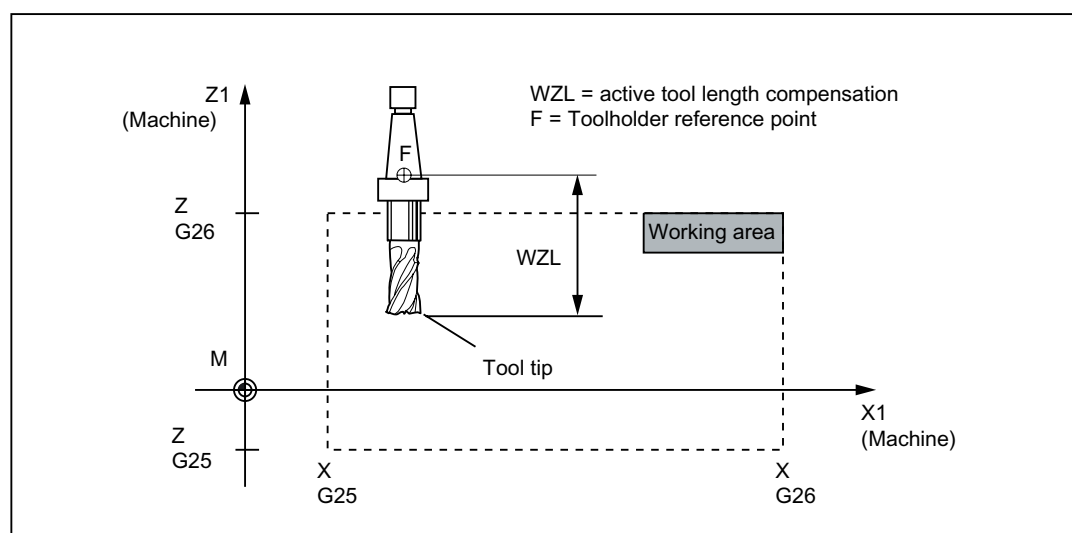


Figure 9-13 Programmable working area limitation (example: 2-dimensional)

---

**Note**

**Identifiers**

- The channel axis designation from MD20080 \$MC\_AXCONF\_CHANAX\_NAME\_TAB should be used for G25, G26.  
With SINUMERIK 802D sl, kinematic transformations (TRANSMIT, TRACYL) are possible. It is possible that different axis designations for MD20080 and the MD20060 \$MC\_AXCONF\_GEOAX\_NAME\_TAB geometry axis designation are specified here.
  - G25, G26 is also used in connection with the address S for the spindle speed limitation (see also Section "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 X10 Y-20 Z30      ; Values of the lower working area limitation
N20 G26 X100 Y110 Z300   ; Values of the upper working area limitation
N30 T1 M6
N40 G0 X90 Y100 Z180
N50 WALIMON              ; Working area limitation ON
...                      ; Work only within working range limits
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 rapid positioning of the tool, but not for **direct workpiece machining**.

All the 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 or three axes are traversed simultaneously, the path velocity (e.g. the resulting velocity at the tool tip) must be selected such that the **maximum possible path velocity** with consideration of all axes involved results.

A programmed feedrate (F word) has no meaning for G0. G0 remains active until canceled by another instruction from this G group (G1, G2, G3, ...).

#### Programming

G0 X... Y... Z...	; Cartesian coordinates
G0 AP=... RP=...	; Polar coordinates
G0 AP=... RP=... Z...	; Cylindrical coordinates (3dimensional)

---

#### Note

Another option for linear programming is available with the angle specification ANG=... (see Section "Blueprint programming").

---

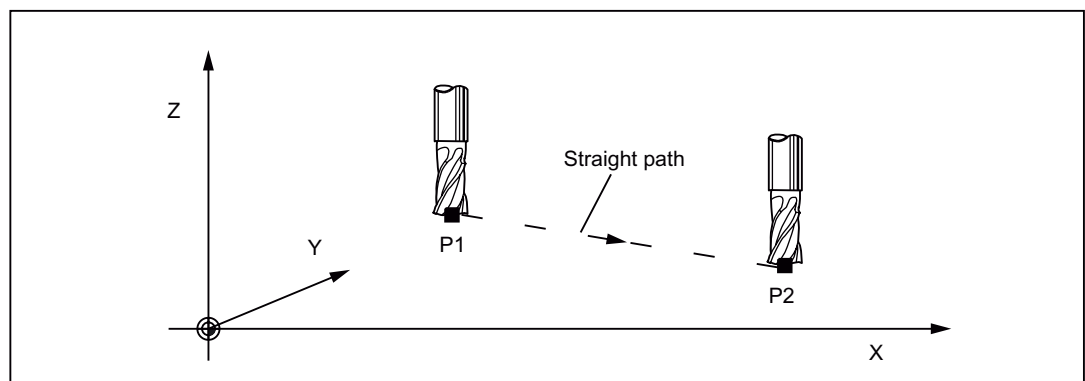


Figure 9-14 Linear interpolation with rapid traverse from point P1 to P2

### Programming example

```
N10 G0 X100 Y150 Z65 ; Cartesian coordinate  
...  
N50 G0 RP=16.78 AP=45 ; Polar coordinate
```

### Information

Another group of G functions exists for movement to the position (see Section "Exact stop / continuous-path control mode: G9, 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. The **path velocity** is determined by the programmed **F word**.

All axes can be traversed simultaneously.

G1 remains active until canceled by another instruction from this G group (G0, G2, G3, ...).

### Programming

```
G1 X... Y... Z... F... ; Cartesian coordinates  
G1 AP=... RP=... F... ; Polar coordinates  
G1 AP=... RP=... Z... F... ; cylindrical coordinates (3dimensional)
```

---

#### Note

Another option for linear programming is available with the angle specification ANG=... (see Section "Blueprint programming").

---

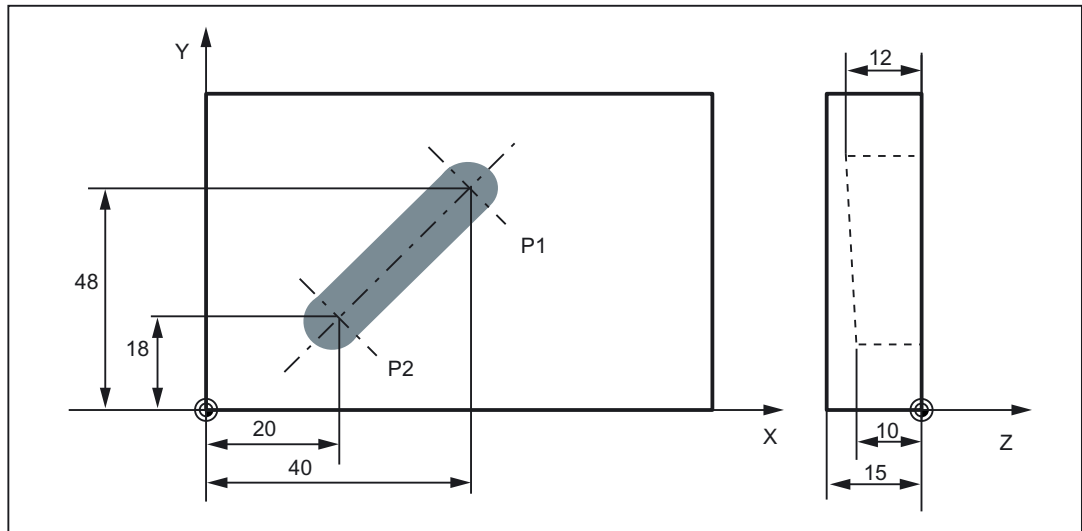


Figure 9-15 Linear interpolation in three axes using the example of a slot

### Programming example

<pre> N05 G0 G90 X40 Y48 Z2 S500 M3 N10 G1 Z-12 F100 N15 X20 Y18 Z-10 N20 G0 Z100 N25 X-20 Y80 N30 M2 </pre>	<pre> ; The tool traverses in rapid traverse on ; P1, three axes concurrently, spindle ; speed = 500 rpm, clockwise ; Infeed on Z-12, feed 100 mm/min ; Tool travels on a straight line in ; space on P2 ; Retraction in rapid traverse ; End of program </pre>
--	---

To machine a workpiece, spindle speed S ... and direction M3/M4 are required (see Section "Spindle movement").

### 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:

G2: clockwise

G3: counter-clockwise

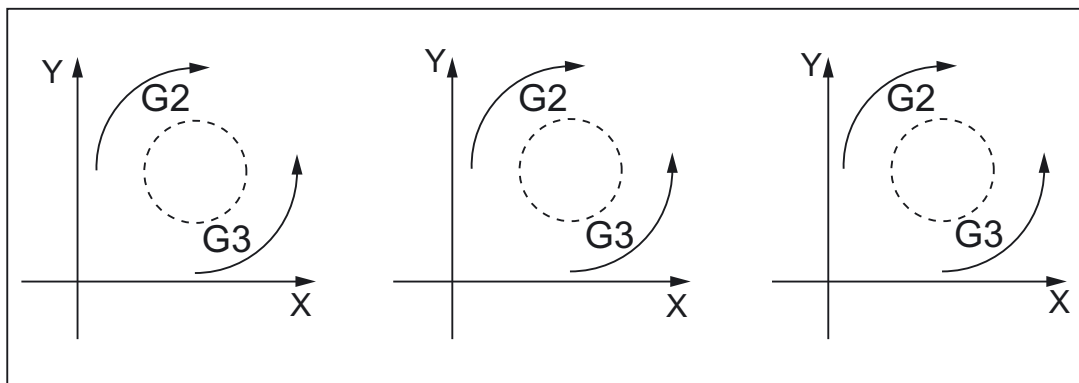


Figure 9-16 Definition of the direction of rotation of the circle G2/G3 in the three possible planes

The description of the desired circle can be given in various ways:

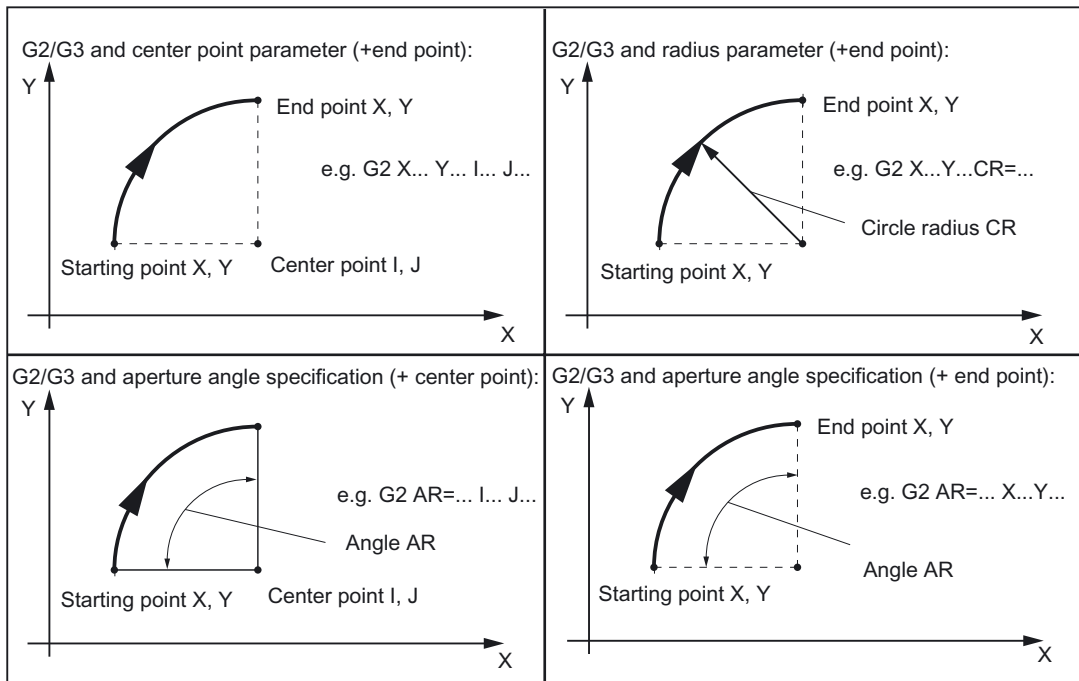


Figure 9-17 Possibilities of circle programming with G2/G3 using the example of the axes X/Y and G2

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 point
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).

## Information

**Full circles** in a block are only possible if the center point and the end point are specified.

For circles with radius specification, the arithmetic sign of CR=... is used to select the correct circle. It is possible to program two circles with the same starting and end points, as well as with the same radius and the same direction. The negative sign in front of CR=-... determines the circle whose circle segment is greater than a semi-circle; otherwise, the circle with the circle segment is less than or equal to the semi-circle and determined as follows:

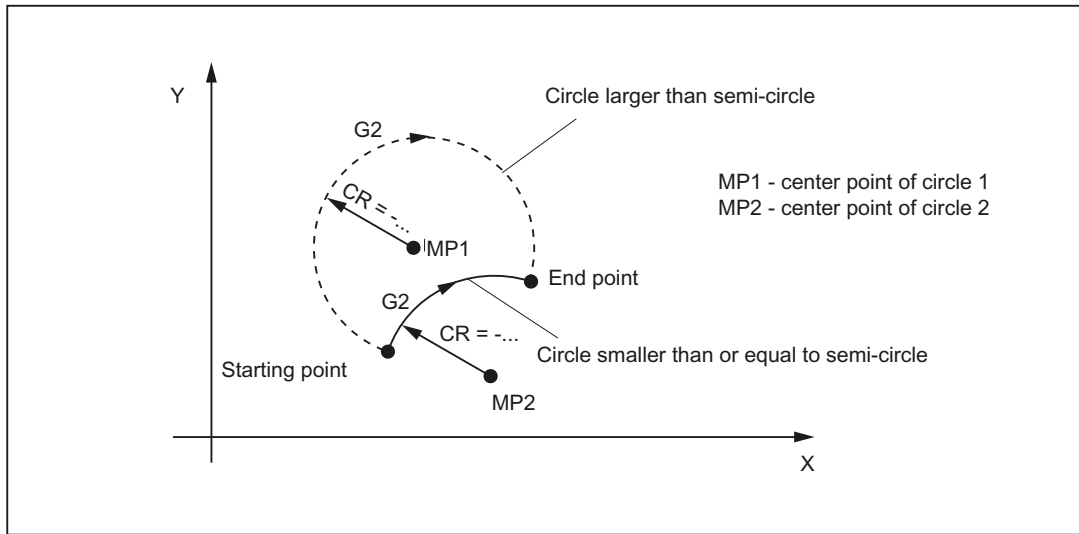


Figure 9-18 Selection of the circle from two possible circles with radius specification using the sign of CR=

**Programming example: Definition of center point and end point**

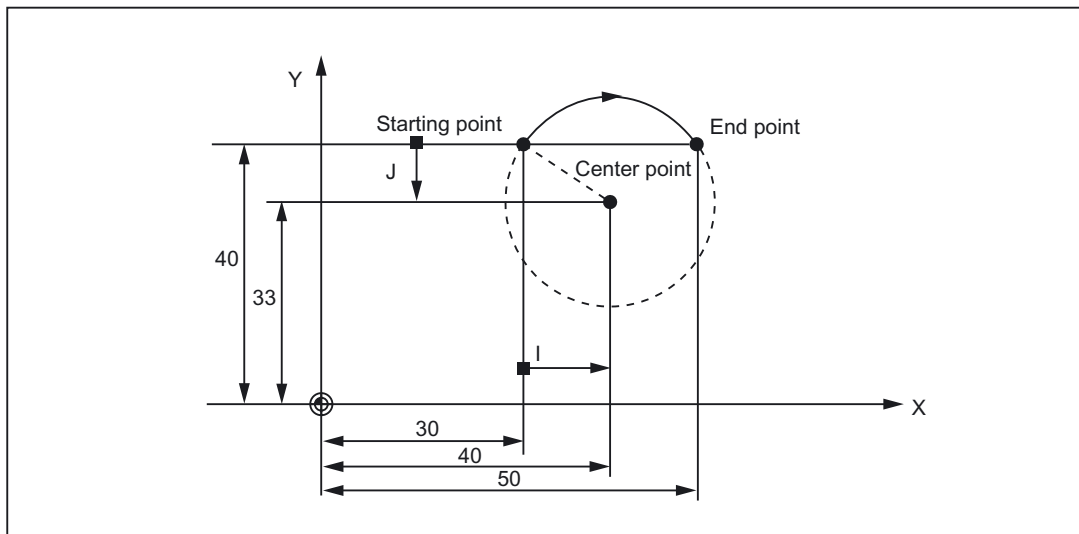


Figure 9-19 Example for center point and end point specification

```
N5 G90 X30 Y40 ; Starting point circle for N10
N10 G2 X50 Y40 I10 J-7 ; End point and center point
```

**Note**

Center point values refer to the circle starting point!

### Programming example: End point and radius specification

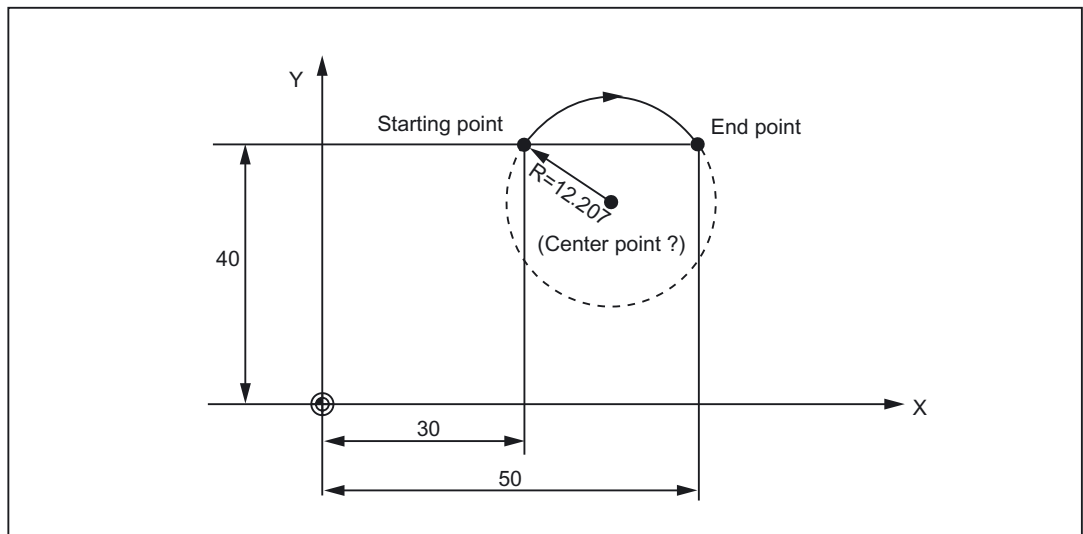


Figure 9-20 Example for end point and radius specification

```
N5 G90 X30 Y40 ; Starting point circle for N10  
N10 G2 X50 Y40 CR=12.207 ; End point and radius
```

#### Note

With a negative leading sign for the value with CR=-..., a circular segment larger than a semi-circle is selected.

Programming example: Definition of end point and aperture angle

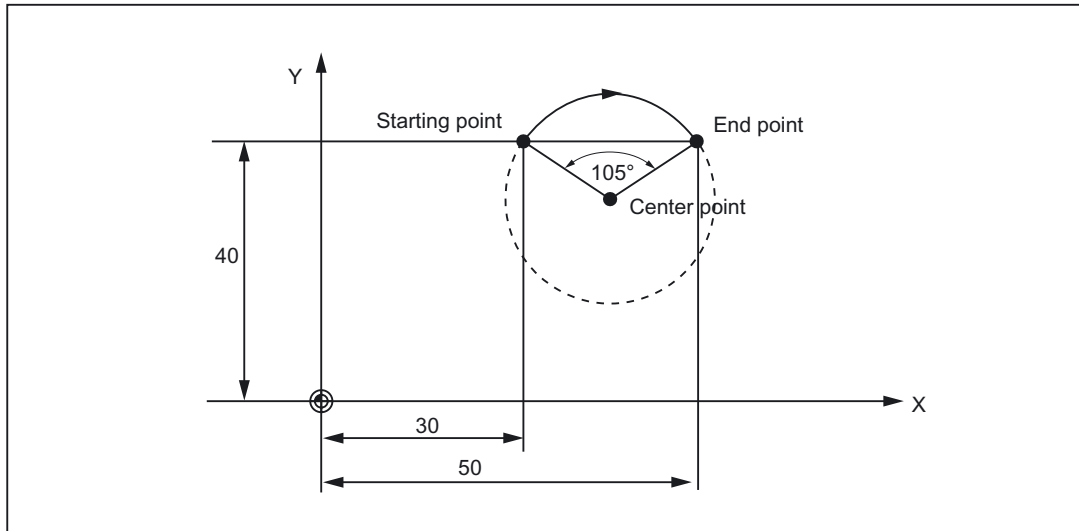


Figure 9-21 Example for end point and aperture angle specification

```
N5 G90 X30 Y40 ; Starting point circle for N10
N10 G2 X50 Y40 AR=105 ; End point and aperture angle
```

Programming example: Definition of center point and aperture angle

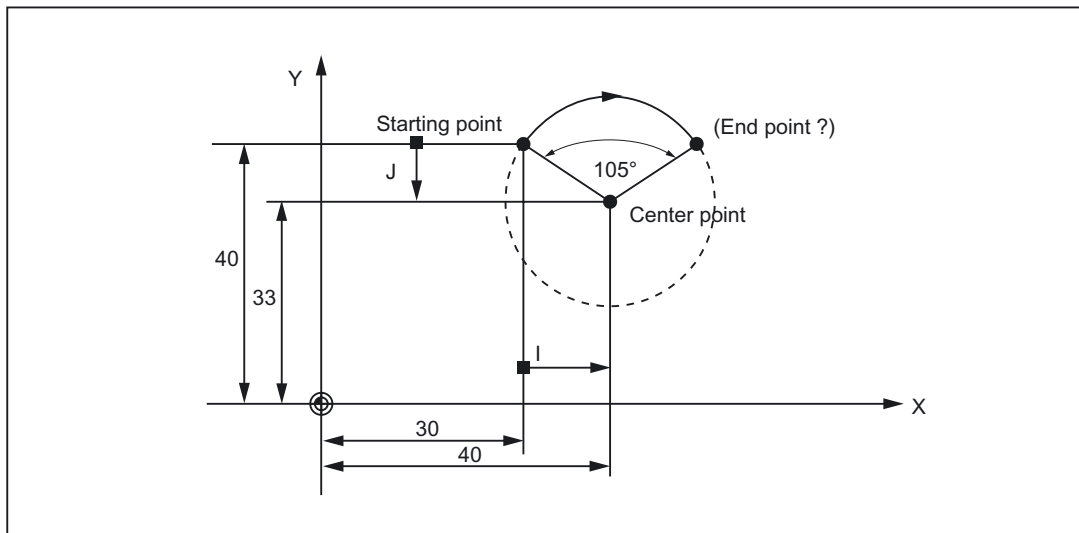


Figure 9-22 Example for center point and aperture angle specification

```
N5 G90 X30 Y40 ; Starting point circle for N10
N10 G2 I10 J-7 AR=105 ; Center point and aperture angle
```



**Note**

Center point values refer to the circle starting point!

**Programming example: Polar coordinates**

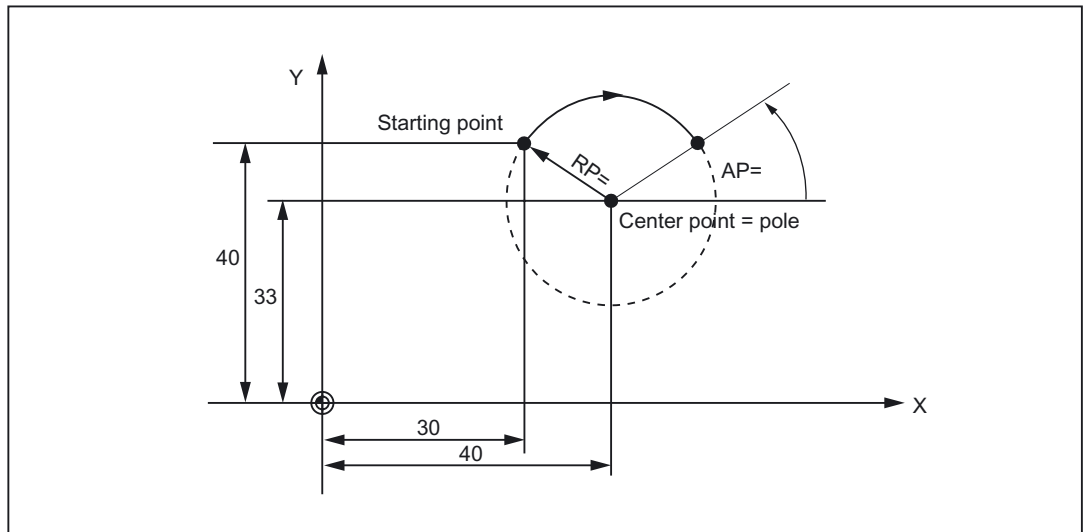


Figure 9-23 Example for circle with polar coordinates

```
N1 G17 ; X/Y plane
N5 G90 G0 X30 Y40 ; Starting point circle for N10
N10 G111 X40 Y33 ; Pole = circle center
N20 G2 RP=12.207 AP=21 ; Polar specifications
```

### 9.3.4 Circular interpolation via intermediate point: CIP

#### Functionality

If you know **three contour points** of the circle, instead of center point or radius or aperture angle, then it is advantageous to use the CIP function.

The direction of the circle results here from the position of the intermediate point (between starting and end points). The intermediate point is written according to the following axis assignment:

I1=... for the X axis,

J1=... for the Y axis,

K1=... for the Z axis.

CIP remains active until canceled by another instruction from this G group (G0, G1, G2, ...).

#### Note

The configured dimensional data G90 or G91 applies to the end point **and** the intermediate point.

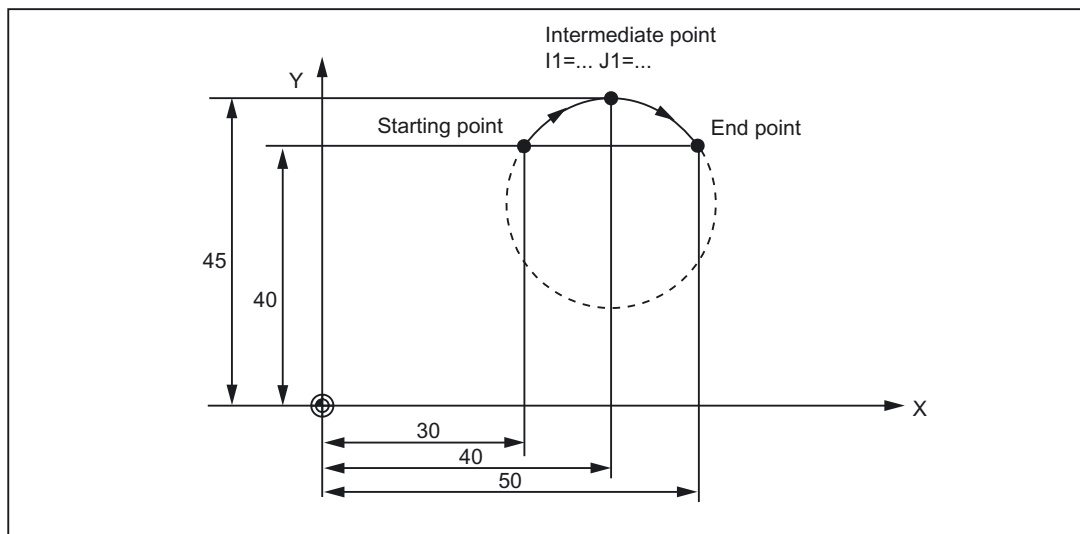


Figure 9-24 Circle with end point and intermediate point specification using the example of G90

#### Programming example

```
N5 G90 X30 Y40 ;Starting point circle for N10
N10 CIP X50 Y40 I1=40 J1=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 G17 through G19, a circle is generated which is connected tangentially to the previous path segment (circle or straight line) in this plane.

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.

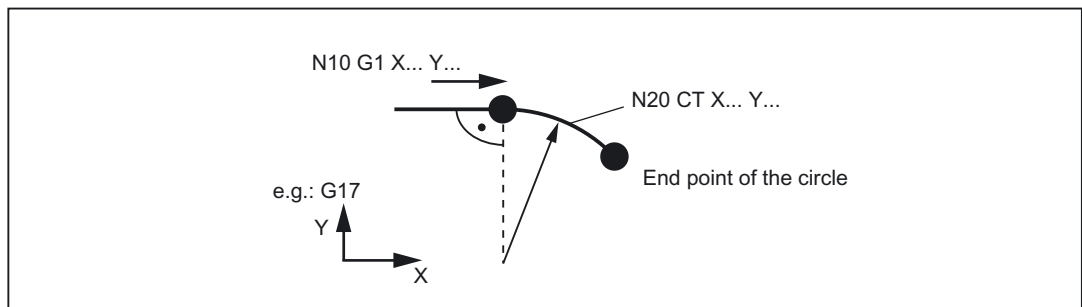


Figure 9-25 Circle with tangential transition to the previous path section

#### Programming example

```
N10 G1 X20 F300 ; Straight line  
N20 CT X... Y... ; Circle with tangential connection
```

### 9.3.6 Helix interpolation: G2/G3, TURN

#### Functionality

With helix interpolation, two movements are overlaid:

- Circular movement in the G17, G18 or G19 plane
- Linear movement of the axis standing vertically on this plane.

The number of additional full-circle passes is programmed with TURN=. These are added to the actual circle programming.

The helix interpolation can preferably be used for the milling of threads or of lubricating grooves in cylinders.

#### Programming

G2/G3 X... Y... I... J... TURN=...	; Center and end points
G2/G3 CR=... X... Y... TURN=...	; Circle radius and end point
G2/G3 AR=... I... J... TURN=...	; Opening angle and center point
G2/G3 AR=... X... Y... TURN=...	; Opening angle and end point
G2/G3 AP=... RP=... TURN=...	; Polar coordinates, circle around the pole

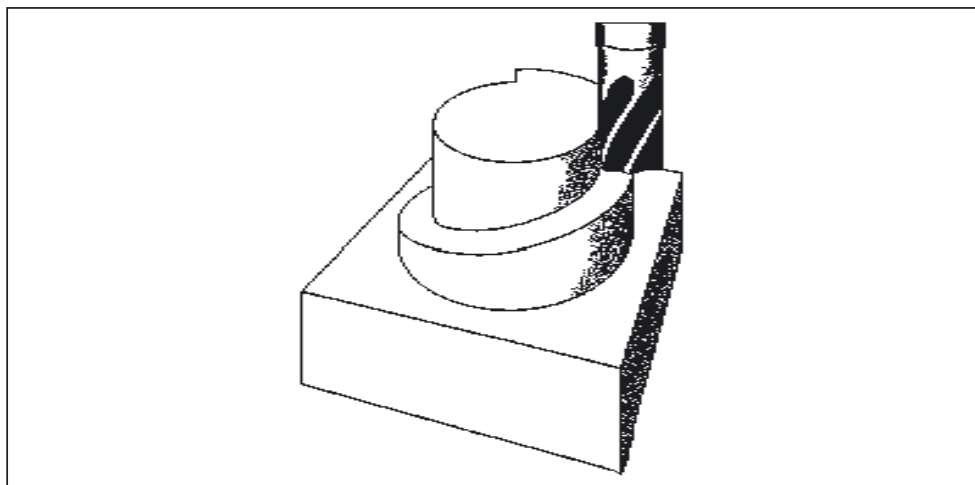


Figure 9-26 Helical interpolation

#### Programming example

```
N10 G17 ; X/Y plane, Z standing vertically on it
N20 ... Z...
N30 G1 X0 Y50 F300 ; Approach starting point
N40 G3 X0 Y0 Z33 I0 J-25 TURN= 3 ; Helix
...
```

### 9.3.7 Thread cutting with constant lead: G33

#### Functionality

This requires a spindle with position measuring system.

The function G33 can be used to machine threads with constant lead of the following type: If an appropriate tool is used, tapping with compensating chuck is possible.

The compensating chuck compensates the resulting path differences to a certain limited degree.

The drilling depth is specified by specifying one of the axes X, Y or Z; the thread pitch is specified via the relevant I, J or K.

G33 remains active until canceled by another instruction from this G group (G0, G1, G2, G3, ...).

#### Right-hand or left-hand thread

Right-hand or left-hand thread is set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) - see Section "Spindle movement"). To do this, the rotation value must be programmed under address S or a rotation speed must be set.

---

#### Note

A complete cycle of tapping with compensating chuck is provided by the standard cycle CYCLE840.

---

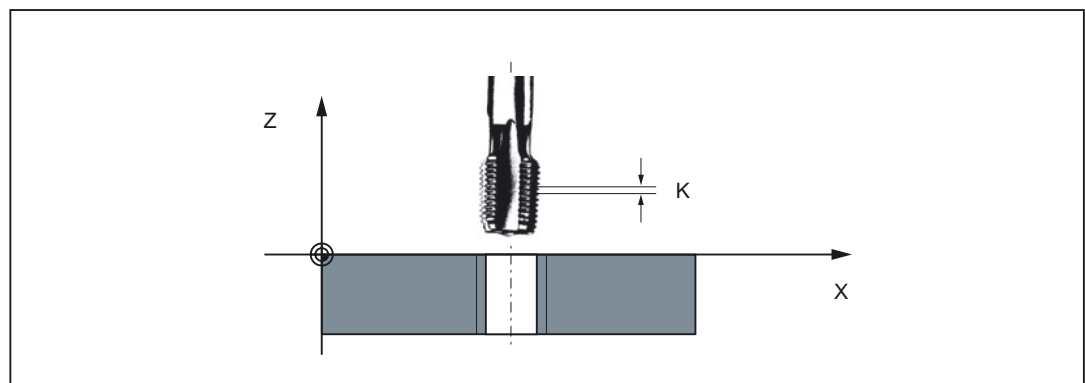


Figure 9-27 Tapping using G33

#### Programming example

<pre>N10 G54 G0 G90 X10 Y10 Z5 S600 M3 N20 G33 Z-25 K0.8</pre>	<pre>; metric thread 5, ; pitch as per table: 0.8 mm/rev., hole already premachined ; Approach starting point, clockwise spindle rotation ; Tapping, end point -25 mm</pre>
--	---

```
N40 Z5 K0.8 M4 ; Retraction, counter-clockwise spindle
rotation
N50 G0 X... Y... Z...
```

### Axis velocity

With G33 threads, the velocity of the axis for the thread lengths is determined on the basis of the spindle speed and the thread pitch. 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.

---

#### Note

##### Override switch

- The spindle speed override switch should remain unchanged for thread machining.
  - The feedrate override switch has no meaning in this block.
- 

## 9.3.8 Tapping with compensating chuck: G63

### Functionality

G63 can be used for tapping with compensating chuck. The programmed feedrate F must match with the spindle speed S (programmed under the address "S" or specified speed) and with the thread pitch of the drill:

$$F \text{ [mm/min]} = S \text{ [rpm]} \times \text{thread pitch [mm/rev.]}$$

The compensating chuck compensates the resulting path differences to a certain limited degree.

The drill is retracted using G63, too, but with the spindle rotating in the opposite direction M3 <-> M4.

G63 is non-modal. In the block after G63, the previous G command of the "Interpolation type" group (G0, G1,G2, ...) is active again.

### Right-hand or left-hand thread

Right-hand or left-hand thread is set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) - see Section "Spindle movement").

---

#### Note

The standard cycle CYCLE840 provides a complete tapping cycle with compensating chuck (but with G33 and the relevant prerequisites).

---

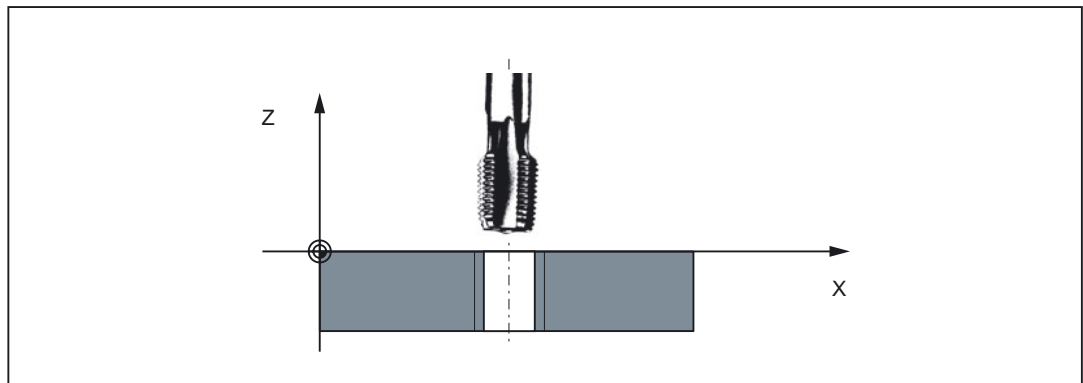


Figure 9-28 Tapping using G63

### Programming example

```
N10 G54 G0 G90 X10 Y10 Z5 S600 M3      ; metric thread 5,  
                                         ; lead as per table: 0.8 mm/rev., hole  
                                         ; already premachined  
                                         ; Approach starting point, clockwise  
                                         ; spindle rotation  
N20 G63 Z-25 F480                       ; Tapping, end point -25 mm  
N40 G63 Z5 M4                           ; Retraction, counter-clockwise spindle  
                                         ; rotation  
N50 X... Y... Z...
```

### 9.3.9 Thread interpolation: G331, G332

#### Functionality

This requires a position-controlled spindle with a position measuring system.

By using G331/G332, threads can be tapped **without** compensating chuck if the dynamic properties of the spindle and the axis allow it.

If, however, a compensating chuck is used, the path differences to be compensated by the compensating chuck are reduced. This allows tapping at higher spindle speeds.

Drilling is done using G331, retraction is done using G332.

The drilling depth is specified by specifying one of the axes X, Y or Z; the thread pitch is specified via the relevant I, J or K.

For G332, the same lead is programmed as for G331. Reversal of the spindle direction of rotation occurs automatically.

The spindle speed is programmed with S and without M3/M4.

Before tapping the thread using G331/G332, the spindle must be switched to the position-controlled mode with SPOS=... (see also Section "SPOS spindle positioning").

#### 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)

---

#### Note

A complete thread tapping cycle with thread interpolation is provided with the standard cycle CYCLE84.

---

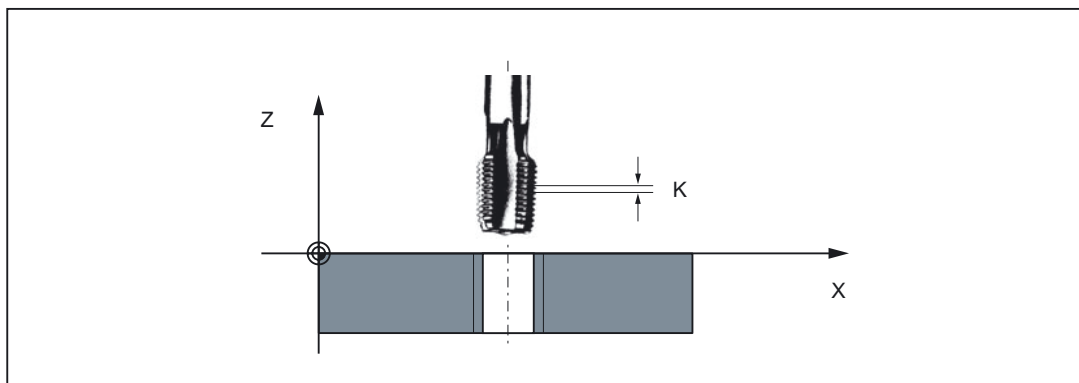


Figure 9-29 Tapping using G331/G332



## 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 as per table: 0.8 mm/rev., hole  
already premachined:  
N5 G54 G0 G90 X10 Y10 Z5           ; Approach starting point  
N10 SPOS=0                          ; Spindle in position control  
N20 G331 Z-25 K0.8 S600             ; Tapping, K positive = clockwise  
                                     of the spindle, end point Z=-25 mm  
N40 G332 Z5 K0.8                    ; Retraction  
N50 G0 X... Y... 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 speed of each axis is its rapid traverse.

G75 requires a separate block and is non-modal. The machine axis identifier must be programmed!

In the block after G75, the previous G command of the "Interpolation type" group (G0, G1,G2, ...) is active again.

#### Programming

G75 FP=<n> X1=0 Y1=0 Z1=0

#### Note

FPn references with axis machine data MD30600 \$MA\_FIX\_POINT\_POS[n-1]. If no FP has been programmed, then the first fixed point will be selected.

Table 9- 3 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 is specified, fixed point 1 is approached automatically.
X1=0 Y1=0 Z1=0	Machine axes to be traversed to the fixed point. Here, 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 Z1=0 ; Approach fixed point 1 in Z
N10 G75 FP=2 X1=0 Y1=0 ; Approach fixed point 2 in X and Y,
e. g. to change a tool
N30 M30 ; End of program
    
```

---

**Note**

The programmed position values for X1, Y1, 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. The direction and speed of each axis are stored in machine data.

G74 requires a separate block and is non-modal. 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 Y1=0 Z1=0
```

---

**Note**

The programmed position values for X1, Y1, Z1 (any value, 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 edge of a connected measuring probe are registered and stored. The measurement result can be read for each axis in the program.

For MEAS, the movement of the axes is halted when the selected switching edge of the probe appears and the remaining distance to go is deleted.

#### Programming

MEAS=1	G1 X... Y... Z... F...	; Measuring with rising edge of the probe, clearing the distance to go
MEAS=-1	G1 X... Y... Z... F...	; Measuring with falling edge of the probe, clearing the distance to go
MEAW=1	G1 X... Y... Z... F...	; Measuring with rising edge of the probe, <b>without</b> clearing the distance to go
MEAW=-1	G1 X... Y... Z... F...	; Measuring with falling edge of the probe, <b>without</b> clearing the distance to go

 <b>CAUTION</b>
--

For MEAW: Measuring probe travels to the programmed position even after it has triggered. Risk of destruction!
--

#### Measuring job status

If the probe has switched, the variable \$AC\_MEA[1] after the measuring block has the value =1; otherwise, value = 0.

At the start of a measuring block, the variable is set to the value=0.

#### Measuring result

When the probe is successfully activated, the result of the measurement is available for the axes traversed in the measuring block with the following variables after the measuring block:

in the machine coordinate system: \$AA\_MM[*axis*]

in the workpiece coordinate system: \$AA\_MW[*axis*]

### Programming example

```

N10 MEAS=1 G1 X300 Z-40 F4000           ; Measurement with deletion of distance-
                                         to-go,
                                         ; Probe with 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 Tangential control: TANG, TANGON, TANGOF, TLIFT, TANGDEL

### Functionality

The function is only available for SINUMERIK 802D sl pro.

This function is intended for use in technologies other than milling.

When using the SINUMERIK control system in fields of technology where, for example, a tool must be guided along the tangent (i.e. tangentially) towards the workpiece contour to be traversed, the "tangential control" will be used. For example:

- Controlling the workpiece alignment when machining using a belt saw
- Controlling a pinion-type cutter when machining glass, leather, textiles or paper

The TANG( ) function is used to define coupled axes using a coupling factor. Coupled axes include a following axis (rotary axis) and two master axes (axes of the machining plane). The following axis is guided along the tangent to the path traversed by the master axes.

TANGON( ) is used to enable the coupling, TANGOF( ) is used to disable the coupling. It is possible to specify an offset angle for the following axis (rotary axis) by specifying an angle programmed in TANGON( ). The TANGDEL( ) instruction can be used to delete a defined coupling in the disabled condition.

Defined parameters or values must be transferred using the relevant functions. If no values are specified for the axes in all parameters, these need not be written.

### Programming

```

TANG ( Faxis, Laxis1, Laxis2, Coupling, CS, Opt ) ; Definition of the tangential coupling
TANGON ( Faxis, Angle, Dist, Angletol ) ; Activate tangential control
TANGOF ( Faxis ) ; Deactivate tangential control

```

TLIFT( <i>Faxis</i> )	; Insert intermediate block at contour corners
TANGDEL( <i>Faxis</i> )	; Delete the definition of the tangential coupling

### Explanation of the parameters

<i>Faxis</i>	Following axis (tangentially followed rotary axis)
<i>Laxis1, Laxis</i>	Master axis 1 and 2 (path axes, which determine the tangent for the correction of the following axis).
<i>Couple</i>	Coupling factor (relationship between the angle change of the tangent and the following axis). Optional specification, default = 1
<i>C</i>	Identifier for "coordinate system", specified optionally: "B" = basic coordinate system (default)
<i>Opt</i>	Optimization: "S" = Default or "P" = Automatic adaptation of the time characteristic for following and master axes
<i>Angles,</i>	Offset angle of following axis
<i>Dist</i>	Smoothing path of following axis, required with Opt = "P"
<i>Angletol</i>	Angle tolerance of following axis (optional), evaluation only with Opt = "P"

### Information

If *Opt* = "P", the dynamic properties of the following axis are taken into account in the velocity limitation of the master axes.

The parameters (*Dist* and *Angletol*) limit the error between the following axis and the tangent of the master axes precisely. Any sudden velocity changes of the following axis due to sudden changes in the leading axis contour are rounded or smoothed using (*Dist* and *Angletol*). The following axis is controlled using LookAhead to keep the deviation as low as possible.

Inserting an intermediate block at contour corners: **TLIFT( )**

At one corner of the contour the tangent changes and thus the setpoint position of the following axis. Normally, the axis attempts to compensate this sudden change with its maximum possible velocity. This, however, results in a deviation to the desired tangential setting over a certain distance on the contour after the corner. If this cannot be tolerated for technological reasons, the TLIFT( ) instruction can be used to cause the control system to stop at the corner and to rotate the following axis in the new tangent direction in an automatically generated intermediate block. The angle change from which an automatic intermediate block is inserted can be configured via machine data.

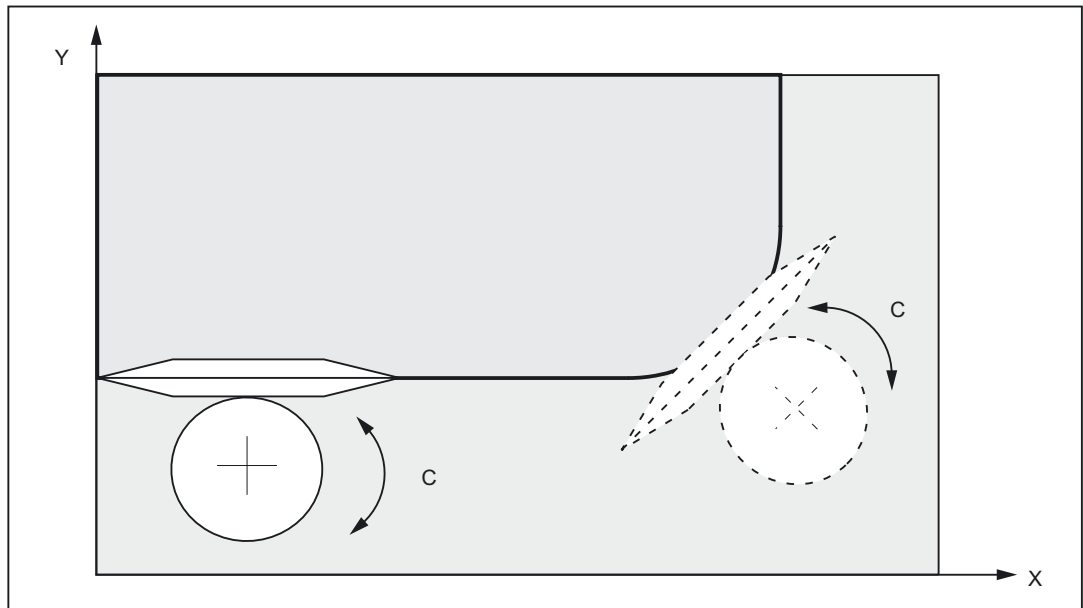


Figure 9-30 Tangential control, e.g. of a pinion-type cutter: X, Y = master axes, C = following axis

### Programming example

```

N10 TANG(C, X, Y,1)           ; Define coupling of the tangential control
N20 ...                       ; Approach starting point
N30 TANGON(C,0)               ; Activate coupling, C axis alignment 0 degrees
N40 G1 F800 X10 Y20           ; Retract from contour in X, Y
...
N100 TANGOF(C)                ; Deactivate coupling
...
N200 M2

```

### 9.3.14 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 individual axis velocities therefore result from the portion of the axis path in the overall distance to be traversed.

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...

---

#### Note

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: Feedrate F in mm/spindle revolutions  
(only meaningful when the spindle is running)

---

#### Note

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
```

---

#### Note

Write a new F word if you change G94 - G95.

---



### 9.3.15 Feedrate override for circles: CFTCP, CFC

#### Functionality

For activated **tool radius compensation** (G41/G42, see Section "Selection of the tool radius compensation: G41/G42") **and circle programming**, it is imperative to correct the feedrate at the cutter center point if the **programmed F value** is to act at the circle contour.

Internal and external machining of a circle and the current tool radius are taken into account automatically if the tool radius compensation is enabled.

This feedrate correction (override) is not necessary for linear paths. The path velocities at the cutter center point and at the programmed contour are identical.

If you wish the programmed feedrate always to act at the cutter center point path, then disable the feedrate override. The modally acting G group that contains CFTCP/CFC (G functions) is provided for switching.

#### Programming

CFTCP ; Feedrate override OFF (the programmed feedrate acts at the milling cutter center point)  
CFC ; Feedrate override with circle ON

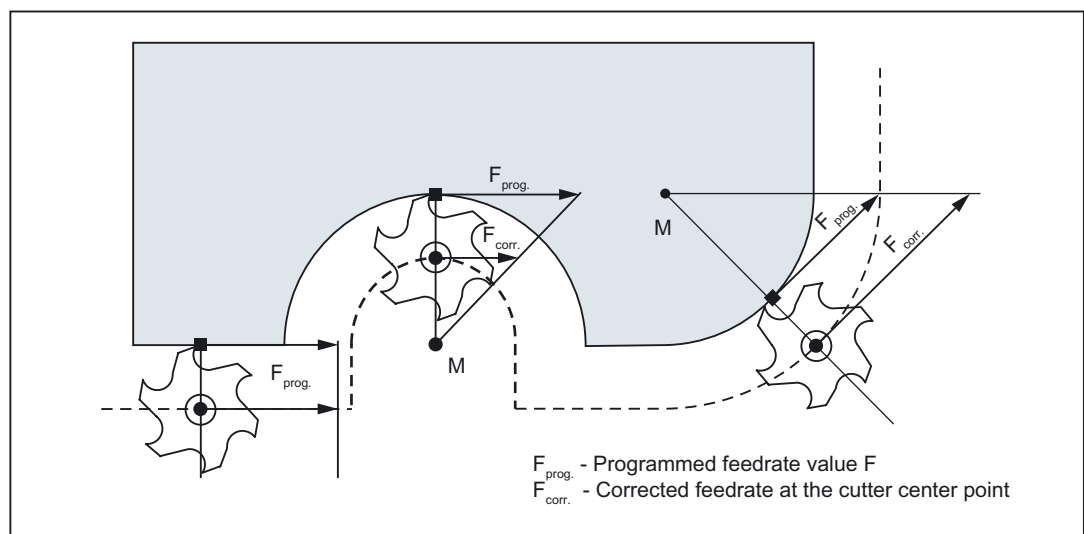


Figure 9-31 Feedrate override G901 with internal / external machining

### Corrected feedrate

- External machining:

$$F_{\text{corr.}} = F_{\text{prog.}} \cdot (r_{\text{cont}} + r_{\text{tool}}) / r_{\text{cont}}$$

- Internal machining:

$$F_{\text{corr.}} = F_{\text{prog.}} \cdot (r_{\text{cont}} - r_{\text{tool}}) / r_{\text{cont}}$$

$r_{\text{cont}}$ : Radius of the circle contour

$r_{\text{tool}}$ : Tool radius

### Programming example

```
N10 G42 ... ; Tool radius compensation ON
N20 CFC ... ; Feedrate override with circle ON
N30 G2 X... Y... I... J... ; Feed value acts on contour
F350
N40 G3 X... Y... I... J... ; Feed value acts on contour
...
N70 CFTCP ; Feedrate override OFF, programmed feedrate value
acts at the milling cutter center point
```

### 9.3.16 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 boundaries and for block advancing. Example: 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 modally effective
G64	; Continuous-path mode
G9	; Exact stop non-modally effective
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.

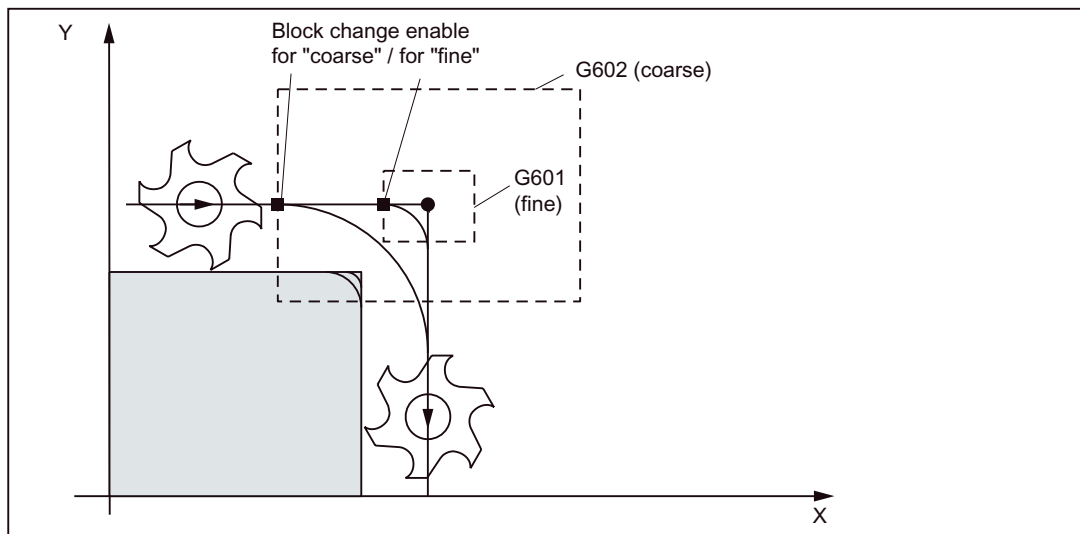


Figure 9-32 Exact stop window coarse or fine, in effect for G60/G9; enlarged display of the windows

### Programming example

```

N5 G602 ; Exact stop window coarse
N10 G0 G60 X... ; Exact stop modal
N20 X... Y... ; G60 continues to act
...
N50 G1 G601 ... ; Exact stop window fine
N80 G64 X... ; Switching over to continuous-path mode
...
N100 G0 G9 X... ; Exact stop acts only in this block
N111 ... ; Again continuous-path mode
    
```

---

#### Note

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 X... F...      ; Continuous-path mode
N20 Y..                   ; 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.

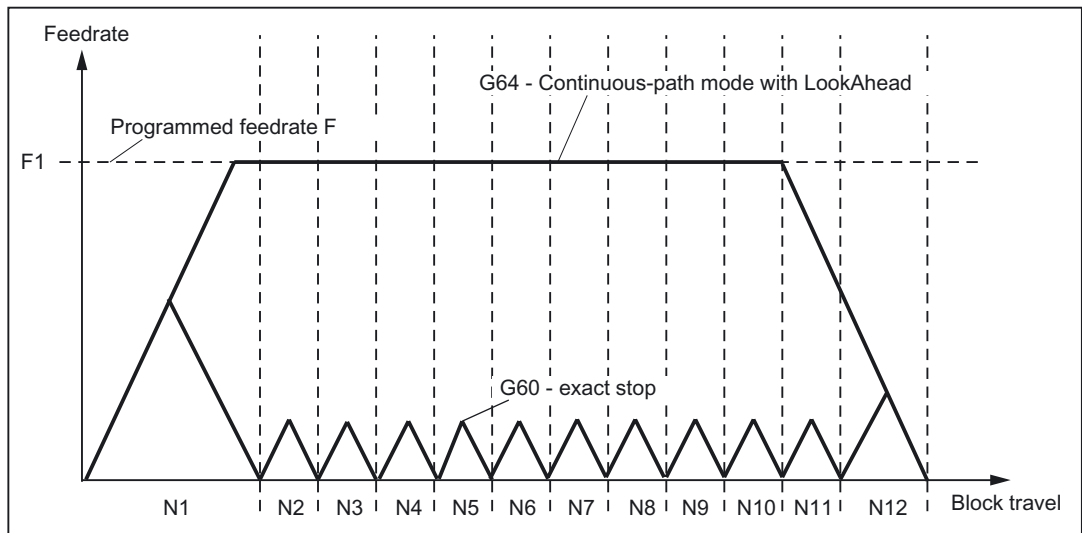


Figure 9-33 Comparison of the G60 and G64 velocity behavior with short travels in the blocks

### 9.3.17 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 along a non-linear, constant characteristic 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.

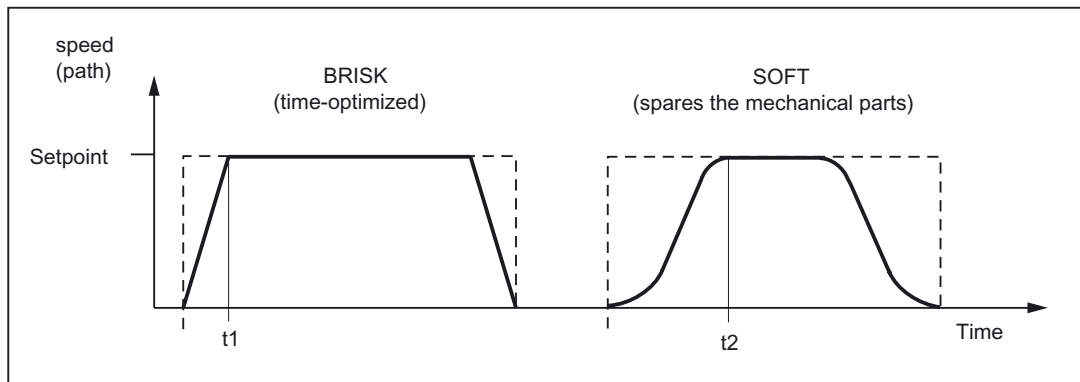


Figure 9-34 Basic course of the path velocity when using BRISK/SOFT

#### Programming

BRISK ; Jerking path acceleration  
SOFT ; Jerk-limited path acceleration

#### Programming example

```
N10 SOFT G1 X30 Z84 F650 ; Jerk-limited path acceleration  
...  
N90 BRISK X87 Z104 ; Continuing with jerking path acceleration  
...
```

### 9.3.18 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  $\leq 200\%$  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 (depending on whether it is the axis or spindle; for the spindle it depends further on the gear stage and whether it is in 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
```

#### Effectivity

The limitation is effective in all interpolation types of the AUTOMATIC and MDA modes. The limitation is not active in JOG mode and during 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.

 <b>CAUTION</b>
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.

### 9.3.19 Traversing with feedforward control: FFWON, FFWOF

#### Functionality

The feedforward control can be used to reduce the following error in the traversed path to zero.

Traversing with feedforward control allows higher path accuracy and thus improved machining results.

#### Programming

FFWON ; Feedforward control ON  
FFWOF ; Feedforward control OFF

#### Programming example

```
N10 FFWON ; Feedforward control ON  
N20 G1 X... Y... Z... F900  
...  
N80 FFWOF ; Feedforward control OFF
```



## 9.3.20 Enhanced surface quality through compressor: COMPCAD

### Functionality

This function is only available for SINUMERIK 802D sl pro.

CAD/CAM systems normally produce linear blocks that meet the configured accuracy specifications. In the case of complex contours, a large volume of data and short path sections can result. The short path sections restrict the processing rate. By using the compressor, it is possible to summarize short path blocks in a path section. The number of blocks to be executed is compressed. Internally, the G1 linear interpolation is converted into a polynomial interpolation.

By using the COMPCAD G code, you can select a compression optimized with reference to the surface quality and the velocity, whereby the accuracy of the interpolation can be defined via machine data.

COMPCAD is processor and memory-intensive. COMPCAD should only be used if measures to improve the surface cannot be taken by the CAD/CAM system in advance.

Features:

- COMPCAD produces polynomial blocks with a continuous acceleration.
- With adjacent paths, deviations head in the same direction.
- With the SD42470 setting data: CRIT\_SPLINE\_ANGLE can be used to define a limit angle from which the COMPCAD will leave the corners.
- COMPCAD eliminates poor surface transitions. In doing so, however, the tolerances are largely adhered to but the corner limit angle is ignored.

### Programming

```
COMPCAD      ; Compressor for surface optimization ON  
COMPOF      ; Compressor OFF
```

### Programming example

```
N10 G0 X30 Y6 Z40  
N20 G1 F10000  
N30 SOFT  
N40 COMPCAD      ; Compressor for surface optimization ON  
N50 STOPFIFO    ; Stops the fast machining step  
N60 Z32.499  
N61 X41.365 Z32.500  
N62 X43.115 Z32.497  
N63 X43.365 Z32.477  
N64 X43.556 Z32.449  
N65 X43.818 Z32.387  
N66 X44.076 Z32.300
```

```

...
N80 COMPOF                ; Compressor OFF
N90 G0 Z50
N100 M2
    
```

**Information**

The compressor processes blocks that contain X, Y and Z axis motions. Any M commands, changes in the spindle speed, etc. will interrupt the compressor.

This function must be configured during the start-up using various machine data.

**References**

SINUMERIK 802D sl instruction manual for turning, milling, grinding, nibbling

**9.3.21 Fourth axis**

**Functionality**

Depending on the machine variant, a fourth axis may be necessary, e.g. rotary table, swivel axis, etc. This axis can be provided as linear or rotary axis. The designation of this axis must be configured accordingly, e.g. U, C, or A etc. For rotary axes, the traversing range can be configured between 0 ...360 degrees (modulo-behavior).

With an appropriate machine design, the fourth axis can be traversed linear simultaneously with the remaining axes. If the axis is traversed together with the remaining axes (X, Z) in a block that contains G1 or G2/G3, no component of the feedrate F is assigned to this axis; its velocity will depend on the time for traveling along the X and Z axes, and its motion starts and ends with the remaining path axes. The velocity, however, must not be greater than the defined limit value.

If in one block only this fourth axis is programmed, the axis will traverse using the active feedrate F if G1 is active. If the axis is a rotary axis, the unit for F is degrees/min with G94 or degrees/rev. of the spindle with G95.

For this axis, it is also possible to specify (G54 ... G59) and program (TRANS, ATRANS) offsets.

**Programming example**

The fourth axis is a swivel table (rotary axis) and has the axis identifier A:

```

N5 G94                    ; Feedrate F in mm/min or degrees/min
N10 G0 X10 Y20 Z30 A45    ; Traverse X-Y-Z path with rapid traverse, A at the
                           same time
N20 G1 X12 Y21 Z33 A60 F400 ; Traverse X-Y-Z path at 400 mm/min, A at the same
                           time
N30 G1 A90 F3000          ; Axis A 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
```

### 9.3.22 Dwell time: G4

#### Functionality

Between two NC blocks, you can interrupt the machining for a defined time by inserting a **separate block** with G4; e.g. for relief cutting.

The words with F... or S... are only used in this block for the specified time. 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 F200 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
```

---

#### Note

G4 S.. is only possible if a controlled spindle is available (if the speed specifications are also programmed via S...).

---

### 9.3.23 Travel to fixed stop

#### Functionality

This function is available for 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 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

---

#### Note

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 FXST[Z1]=12.3 FXSW[Z1]=2
           ; for Z1 machine axis FXS function selected,
           ; clamping torque 12.3%,
           ; window width 2 mm
```

---

#### Note

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.

---

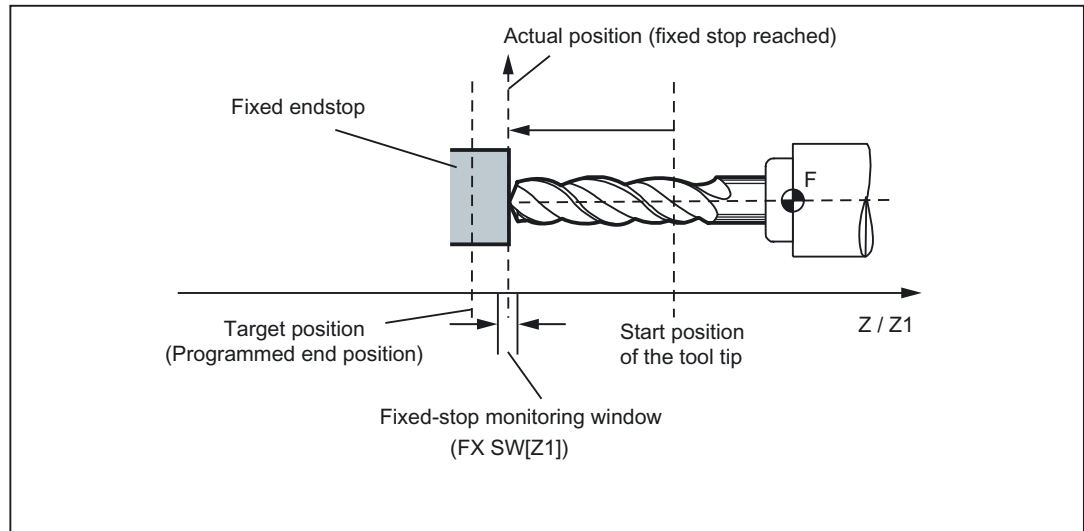


Figure 9-35 Example of travel to fixed stop: The tool is traversed along the Z axis against a fixed stop.

### 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 FXST[X1]=12.3	; Selected for machine axis X1 FXS, 12.3% clamping torque and window width from SDs
N20 X250 Z100 F100 FXS[X1]=1 FXST[X1]=12.3 FXSW[X1]=2	; Selected for machine axis X1 FXS, 12.3% clamping torque and window width 2 mm
N20 X250 Z100 F100 FXS[X1]=1 FXSW[X1]=2	; Selected for machine axis X1 FXS, 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 FXS[X1] = 0 ; Axis X1 is retracted from the fixed
stop to position X= 200 mm.
```

**Note**

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.

**Note**

"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. insertion of a spindle sleeve or quill).

**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 successfully approached  
(axis is in 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"

## References

SINUMERIK 802D sl Function Manual for Turning, Milling, Nibbling; Travel to Fixed Stop

### 9.3.24 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

<code>FENDNORM</code>	; Automatic corner deceleration OFF
<code>G62</code>	; Corner deceleration at inside corners when tool radius offset is active
<code>G621</code>	; 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:

$$F * (\text{override for feed rate reduction}) * \text{feed rate override}$$

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.**

**9.3.25 Coupled axes**

**9.3.25.1 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

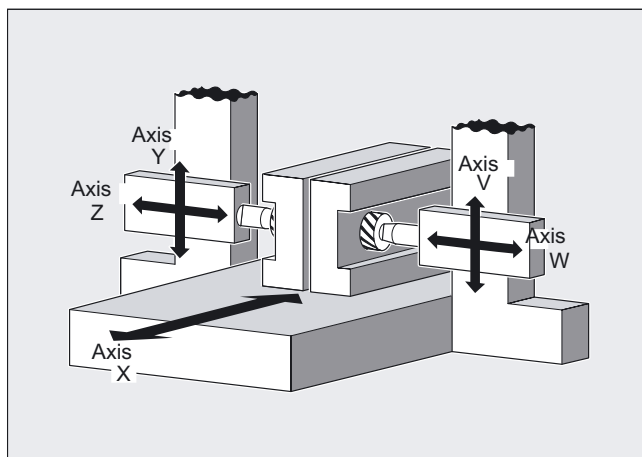


Figure 9-36 Coupled motion, example two-side machining



## 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 <b>Note:</b> 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: $\text{<coupling factor>} = \text{path of coupled-motion axis} / \text{path of leading axis}$ 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.

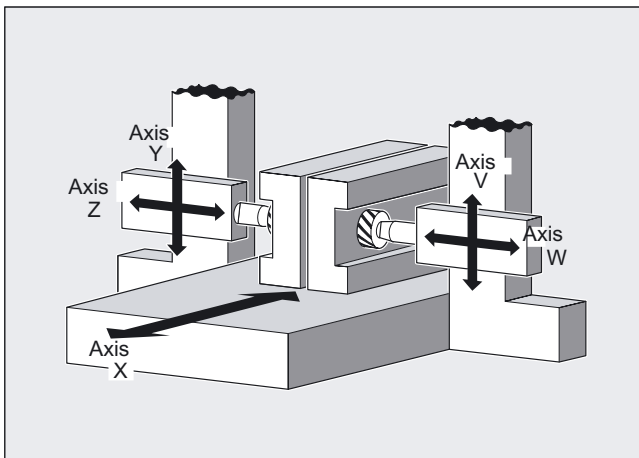


Figure 9-37 Coupled motion, programming example

```

...
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.

```

### 9.3.25.2 Coupled axis motion (TRAILON, TRAILOF): Further Information

#### 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 ( $\$AA\_TYP \neq 1$ ), 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.

 <b>CAUTION</b>
--

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

### 9.3.25.3 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
```

### 9.3.25.4 Master/slave group (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS): Further Information

#### 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.

MASLOFS The master/slave definitions specified in the machine data are kept. 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 in revolutions per minute under the address S provided that the machine possesses a controlled spindle.

The direction of rotation and the start or end of the movement are specified via M commands (also see Section "Miscellaneous function M").

M3: Spindle clockwise

M4: Spindle counter-clockwise

M5: Spindle stop

---

#### Note

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.

**Default setting:** The axis movements will only start once the spindle has accelerated to speed (M3, M4). M5 is also issued before the axis movement. However, there is no waiting for spindle standstill. The axis movements begin before spindle standstill.

The spindle is stopped at program end or with RESET.

At program start, spindle speed zero (S0) is in effect.

---

#### Note

Other settings can be configured via machine data.

---

#### Programming example

```
N10 G1 X70 Z20 F300 S270 M3 ; Before the axis traversing X, Z the spindle  
                                accelerates to 270 rpm, 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 for a controlled spindle by writing G25 or G26 and the spindle address S with the speed limit value. This overwrites the values entered in the setting data at the same time.

G25 and G26 each require a separate block. 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. Appropriate inputs via the operator panel can activate various setting data for further limiting.

### Programming example

```
N10 G25 S12           ; Lower spindle speed limitation : 12 rpm  
N20 G26 S700          ; Upper spindle speed limitation : 700 rpm
```

---

#### Note

G25/G26 are used in conjunction with axis addresses for a working area limitation (see Section "Working area limitation").

---



### 9.4.3 Spindle positioning: SPOS

#### Functionality

**Requirement:** The spindle must be technically designed for position control.

With the function SPOS= you can position the spindle in a specific **angular position**. The spindle is held in the position by position control.

The **speed** of the positioning procedure is defined in machine data.

With SPOS=*value* from the M3/M4 movement, the respective **direction of rotation** is maintained until the end of the positioning. When positioning from standstill, the position is approached via the shortest path. The direction results from the respective start and end position.

Exception: First movement of the spindle, i.e. if the measuring system is not yet synchronized. In this case, the direction is specified in machine data.

Other movement specifications for the spindle are possible with SPOS=ACP(...), SPOS=ACN(...), ... as for rotary axes (see Section "4. Axis").

The spindle movement takes place parallel to any other axis movements in the same block. This block is ended when both movements are finished.

#### Programming

SPOS=...	; Absolute position: 0 ... <360 degrees
SPOS=ACP(...)	; Absolute dimensions, approach position in positive direction
SPOS=ACN(...)	; Absolute dimensions, approach position in negative direction
SPOS=IC(...)	; Incremental dimensions, leading sign determines the traversal direction
SPOS=DC(...)	; Absolute dimensions, approach position directly (on the shortest path)

#### Programming example

```

N10 SPOS=14.3           ; Spindle position 14.3 degrees
...
N80 G0 X89 Z300 SPOS=25.6 ; Positioning spindle with axis movements
                        ; This block is ended when all movements have
                        ; finished
N81 X200 Z300          ; The N81 block only begins once the spindle position
                        ; from N80 is reached

```

## 9.4.4 Gear stages

### Function

Up to 5 gear stages can be configured for a spindle for speed / torque adaptation. The selection of a gear stage takes place in the program via M commands (see Section "Miscellaneous function M"):

- M40: Automatic gear stage selection
- M41 to M45: Gear stage 1 to 5

## 9.5 Contour programming support

### 9.5.1 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.

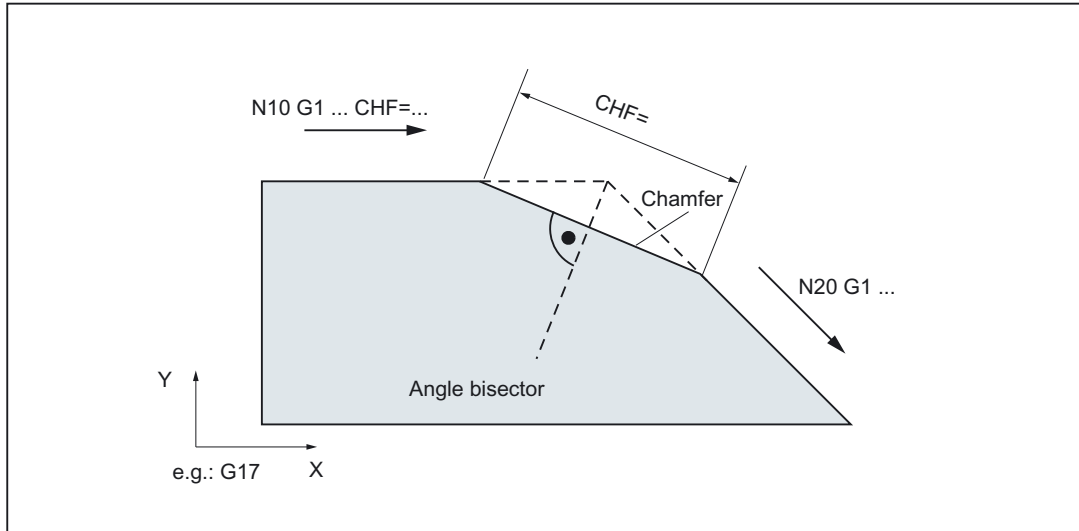


Figure 9-38 Inserting a chamfer with CHF using the example: Between two straight lines

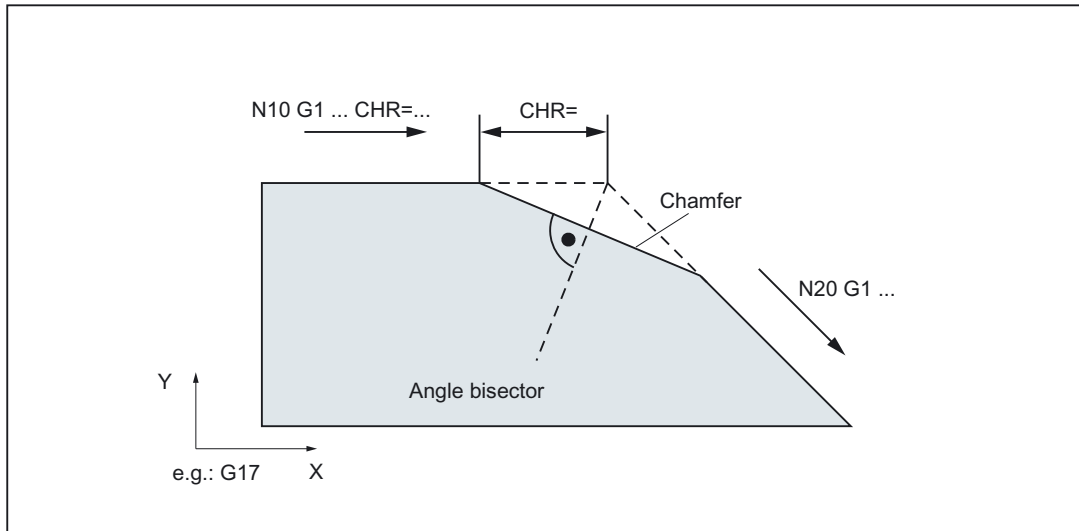


Figure 9-39 Inserting a chamfer with CHR using the example: Between two straight lines

## 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.

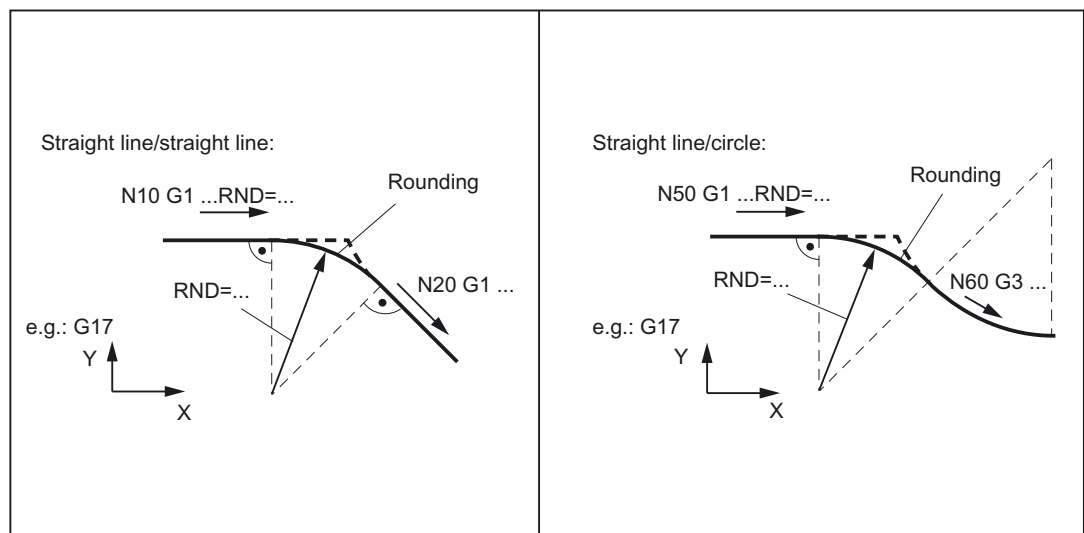


Figure 9-40 Examples for inserting roundings

## 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
...

```

## 9.5.2 Contour definition programming

### Functionality

If the end points for the contour are not directly specified in the machining drawing, it is also possible to use an angle specification `ANG=...` to determine the straight line. In a contour corner, you can insert the elements chamfer or rounding. The respective instruction `CHR= ...` or `RND=...` is written in the block, which leads to the corner.

The blueprint programming can be used in blocks with **G0** or **G1** (linear contours).

Theoretically, any number of straight line blocks can be connected and a rounding or a chamfer can be inserted between them. Every straight line must be clearly identified by point values and/or angle values.

### Programming

`ANG=...` ; Angle specification for defining a straight line  
`RND=...` ; Insert rounding, value: Radius of chamfer  
`CHR=...` ; Insert chamfer, value: Side length of the chamfer

### Information

The blueprint programming function is executed in the current plane G17 to G19. It is not possible to change the plane during blueprint programming.

If radius and chamfer are programmed in one block, only the radius is inserted regardless of the programming sequence.

### Angle ANG

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, an angle parameter can be used for uniquely defining the straight line path. The angle is always referred to the abscissa of the current plane G17 to G19, e.g. for G17 on the X axis. Positive angles are aligned counter-clockwise.

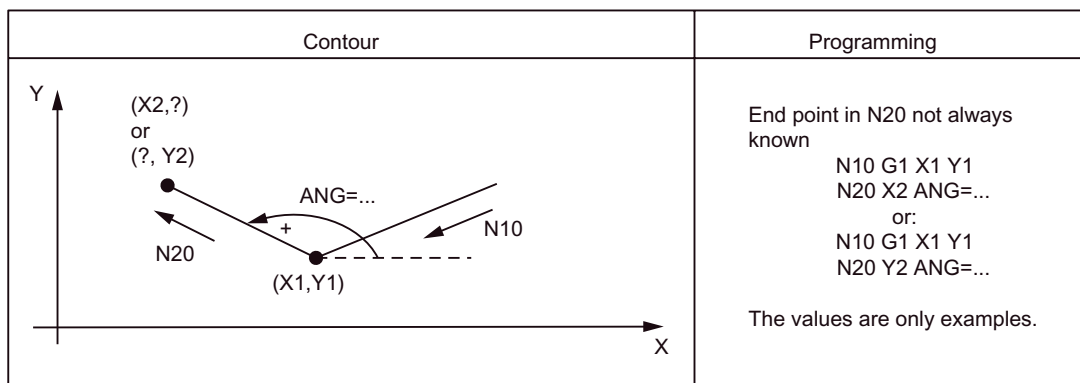


Figure 9-41 Specification of an angle for determination of a straight line using the example of the G17 plane

Contour	Programming
	<p>End point in N20 unknown  N10 G1 X1 Y1  N20 ANG=...1  N30 X3 Y3 ANG=...2</p> <p>The values are only examples.</p>
	<p>End point in N20 unknown,  insert rounding:  N10 G1 X1 Y1  N20 ANG=...1 RND=...  N30 X3 Y3 ANG=...2</p> <p>analog  Inserting a chamfer:  N10 G1 X1 Y1  N20 ANG=...1 CHR=...  N30 X3 Y3 ANG=...2</p>
	<p>End point in N20 known  Inserting a rounding:  N10 G1 X1 Y1  N20 X2 Y2 RND=...  N30 X3 Y3</p> <p>analog  Inserting a chamfer:  N10 G1 X1 Y1  N20 X2 Y2 CHR=...  N30 X3 Y3</p>
	<p>End point in N20 unknown  Inserting a rounding:  N10 G1 X1 Y1  N20 ANG=...1 RND=...1  N30 X3 Y3 ANG=...2 RND=...2  N40 X4 Y4</p> <p>analog  Inserting a chamfer:  N10 G1 X1 Y1  N20 ANG=...1 CHR=...1  N30 X3 Y3 ANG=...2  N40 X4 Y4  CHR=...2</p>

Figure 9-42 Multiple block contours using the example of the G17 plane

## 9.6 Tool and tool offset

### 9.6.1 General Information

#### Functionality

When creating programs for machining workpieces, it is not necessary to take into account the tool length or the tool radius. You program the workpiece dimensions directly, for example following the drawing.

You enter the tool data separately in a special data section.

Simply call the required tool with its offset data in the program and enable the tool radius compensation if necessary. The control system performs the required path compensations based on this data to create the described workpiece.

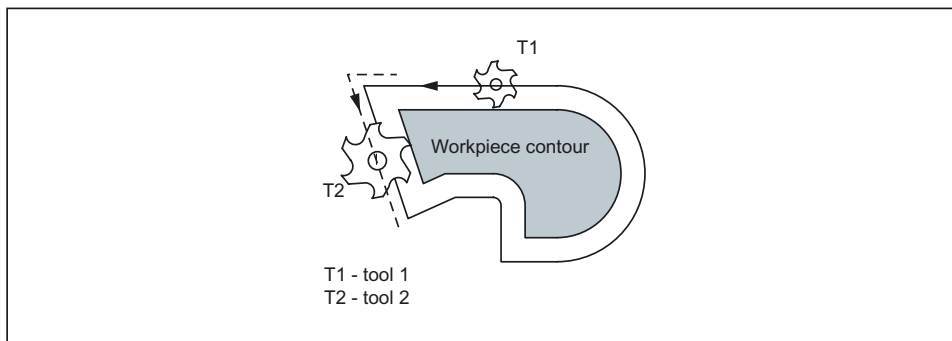


Figure 9-43 Machining of a workpiece with different tool radii

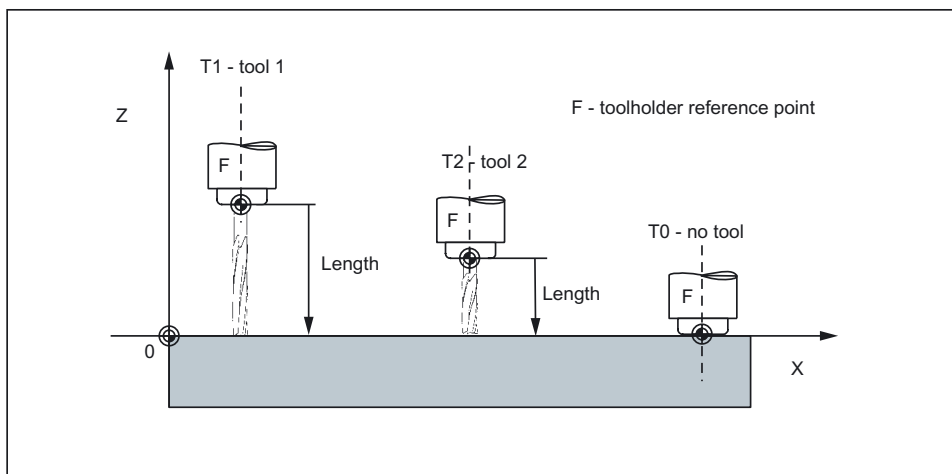


Figure 9-44 Approaching the workpiece position Z0 - different length compensations

#### See also

Entering tools and tool offsets (Page 36)



## 9.6.2 Tool T

### 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:

- The tool change (tool call) is performed either directly using the T word or
- The change takes place after the preselection with the T word by an additional instruction **M6** (see also Section 9.35 "Miscellaneous functions M").

---

#### Note

If a certain tool was activated, it remains stored as an active tool even beyond the end of the program and after turning off / turning 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, T0 - no tool

---

#### 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
; Tool change with M6:
N10 T14 ... ; Preselect tool 14
...
N15 M6 ; Perform tool change; thereafter, T14 is
active
```

### 9.6.3 Tool compensation number D

#### 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 tool is required, it can be programmed with D and the corresponding number.

If no D word is written, D1 takes effect **automatically**.

When **D0** is programmed, offsets for the tool have no **effect**.

---

#### Note

The following maximum values for tool compensation 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).
- 

#### Programming

D... ; Tool offset number: 1 ... 9,  
 D0: No compensations active!

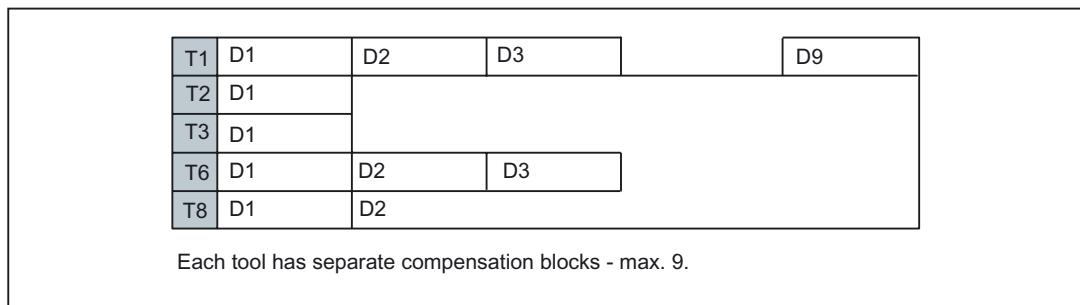


Figure 9-45 Examples for assigning tool compensation numbers / tool

#### Information

The **tool length compensations** are effective **immediately** once the tool is active - if no D number has been programmed - with the values of D1.

The offset is applied with the first programmed traverse of the respective length offset axis. Observe any active G17 to G19.

A **tool radius compensation** must also be activated by G41/G42.

## Programming example

Tool change **without M6 command** (only with T):

```
N5 G17                ; Determines the length offset axis (here Z axis)
N10 T1                ; Tool 1 is activated with the associated D1
N11 G0 Z...          ; For G17, Z is length offset axis, the length offset
                    ; compensation is overlaid here
N50 T4 D2            ; Load tool 4, D2 from T4 is active
...
N70 G0 Z... D1       ; D1 for tool 4 active, only cutting edge changed
```

Tool change using the **M6 command**:

```
N5 G17                ; Determines the length offset axis (here Z axis)
N10 T1                ; Tool preselection
...
N15 M6               ; Tool change, T1 is active with the appropriate D1
N16 G0 Z...          ; For G17, Z is length offset axis, the length offset
                    ; compensation is overlaid here
...
N20 G0 Z... D2       ; D2 for tool 1 is active; for G17, Z is length offset axis,
                    ; the difference of the D1->D2 length offset is overlaid here
N50 T4                ; T4 tool preselection; note: T1 with D2 is still active !
...
N55 D3 M6            ; Tool change, T4 is active with the appropriate D3
...
```

## Contents of a compensation memory

Enter the following in the offset memory:

- Geometrical dimensions: length, radius.

They consist of several components (geometry, wear). The control computes the components to a certain dimension (e.g. overall length 1, total radius). The respective overall dimension becomes effective when the compensation memory is activated.

How these values are calculated in the axes is determined by the tool type and the commands G17, G18, G19 (see following illustrations).

- Tool type

The tool type (drill, cutter) defines which geometry data are necessary and how they are taken into account.

**Tool special cases**

For the tool types 'cutter' and 'drill', the parameters for length 2 and length 3 are only required for special cases (e.g. multi-dimensional length offset for an angle head construction).

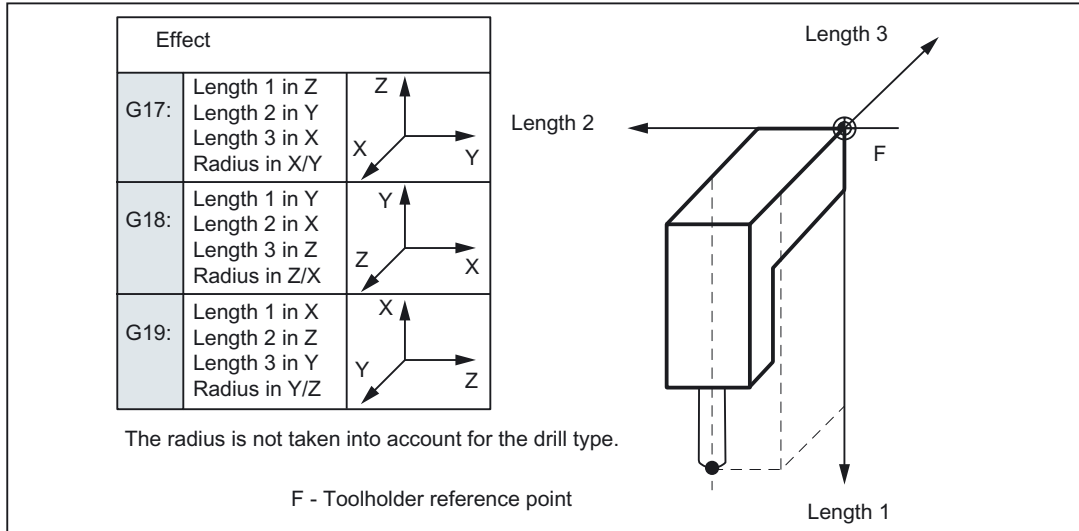


Figure 9-46 Effect of the tool length compensation - 3D (special case)

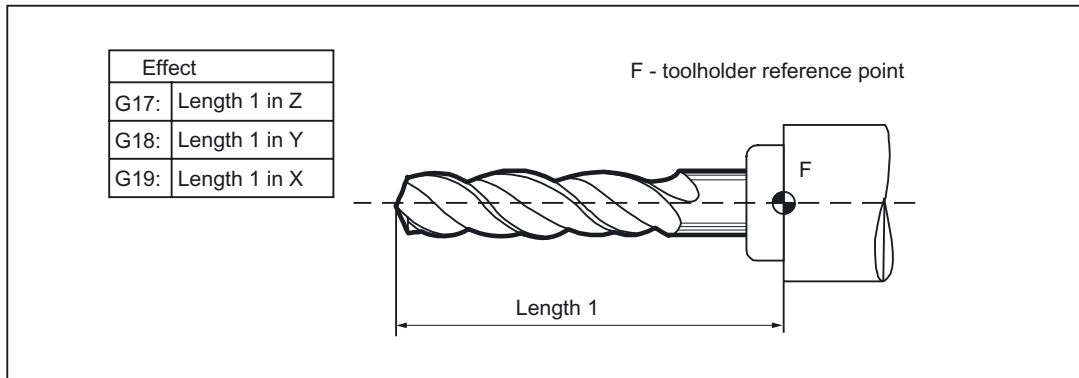


Figure 9-47 Effect of the offsets with the tool type 'drill'

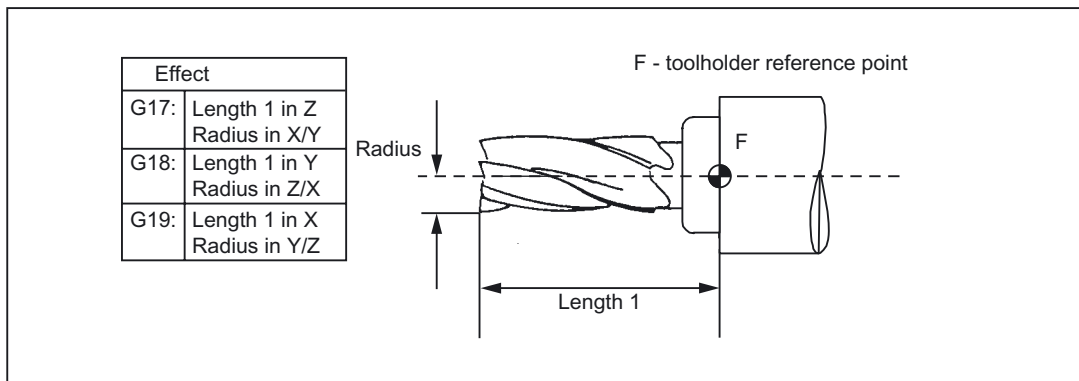


Figure 9-48 Effect of the offsets with the tool type 'cutter'

## 9.6.4 Selecting the tool radius compensation: G41, G42

### Functionality

The control system is working with tool radius compensation in the selected plane G17 to G19.

A tool with a corresponding D number must be active. The tool radius compensation is activated by G41/G42. The control system automatically calculates the required equidistant tool paths for the programmed contour for the respective current tool radius.

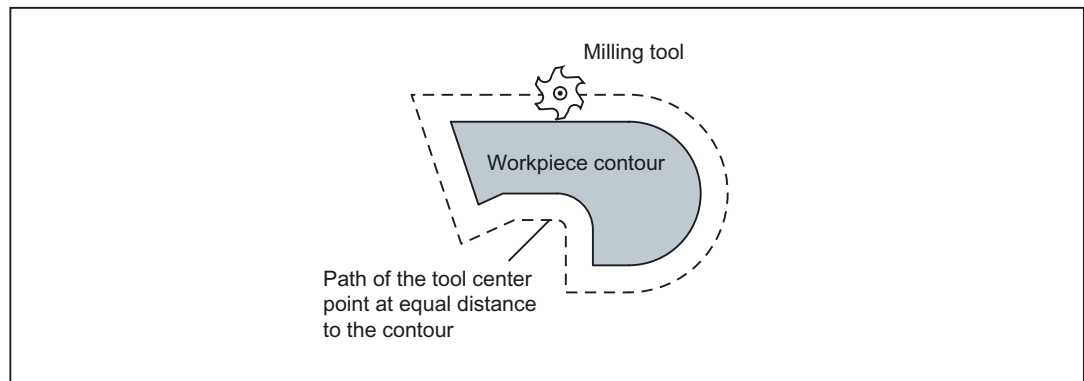


Figure 9-49 Tool radius compensation

### Programming

G41 X... Y... ; Tool radius compensation left of contour  
G42 X... Y... ; Tool radius compensation right of contour

---

#### Note

The selection can only be made for linear interpolation (G0, G1).

Program both axes of the plane (e.g. with G17: X, Y). If you only specify one axis, the second axis is automatically completed with the last programmed value.

---

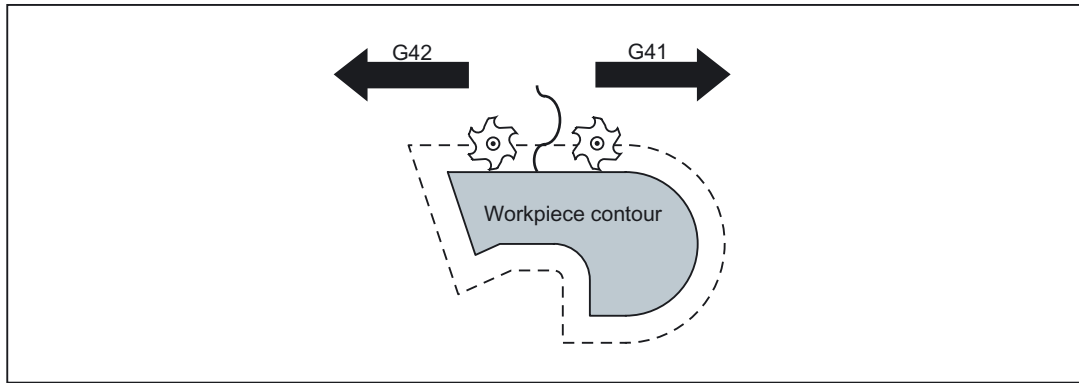


Figure 9-50 Compensation to the right/left of the contour

**Starting the compensation**

The tool travels in a straight line directly to the contour and is positioned perpendicular to the path tangent at the starting point of the contour.

Select the starting point such that a collision-free travel is ensured.

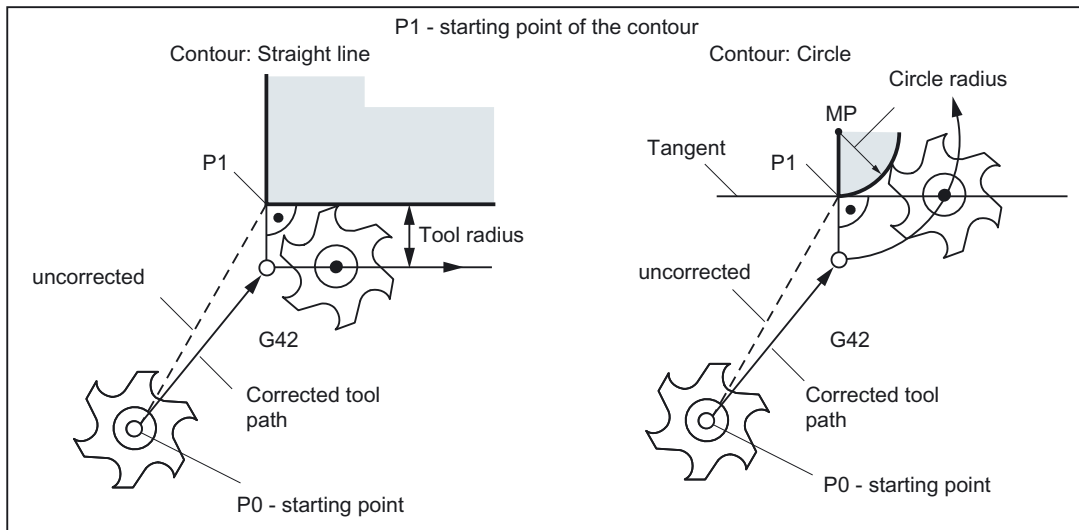


Figure 9-51 Start of the tool radius compensation with G42 as example

**Information**

As a rule, the block with G41/G42 is followed by the block with the workpiece contour. The contour description, however, may be interrupted by 5 blocks which lie between them and do not contain any specifications for the contour path in the plane, e.g. only an M command or infeed motions.

## Programming example

```
N10 T...  
N20 G17 D2 F300 ; Correction number 2, feed 300 mm/min  
N25 X... Y... ; P0 - starting point  
N30 G1 G42 X... Y... ; Selection right of contour, P1  
N31 X... Y... ; Starting contour, circle or straight line
```

After the selection, it is also possible to execute blocks that contain infeed motions or M outputs:

```
N20 G1 G41 X... Y... ; Selection to the left of the contour  
N21 Z... ; Infeed movement  
N22 X... Y... ; Starting contour, circle or straight line
```

### 9.6.5 Corner behavior: G450, G451

#### Functionality

By using the functions G450 and G451, you can set the behavior for a 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 itself. For internal corners, the intersection of the equidistant paths is always approached.

#### Programming

```
G450 ; Transition circle  
G451 ; Point of intersection
```

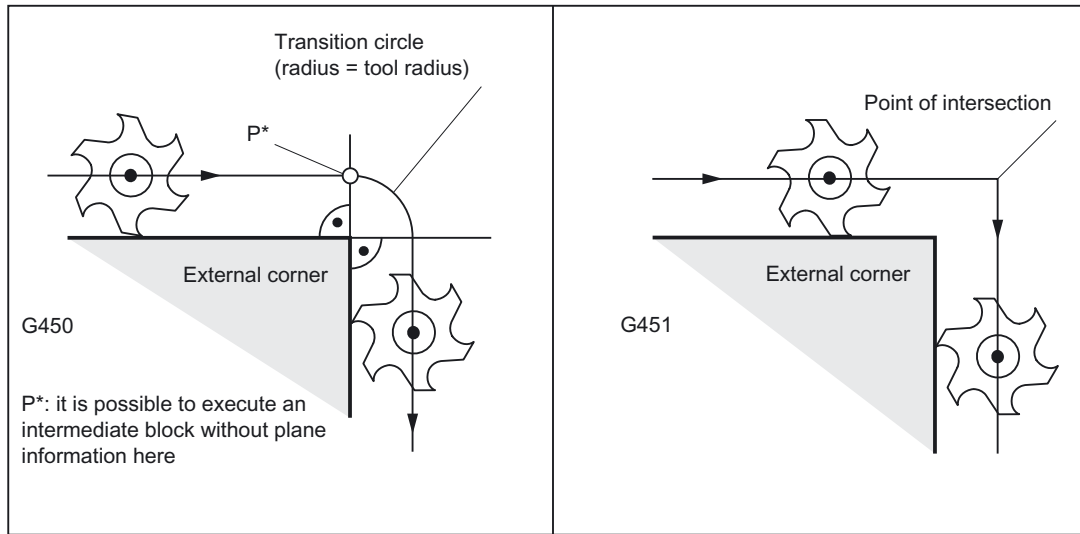


Figure 9-52 Corner behavior at an external corner

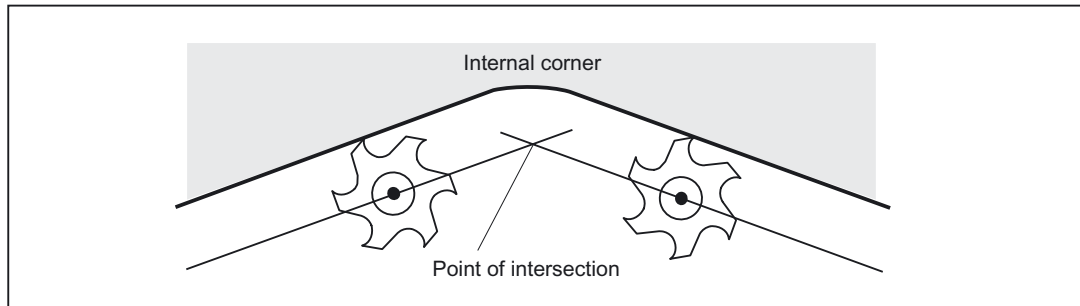


Figure 9-53 Corner behavior at an internal corner

**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.

With acute contour angles and active point of intersection, depending on the tool radius, unnecessary idle motions could result for the tool.

In this case, the control system automatically switches to transition circle for this block if a certain set angle value (100°) is reached.



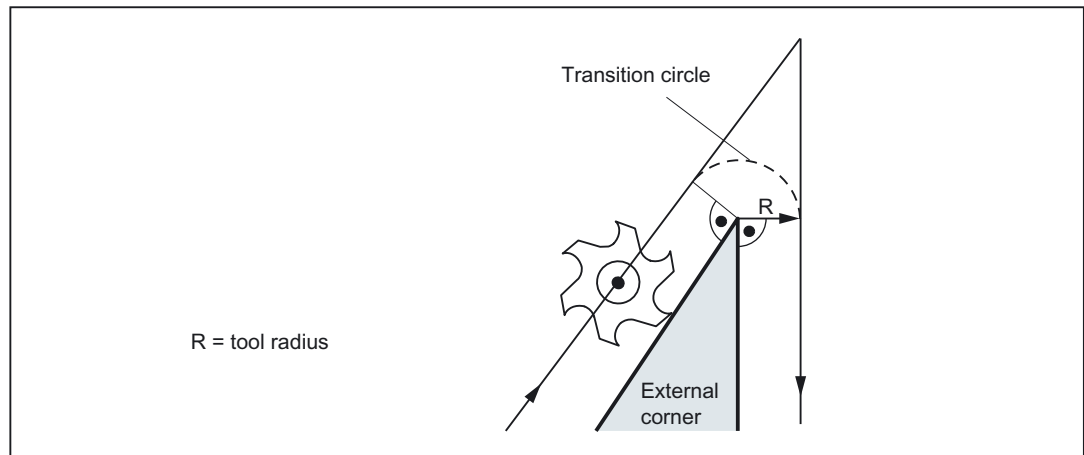


Figure 9-54 Acute contour angle and switching to transition circle

## 9.6.6 Tool radius compensation OFF: G40

### Functionality

The compensation mode (G41/G42) is deselected with G40. G40 is also the activation position at the beginning of the program.

The tool ends the **block in front of G40** in the normal position (compensation vector vertically to the tangent at the end point);

If G40 is active, the reference point is the tool center point. Subsequently, when deselected, the tool tip approaches the programmed point.

Always select the end point of the G40 block such that collision-free traversing is guaranteed!

### Programming

G40 X... Y... ; Tool radius compensation OFF

---

#### Note

The compensation mode can only be deselected with linear interpolation (G0, G1).

Program both axes of the plane (e.g. with G17: X, Y). If you only specify one axis, the second axis is automatically completed with the last programmed value.

---

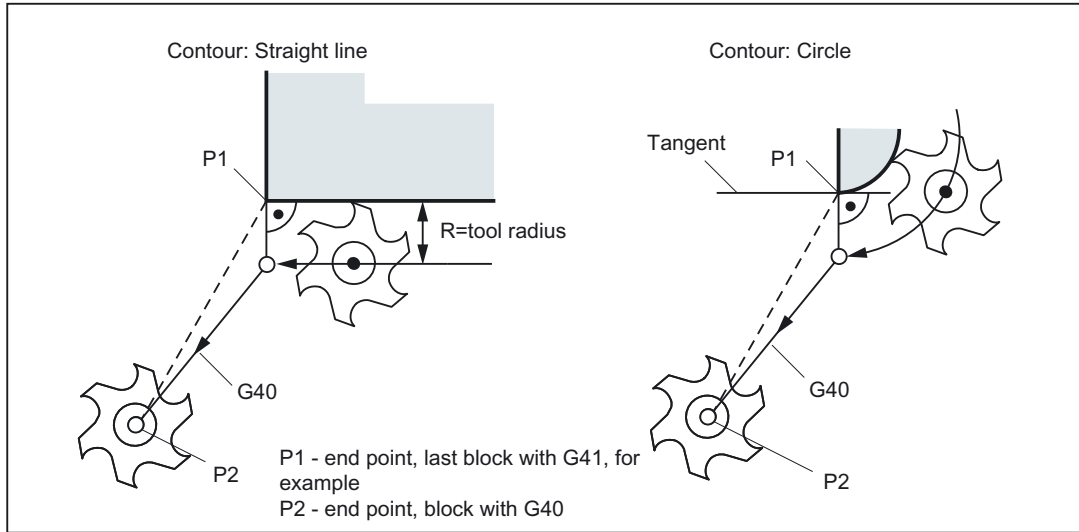


Figure 9-55 Quitting the tool radius compensation

### Programming example

```
...  
N100 X... Y... ; Last block on the contour, circle or straight line,  
P1  
N110 G40 G1 X... Y.. ; Switch off tool radius compensation, P2
```

## 9.6.7 Special cases of the tool radius compensation

### Repetition of the compensation

The same compensation (e.g. G41 -> G41) can be programmed once more without writing G40 between these commands.

The last block in front of the new compensation call ends with the normal position of the compensation vector at 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

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.

### Change of the compensation direction

The compensation direction G41 <-> G42 can be changed without writing G40.

The last block with the old compensation direction ends with the normal position of the compensation vector at the end point. The new compensation direction is executed as a compensation start (default setting at starting point).

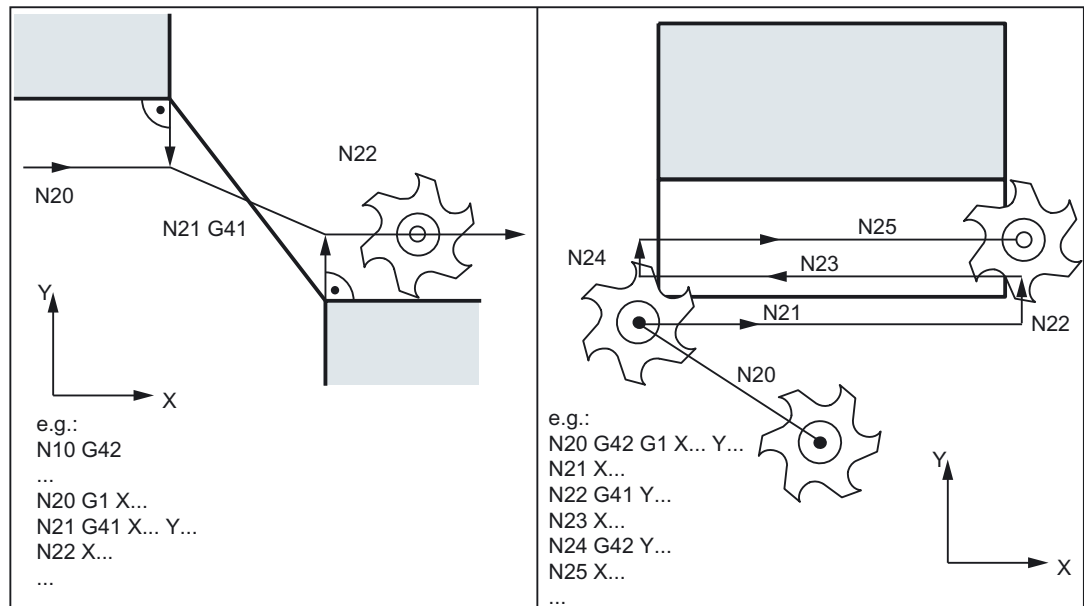


Figure 9-56 Change of the compensation direction

### Cancellation of compensation by M2

If compensation mode is canceled using M2 (end of program) without writing the command G40, the last block with coordinates of the plane (G17 to G19) will end in the normal position

of the compensation vector. **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 travel is smaller than the tool radius;

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 prevents long idle motions (see figure "Acute contour angle and switching to transition circle", Section "Corner behavior: G450, G451").

## 9.6.8 Example of tool radius compensation

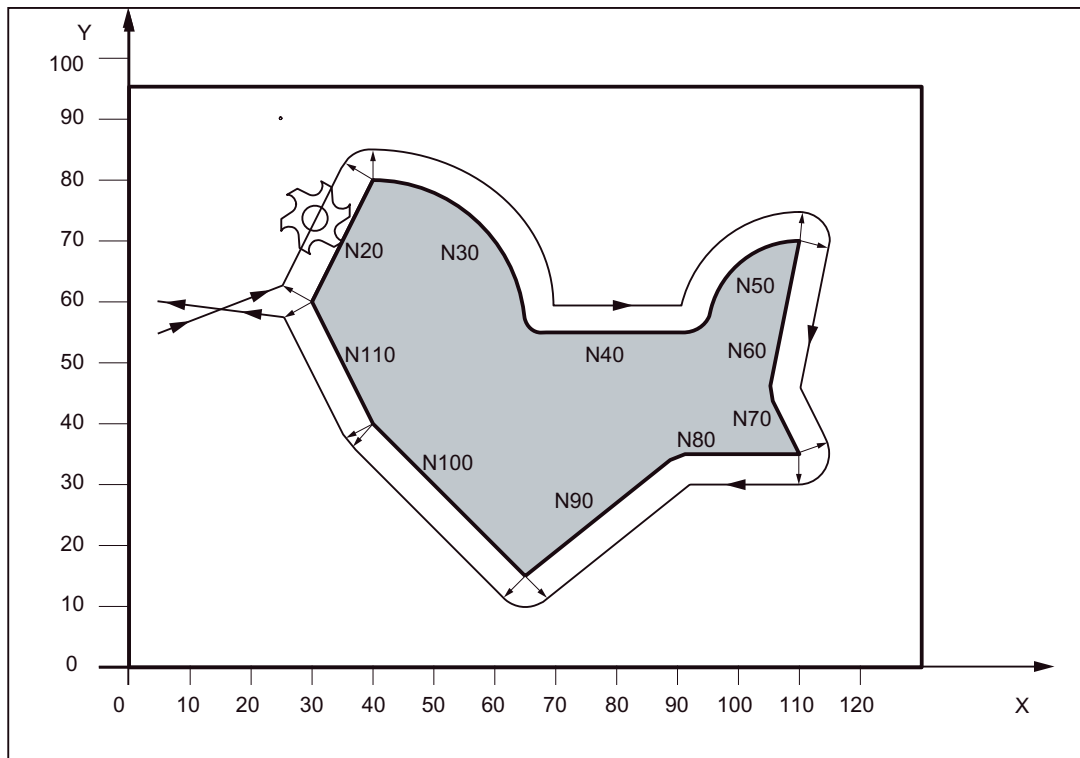


Figure 9-57 Example of tool radius compensation

## Programming example

```
N1 T1 ; Tool 1 with offset D1
N5 G0 G17 G90 X5 Y55 Z50 ; Approach starting point
N6 G1 Z0 F200 S80 M3
N10 G41 G450 X30 Y60 F400 ; Compensation to the left of the contour,
transition circle
N20 X40 Y80
N30 G2 X65 Y55 I0 J-25
N40 G1 X95
N50 G2 X110 Y70 I15 J0
N60 G1 X105 Y45
N70 X110 Y35
N80 X90
N90 X65 Y15
N100 X40 Y40
N110 X30 Y60
N120 G40 X5 Y60 ; Terminate compensation mode
N130 G0 Z50 M2
```

## 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:

$\pm(0.000\ 0001 \dots 9999\ 9999)$   
(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:

$\pm (10^{-300} \dots 10^{+300})$

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] varname41	; STRING type, [string length]: Maximum number of 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:

<code>\$A_DBB[n]</code>	;Data byte (8-bit value)
<code>\$A_DBW[n]</code>	;Data word (16-bit value)
<code>\$A_DBD[n]</code>	;Data double-word (32-bit value)
<code>\$A_DBR[n]</code>	;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

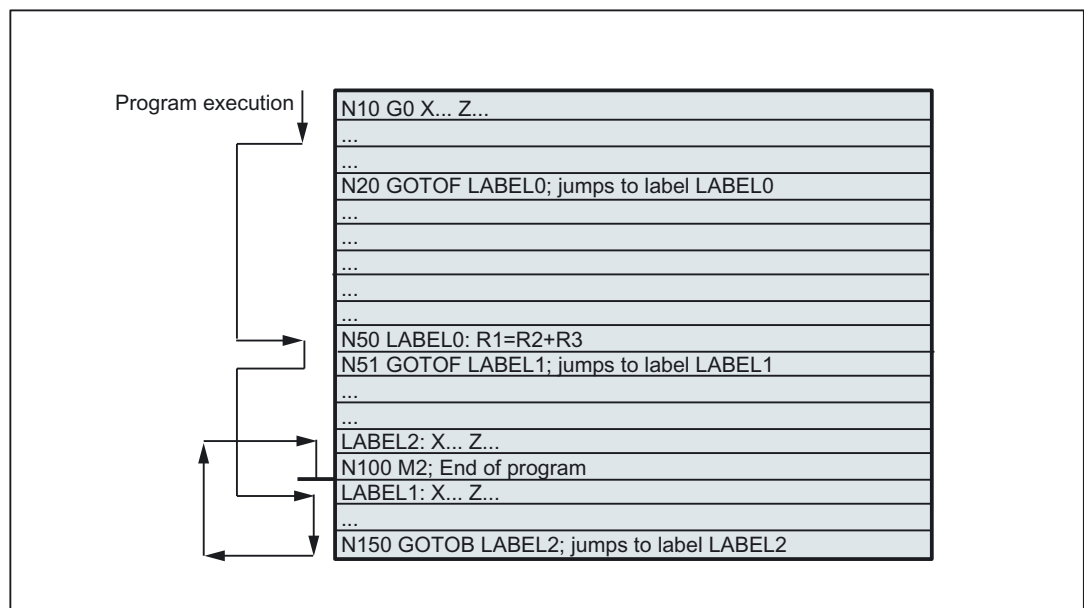


Figure 9-58 Unconditional jumps using an example

### 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

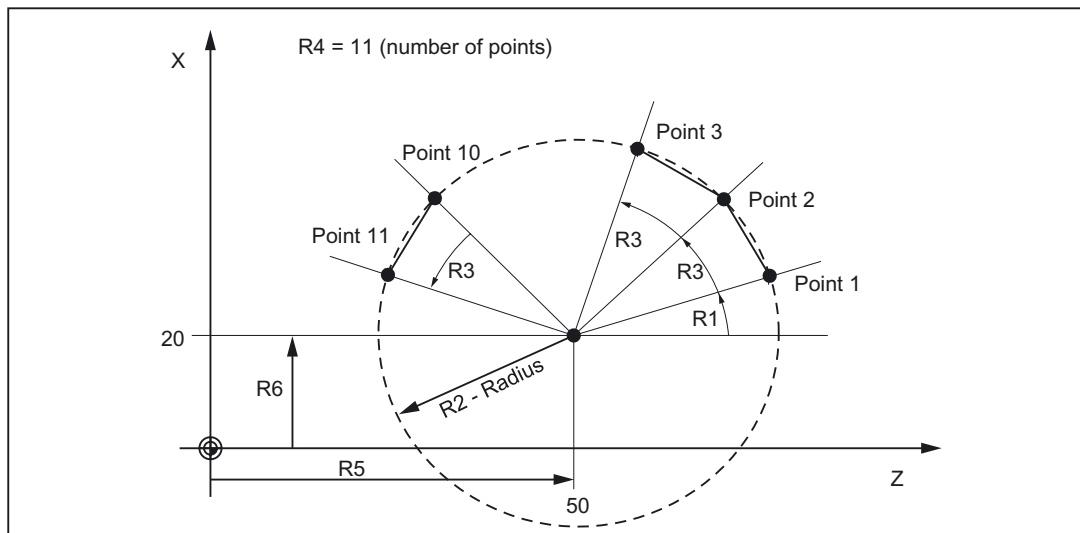


Figure 9-59 Linear approach of points on a circle segment

#### 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

#### Application

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**. The machining cycles contain generally valid machining cases (e.g. drilling, tapping, groove cutting, etc.). By assigning values via included transfer parameters, you can adapt the subroutine to your specific application.

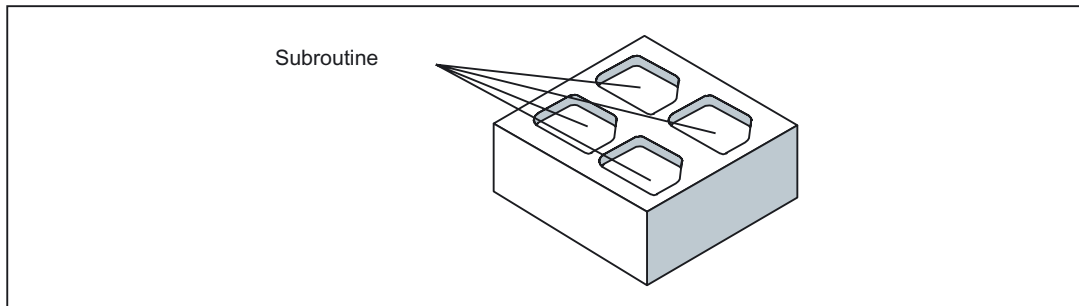


Figure 9-60 Example for using a subroutine for a workpiece four times

#### Set-up

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.

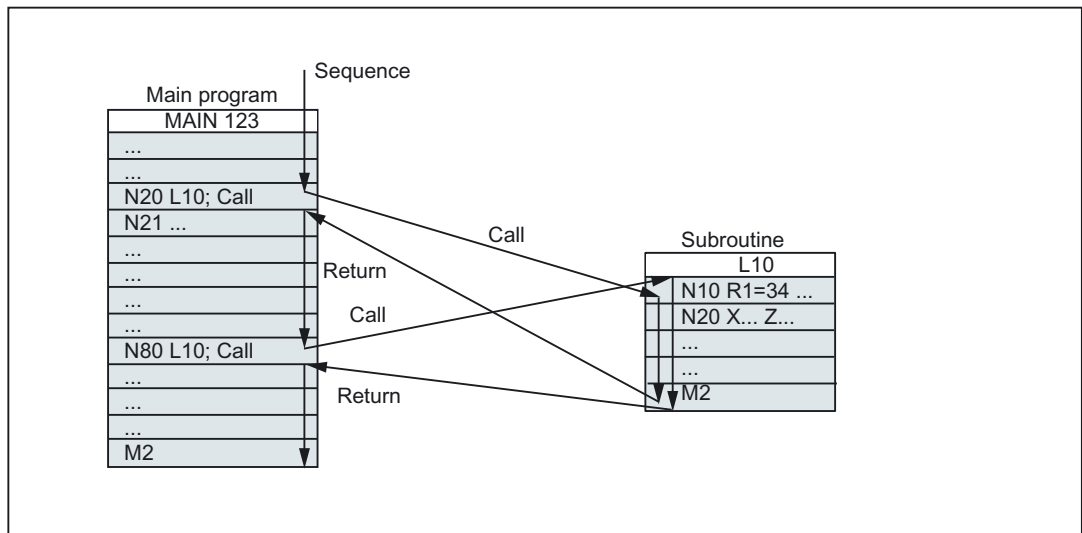


Figure 9-61 Example of sequence when calling a subroutine twice

## Subroutine name

The program 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: **LRAHMEN7**

It is also possible to use the address word **L...** in subroutines. The value can have 7 decimal places (integers only).

---

### Note

With address L, leading zeros are meaningful for differentiation.

---

Example: **L128** is not **L0128** or **L00128**.

These are three different subroutines.

---

### Note

The subroutine name **LL6** is reserved for tool change.

---

### Subroutine call

Subroutines are called in a program (main or subroutine) with their names. To do this, a separate block is required.

**Example:**

```
N10 L785 ; Call subroutine L785  
N20 LRAHMEN7 ; Call subroutine LRAHMEN7
```

### 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 ; Call subroutine 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.

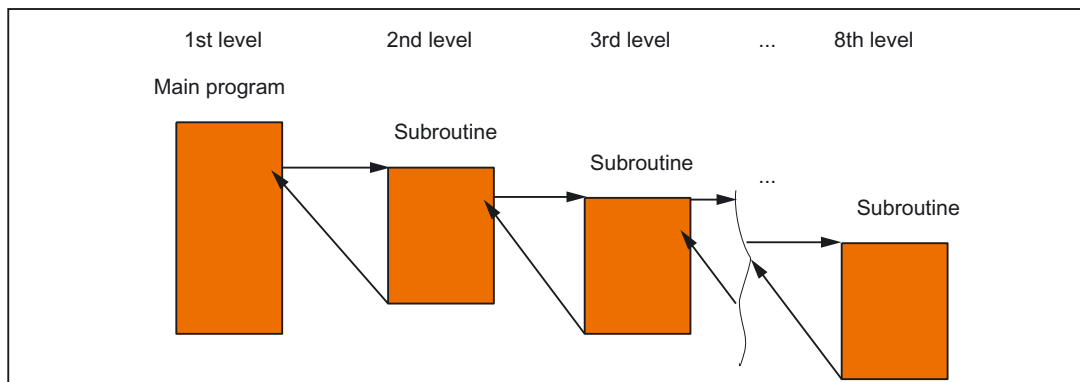


Figure 9-62 Execution with 8 program levels

### 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 4 program levels are needed.

## 9.11.2 Calling machining cycles

### Functionality

Cycles are technology subroutines realizing a certain machining process generally, for example, drilling or milling. Adaptation to the particular problem is performed 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 Modal subroutine call

### Functionality

The subroutine in the block containing MCALL is called automatically after each successive block containing a **path motion**. The call acts until the next MCALL is called.

The modal call of the subroutine which contains MCALL or quitting of the call requires a separate block.

MCALL is advantageous, for example, when producing drill patterns.

### Programming example

Application example: Drilling a row of holes

```
N10 MCALL CYCLE82(...)        ; Drilling cycle 82
N20 HOLES1(...)              ; Cycle for row of holes; after each approach of the
                               ; hole position, CYCLE82(...) will be called with the
                               ; transfer parameters
N30 MCALL                    ; Modal call of CYCLE82(...) completed
```

### 9.11.4 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

<b>NOTICE</b>
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 GOTO, 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.

Type	T	D <sub>Σ</sub>	Tool life [min]			Quantity		
			Setpt.	Prev.lt	Resid. Activ	Setpt.	Prev.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- 4 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: Quantitywithout 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	0.000	<input checked="" type="checkbox"/>	10	9	0	<input type="checkbox"/>
2	1			0.100	0.000	0.100	<input type="checkbox"/>	0	0	0	<input type="checkbox"/>
3	1			0.100	0.000	0.000	<input checked="" type="checkbox"/>	0	0	0	<input type="checkbox"/>
4	1			0.000	0.000	0.000	<input type="checkbox"/>	0	0	0	<input type="checkbox"/>
5	1			0.000	0.000	0.000	<input type="checkbox"/>	0	0	0	<input type="checkbox"/>
6	1			0.000	0.000	0.000	<input type="checkbox"/>	0	0	0	<input type="checkbox"/>
7	1			0.000	0.000	0.000	<input type="checkbox"/>	0	0	0	<input type="checkbox"/>
8	1			0.000	0.000	0.000	<input type="checkbox"/>	0	0	0	<input type="checkbox"/>
9	1			0.000	0.000	0.000	<input type="checkbox"/>	0	0	0	<input type="checkbox"/>
10	4			0.000	0.000	0.000	<input type="checkbox"/>	0	0	0	<input type="checkbox"/>
11	1			0.000	0.000	0.000	<input type="checkbox"/>	0	0	0	<input type="checkbox"/>

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 Smooth approach and retraction

### Functionality

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

The function "Smooth approach and retraction" (SPR) is intended to approach the beginning of a contour tangentially ("smooth"), to a large degree independently of the position of the starting point. The control system will calculate the intermediate points and will generate the required traversing blocks. This function is used preferably in conjunction with the tool radius compensation (TRC). The G41 and G42 commands determine the approach/retraction direction to the left or right of the contour (see also Section "Selecting the tool radius compensation: G41, G42").

The approach/retraction path (straight line, quarter or semi-circle) is selected using a group of G commands. To parameterize this path (circle radius, length, approach straight line), special addresses can be used; this also applies to the feedrate of the infeed motion. The infeed motion can additionally be controlled via another G group.

### Programming

G147	; Approach with a straight line
G148	; Retraction with a straight line
G247	; Approach with a quadrant
G248	; Retraction with a quadrant
G347	; Approach with a semi-circle
G348	; Retraction with a semi-circle
G340	; Approach and retraction in space (basic setting)
G341	; Approach and retraction in the plane
DISR=...	; Approach and retraction with straight lines (G147/G148): Distance of the cutter edge from the start or end point of the contour ; Approach and retraction along circles (G247, G347/G248, G348): Radius of the tool center point path
DISCL=...	; Distance of the end point for the fast infeed motion from the machining plane (safety clearance)
FAD=...	; Speed of the slow infeed motion The programmed value acts according to the active command of the G group 15 (feed: G94, G95)

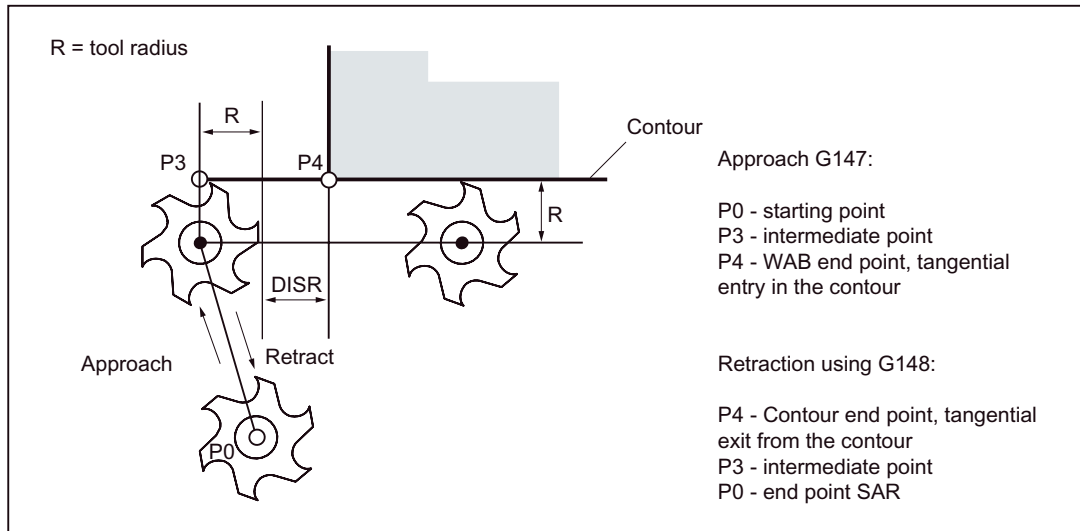


Figure 9-65 Approach along a straight line using the example of G42 or retraction using G41 and completion with G40.

**Programming example: Approach/retraction along a straight line in a plane**

```

N10 T1 ... G17 ; Activate tool, X/Y plane
N20 G0 X... Y... ; Approach P0
N30 G42 G147 DISR=8 F600 X4 Y4 ; Approach, point P4 programmed
N40 G1 X40 ; Continue in the contour
...
N100 G41 ...
N110 X4 Y4 ; P4 - contour end point
N120 G40 G148 DISR=8 F700 X... Y... ; Retraction, point P0 programmed
...
    
```

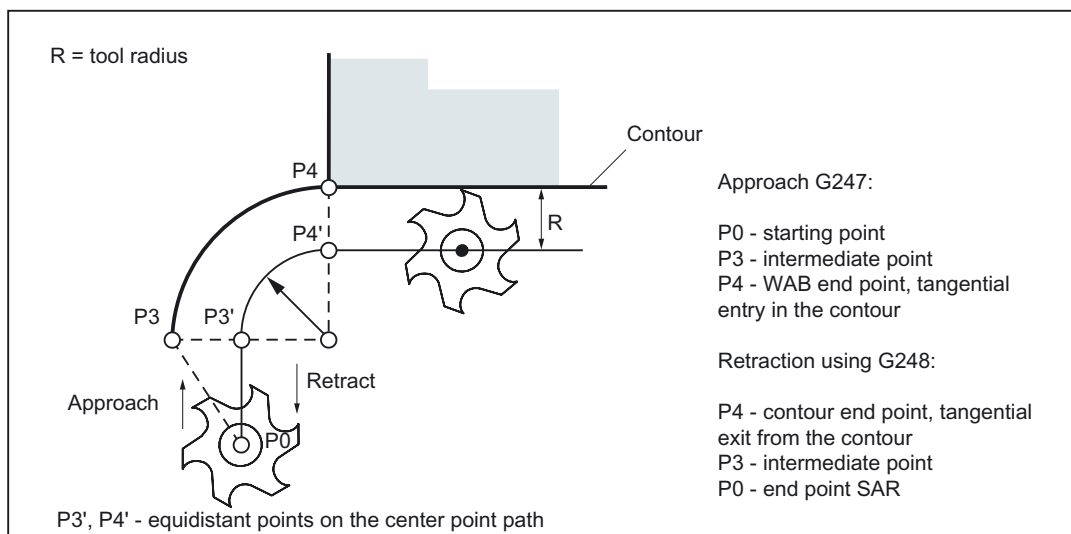


Figure 9-66 Approach along a quadrant using the example of G42 or retraction using G41 and completion with G40.

### Programming example: Approach/retraction along a quarter in a plane

```

N10 T1 ... G17 ; Activate tool, X/Y plane
N20 G0 X... Y... ; Approach P0
N30 G42 G247 DISR=20 F600 X4 Y4 ; Approach, point P4 programmed
N40 G1 X40 ; Continue in the contour
...
N100 G41 ...
N110 X4 Y4 ; P4 - contour end point
N120 G40 G248 DISR=20 F700 X... Y... ; Retraction, point P0 programmed
...
    
```

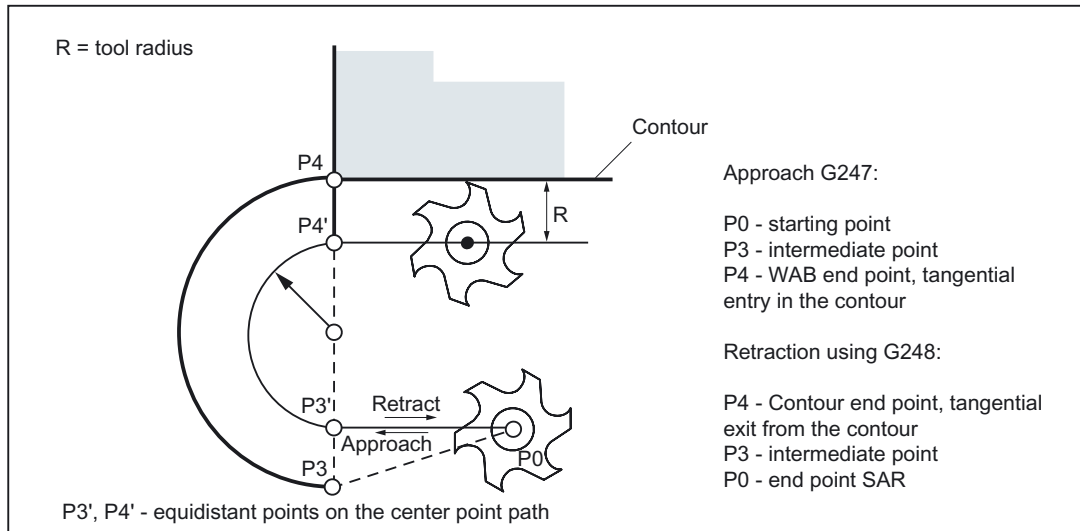


Figure 9-67 Approach along a semi-circle using the example of G42 or retraction using G41 and completion with G40.

**Note**

Make sure that a positive radius is entered for the tool radius. Otherwise, the directions for G41, G42 will be changed.

**Controlling the infeed motion using DISCL and G340, G341**

DISCL=... specifies the distance of point P2 from the machining plane (see following figure).

In the case DISCL=0, the following will apply:

- With G340: The whole approach motion consists only of two blocks (P1, P2 and P3 are identical). The approach contour is generated from P3 to P4.
- With G341: The whole approach motion consists only of three blocks (P2 and P3 are identical). If P0 and P4 are located in the same plane, only two blocks will result (there will be no infeed motion from P1 to P3).

It is monitored that the point defined by DISCL lies between P1 and P3, i.e. with all motions that possess a component which runs vertically to the machining plane, this component must have the same sign. If a reversal of the direction is detected, a tolerance of 0.01 mm is permitted.

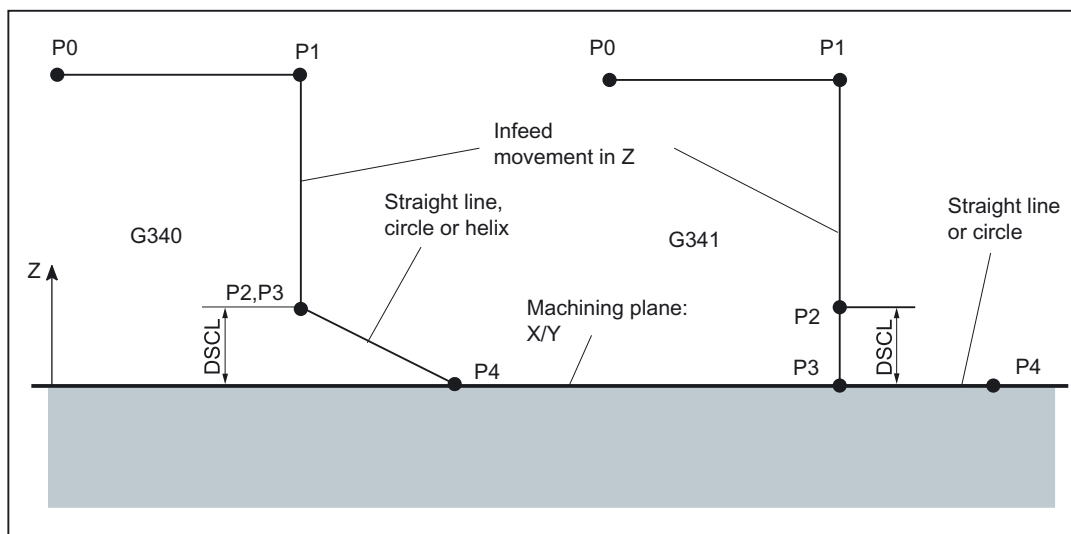


Figure 9-68 Sequence of the approach motion dependent on G340 / G341 (example with G17)

### Programming example: Approach along a semi-circle with infeed

```

N10 T1 ... G17 G90 G94           ; Activate tool, X/Y plane
N20 G0 X0 Y0 Z30                ; Approach P0
N30 G41 G347 G340 DISCL=3 DISR=13 Z=0
F500                             ; Approach along a semi-circle with
                                ; radius: 13 mm,
                                ; Safety clearance to the plane: 3 mm
N40 G1 X40 Y-10
...

```

alternatively N30 / N40:

```
N30 G41 G347 G340 DISCL=3 DISR=13 X40 Y-10 Z0 F500
```

or

```
N30 G41 G347 G340 DISCL=3 DISR=13 F500
```

```
N40 G1 X40 Y-10 Z0
```

Explanation with regard to N30 / N40:

By using G0 (from N20), the point P1 (starting point of the semi-circle, corrected by the tool radius) is approached in the plane Z=30, then lowering to the depth (P2, P3) with Z=3 (DISCL). The contour is reached at point X40 Y-10 in the depth Z=0 (P4) along a helix curve at a feedrate of 500 mm/min.

### Approach and retraction velocities

- Velocity of the previous block (e.g. G0):  
All motions from P0 up to P2 are executed at this speed, i.e. the motion parallel to the machining plane and the part of the infeed motion up to the safety clearance DISCL.
- Programmed feedrate F:  
This feedrate is active from P3 or P2 if FAD is not programmed. If no F word is programmed in the SAR block, the velocity of the previous block will act.
- Programming using FAD:  
Specify the feedrate for
  - G341: Infeed motion vertically to the machining plane from P2 to P3
  - G340: from point P2 or P3 to P4
 If FAD is not programmed, this part of the contour is traversed at the speed which is active modally from the preceding block, in the event that no F command defining the speed is programmed in the SAR block.
- **During retraction**, the roles of the modally effective feedrate from the previous block and the feedrate programmed in the SAR block are changed, i.e. the actual retraction contour is traversed using the old feedrate, and a new velocity programmed using the F word will apply correspondingly from P2 to P0.

### Programming example: Approach along a quadrant, infeed using G341 and FAD

```

N10 T1 ... G17 G90 G94                ; Activate tool, X/Y plane
N20 G0 X0 Y0 Z30                      ; Approach P0
N30 G41 G341 G247 DISCL=5 DISR=13 FAD=500 X40 Y-10 Z=0 F800
N40 G1 X50
...

```

Explanation with regard to N30:

By using G0 (from N20), the point P1 (starting point of the quadrant, corrected by the tool radius) is approached in the plane Z=30, then lowering to the depth (P2) with Z=5 (DISCL). Using a feedrate of FAD=500 mm/min, it is lowered to a depth of Z=0 (P3) (G341). Then, the contour is approached at point X40,Y-10 along a quadrant in the plane (P4) using F=800 mm/min.

### Intermediate blocks

A maximum of five blocks **without** moving the geometry axes can be inserted between an SAR block and the next traversing block.

## Information

Programming when retracting:

- With an SAR block with a geometry axis programmed, the contour ends at P2. The positions on the axes that constitute the machining plane result from the retraction contour. The axis component perpendicular to this is defined by DISCL. With DISCL=0, the motion will run completely in the plane.
- If in the SAR block only the axis is programmed vertically to the machining plane, the contour will end at P1. The positions of the remaining axes will result, as described above. If the SAR block is also the TRC disable block, an additional path from P1 to P0 is inserted such that no motion results at the end of the contour when disabling the TRC.
- If only one axis on the machining plane is programmed, the missing second axis is modally added from its last position in the previous block.

## 9.15 Milling of the peripheral surface - TRACYL

### Functionality

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

- 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.

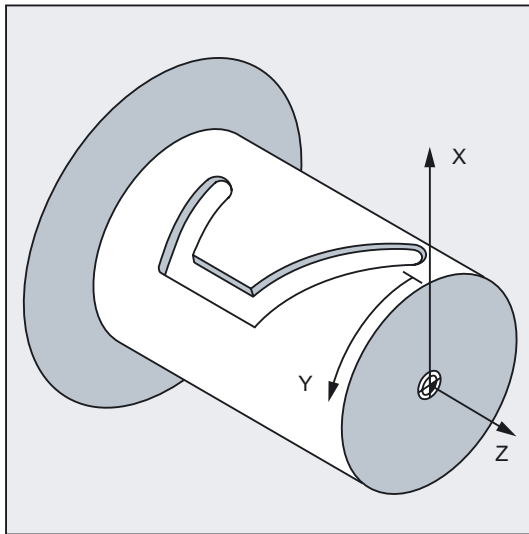


Figure 9-69 Cartesian coordinate system X, Y, Z when programming TRACYL

- The control system transforms the programmed traversing motions in the Cartesian coordinate system X, Y, Z into motions of the real machine axes. A rotary axis (rotary table) is required.



- TRACYL must be configured using special machine data. The rotary axis position at which the value Y=0 is also defined here.
- Milling machines possess a real machine Y axis (YM). An extended TRACYL variant can be configured here. 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.

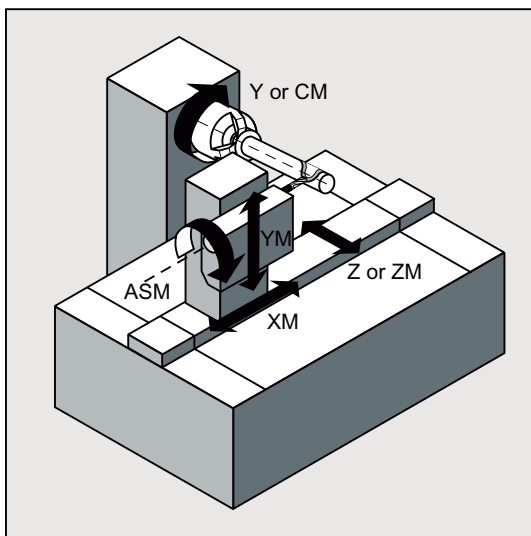


Figure 9-70 Machine kinematics with a machine Y axis (YM)

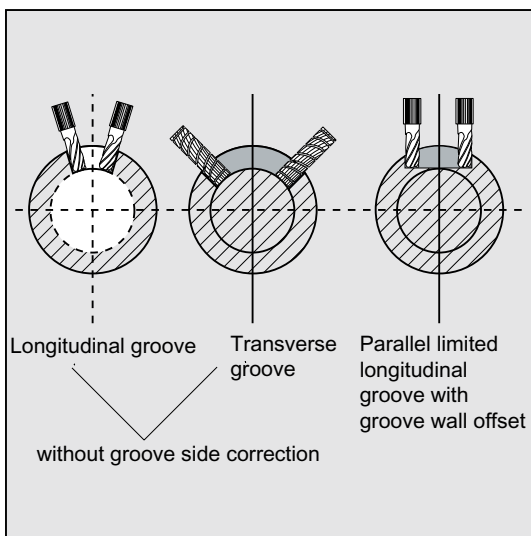


Figure 9-71 Various grooves in cross section

## Programming

TRACYL(d) ; Activate TRACYL (separate block)  
 TRAFOOF ; Deactivate (separate block)  
 d - machining diameter of the cylinder in mm

TRAFOOF deactivates any active transformation function.

### OFFN address

Distance of the groove side to the programmed path

As a rule, the groove center line is 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.

---

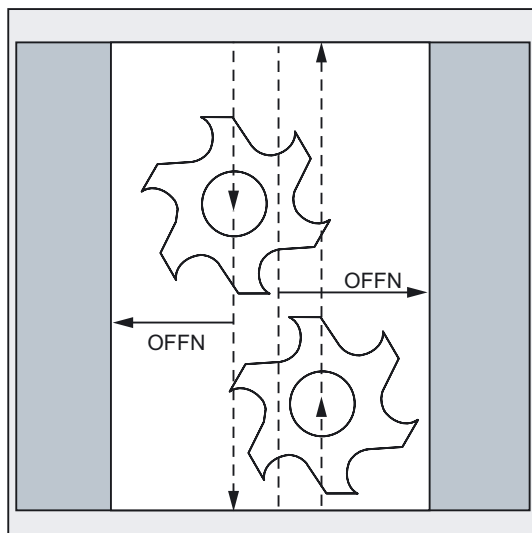


Figure 9-72 Use of OFFN for the groove width

**Note****Programming**

In order to mill with TRACYL, the groove center line is programmed in the part program with the specified coordinates and the (half) groove width is programmed with OFFN.

OFFN does not go into effect until tool radius compensation is selected. Furthermore, OFFN should be  $\geq$  the tool radius to avoid damage to the opposite groove side.

A part program for milling a groove generally comprises the following steps:

1. Select a tool
  2. Select TRACYL
  3. Select an appropriate work offset
  4. Positioning
  5. Program OFFN
  6. Select TRC
  7. Approach block (position TRC and approach groove side)
  8. Program the groove course using the groove center line
  9. Deselect TRC
  10. Retraction block (retract TRC and move away from groove side)
  11. Positioning
  12. Clear OFFN
  13. TRAFOOF (deselect TRACYL)
  14. Reselect the original work offset  
(see also following programming example)
- 

**Information**

- Guiding grooves:

By using a tool diameter that corresponds exactly to the groove width, it is possible to produce an exact groove. The tool radius compensation is not activated for this.

With TRACYL, grooves can also be produced, whose tool diameter is smaller than the groove width. For this, the tool radius compensation (G41, GG42) and OFFN are used.

To avoid accuracy problems, the tool diameter should only be slightly smaller than the groove width.
- When working with TRACYL with groove side correction, the axis used for the correction (YM) should stand on the turning center of the rotary axis. Thus, the groove is created centered on the programmed groove center line.
- Selecting the tool radius compensation (TRC) :

The TRC acts for the programmed groove center line. 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 $\leftrightarrow$ 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.

References

SINUMERIK 802D sl Function Manual, Kinematic Transformations

Programming example

Making a hook-shaped groove

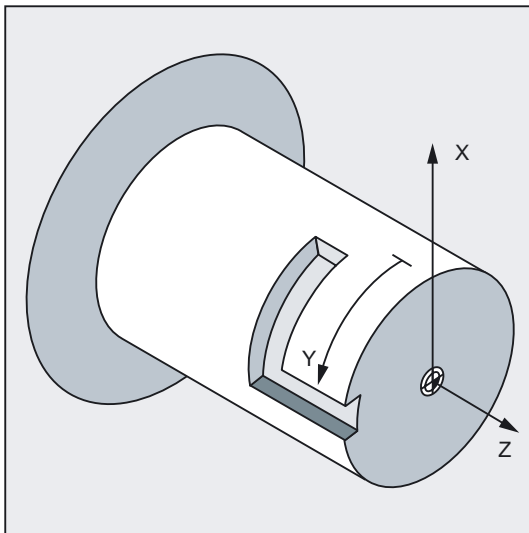


Figure 9-73 Producing a groove (example)

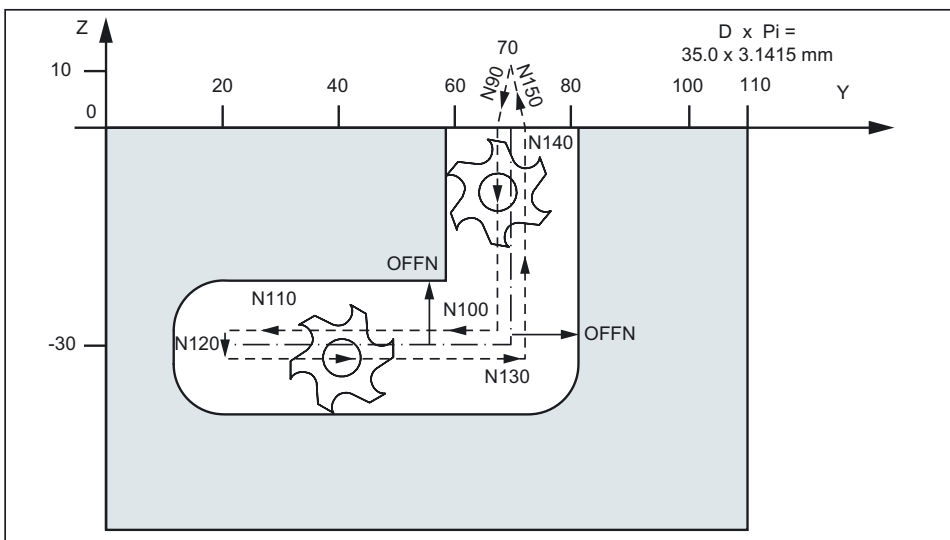


Figure 9-74 Programming the groove, values at the groove base

; Machining diameter of the cylinder at the groove base: 35,0 mm

; Desired total groove width: 24.8 mm, the used milling cutter has a radius of: 10.123 mm

```

N10 T1 F400 G94 G54           ; Milling cutter tool, feed, feed type, work offset
N15 G153 Y60                 ; Traverse Y to turning center of C axis
N30 G0 X25 Z50 C120          ; Approach the starting position
N40 TRACYL (35.0)             ; Activate TRACYL, machining diameter 35.0 mm
N50 G55 G19                   ; Work offset, plane selection: Y/Z plane
N60 S800 M3                   ; Activate spindle
N70 G0 Y70 Z10                ; Start position Y/Z,
                               ; Y is now the geometry axis of the lateral surface
N80 G1 X17.5                  ; Position milling cutter at groove base
N70 OFFN=12.4                 ; Distance from groove side to groove center line is
                               ; 12.4 mm
N90 G1 Y70 Z1 G42             ; Activate TRC, approach the groove wall
N100 Z-30                     ; Groove cut parallel to cylinder axis
N110 Y20                      ; Groove cut parallel to circumference
N120 G42 G1 Y20 Z-30          ; Restart TRC, approach the other groove wall,
                               ; 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              ; Deactivate TRC
N160 G0 X25                   ; Raise milling tool
N170 M5 OFFN=0                ; Deactivate spindle, clear groove wall distance
N180 TRAFOOF                  ; Deactivate TRACYL
N200 G54 G17 G0 X25 Z50 C120 ; Approach the starting position
N210 M2

```



## Cycles

### 10.1 Overview of cycles

Cycles are generally applicable technology subroutines that can be used to carry out a specific machining process, such as drilling of a thread (tapping) or milling of a pocket. 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 Manual Cycles.

#### Drilling cycle, drilling pattern cycles and milling 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 (Drilling 1)
  - CYCLE86: Boring (Drilling 2)
  - CYCLE87: Drilling with stop 1 (Drilling 3)
  - CYCLE88: Drilling with stop 2 (Drilling 4)
  - CYCLE89: Reaming 2 (Drilling 5)

With SINUMERIK 840D, the drilling cycles CYCLE85 ... CYCLE89 are called drilling 1 ... drilling 5, but are nevertheless identical in their function.

- Drilling pattern cycles  
HOLES1: Row of holes  
HOLES2: Circle of holes
- Milling cycles  
CYCLE71: Face milling  
CYCLE72: Contour milling  
CYCLE76: Milling the rectangular spigot  
CYCLE77: Circular spigot milling  
LONGHOLE: Elongated hole  
SLOT1: Groove milling pattern on a circle  
SLOT2: Circumferential groove milling pattern  
POCKET3: Rectangular pocket milling (with any milling tool)  
POCKET4: Circular pocket milling (with any milling tool)  
CYCLE90: Thread milling

The cycles are delivered with the toolbox and, if required, must be loaded into the part program memory using the RS232 interface.

### Cycle auxiliary subroutines

The cycle package includes the following auxiliary subroutines:

- cyclesm.spf
- steigung.spf and
- meldung.spf

These must always be loaded in the control.



## 10.2 Programming cycles

### Call and return conditions

The G functions effective prior to the cycle call and the programmable offsets remain active beyond the cycle.

The machining level (G17, G18, G19) must be defined before calling the cycle. A cycle operates in the current plane with the

- 1. axis of the plane (abscissa)
- 2. axis of the plane (ordinate)
- Drilling axis/infeed axis, third axis, standing vertically to the plane (applicata).

With drilling cycles, the drilling operation is carried out in the axis standing vertically to the current plane. In milling, the depth infeed is carried out in this axis.

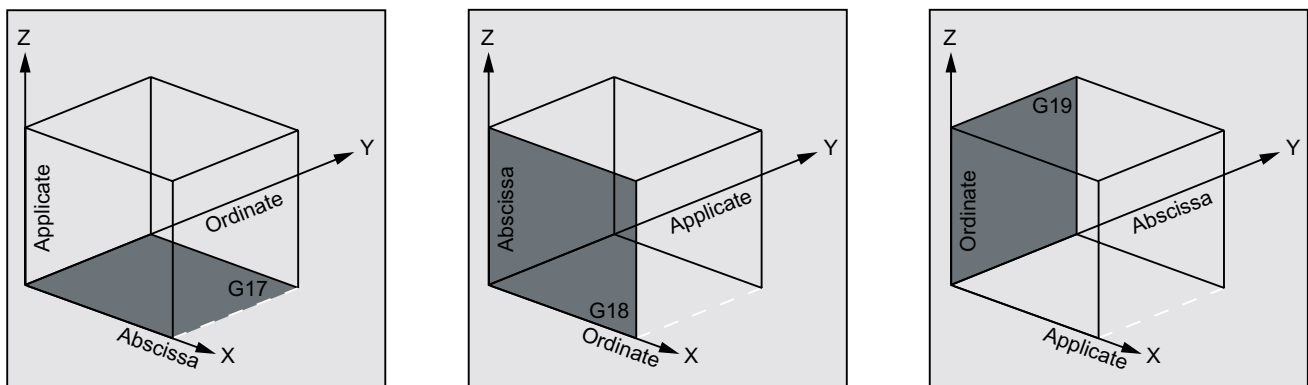


Figure 10-1 Plane and axis assignment

Table 10- 1 Plane and axis assignment

Command	Plane (abscissa/ordinate)	Vertical infeed axis (applicata)
G17	X/Y	Z
G18	Z/X	Y
G19	Y/Z	X

### 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 messages 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 is contained in the "Messages in the cycles" section.

### 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 Guide 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. In this parameter list, the following parameters can be transferred:

- 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 omitting parameters.

If you want to exclude the last transfer parameters that have to be written in a call, you can prematurely terminate the parameter list 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 screen for each cycle (is to be found in the input screen).

### Summary of required files

The following files constitute the basis for cycle support:

- cov.com
- sc.com

---

#### Note

These files must always be loaded in the control system. They are loaded during the start-up of the control system.

---

### Operating the cycle support

To add a cycle call to the program, carry out the following steps one after the other:

- Branching to selection bars for the individual cycles is possible in the horizontal softkey bar using the "Drilling" and "Milling" softkeys provided.
- The cycle selection is carried out using the vertical softkey bar until the appropriate input screen form with the help display appears on the screen.
- Then enter the values for the parameters.

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).

When numerical values are entered, a check is performed to see whether the value lies within the permitted 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.

A total of five cycles is provided for drilling. They all follow a different technological procedure and are therefore parameterized differently.

Boring cycle		Special parameterization features
Reaming 1	CYCLE85	Different feedrates for drilling and retraction
Boring	CYCLE86	Oriented spindle stop, specification of the retraction path, retraction at rapid traverse rate, specification of the direction of rotation of spindle
Drilling with stop 1	CYCLE87	Spindle stop M5 and program stop M0 at the drilling depth; to continue, press NC_Start, retraction at rapid traverse, specification of the direction of rotation of the spindle
Drilling with stop 2	CYCLE88	As with CYCLE87 plus dwell time to drilling depth
Reaming 2	CYCLE89	Drilling and retraction at the same feedrate

The drilling cycles can be modal, that is, they are executed at the end of each block containing motion commands. Further cycles created by the user can also be called modally (see also the "List of instructions" and the "Graphical cycle support in the program editor" sections).

There are two types of parameters:

- Geometrical parameters and
- Machining parameters

The geometrical parameters are identical for all drilling cycles, drilling pattern cycles and milling 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 CYCLE81.

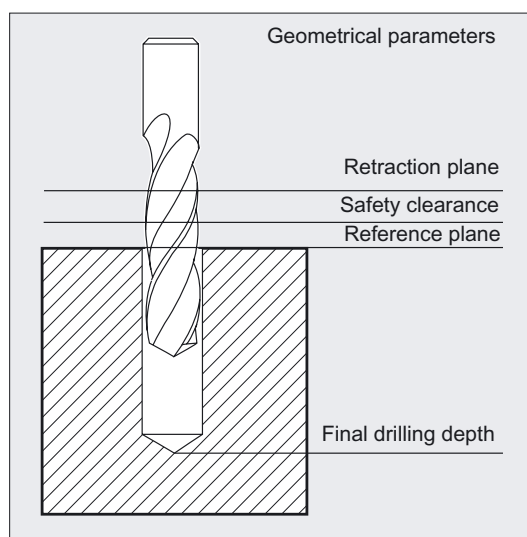


Figure 10-2 Drilling, centering - CYCLE81

The machining parameters have a different meaning and effect in the individual cycles. They are therefore programmed in each cycle separately.

## 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, G18 or G19 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.

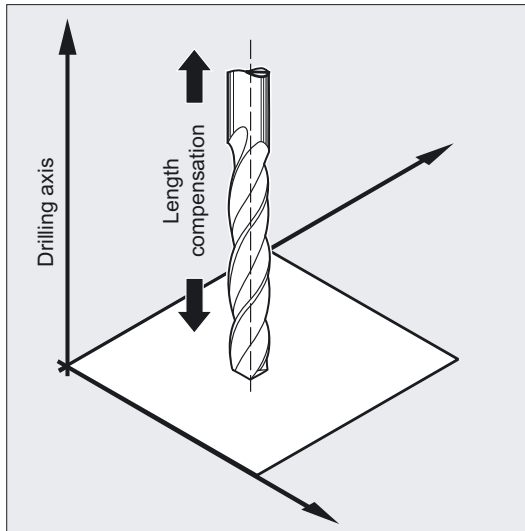


Figure 10-3 Length compensation

### 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.



### 10.4.3 Drilling, centering - CYCLE81

#### Programming

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

Table 10- 2 CYCLE81 parameters

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)

#### 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 retraction 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.

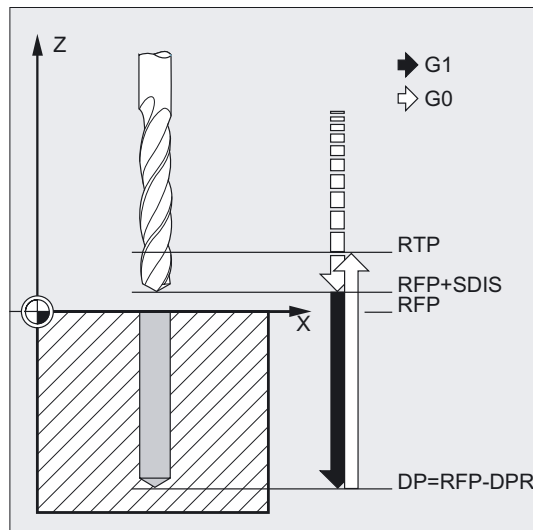


Figure 10-4 Final drilling depth

---

**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 dialog 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

Use this program to produce three drill holes using the CYCLE81 drilling cycle, whereby this is called using different parameters. The drilling axis is always the Z axis.

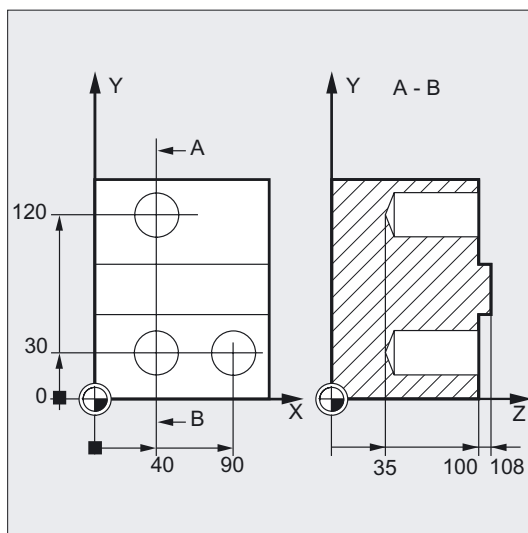


Figure 10-5 CYCLE81 example

<pre> N10 G0 G17 G90 F200 S300 M3 N20 D3 T3 Z110 N30 X40 Y120 N40 CYCLE81(110, 100, 2, 35) N50 Y30 N60 CYCLE81(110, 102, , 35) N70 G0 G90 F180 S300 M03 N80 X90 N90 CYCLE81(110, 100, 2, , 65) N100 M02 </pre>	<pre> ; Specification of technology values ; Approach retraction plane ; Approach of the first drilling position ; Cycle call with absolute final drilling depth, safety clearance and incomplete parameter list ; Approach next drilling position ; Cycle call without safety clearance ; Specification of technology values ; Approach next position ; Cycle call with relative final drilling depth and safety clearance ; End of program </pre>
--	---

**10.4.4 Drilling, counterboring - CYCLE82:****Programming**

CYCLE82 (RTP, RFP, SDIS, DP, DPR, DTB)

**Parameter**

Table 10-3 CYCLE82 parameters

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)
DTB	REAL	Dwell time at final drilling depth (chip breaking)

**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 the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

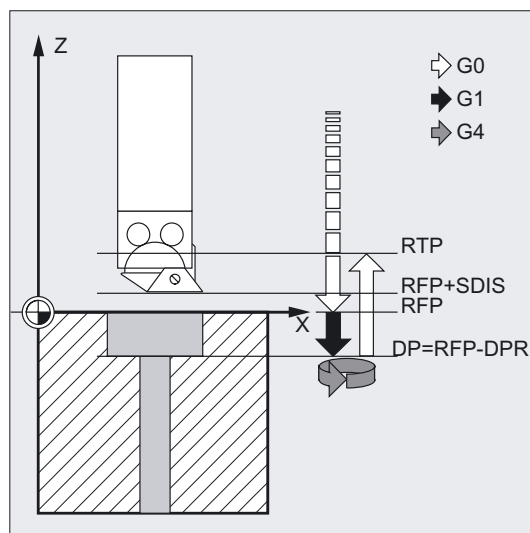


Figure 10-6 Explanation of the CYCLE82 parameters

## 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: Drilling\_counterboring**

The program machines a single hole of a depth of 27 mm at position X24 Y15 in the XY plane with cycle CYCLE82.

The dwell time programmed is 2 s, the safety clearance in the drilling axis Z is 4 mm.

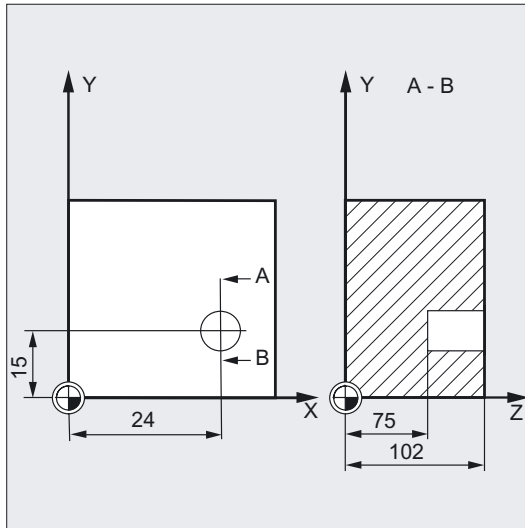


Figure 10-7 CYCLE82 example

N10 G0 G17 G90 F200 S300 M3	; Specification of technology values
N20 D1 T10 Z110	; Approach retraction plane
N30 X24 Y15	; Approach drilling position
N40 CYCLE82 (110, 102, 4, 75, , 2)	; Cycle call with absolute final drilling depth and safety clearance
N50 M02	; End of program

## 10.4.5 Deep-hole drilling - CYCLE83

### Programming

CYCLE83(RTP, RFP, SDIS, DP, DPR, FDEP, FDPR, DAM, DTB, DTS, FRF, VARI)

### Parameter

Table 10- 4 CYCLE83 parameters

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 breaking=0 Swarf 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 swarf 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

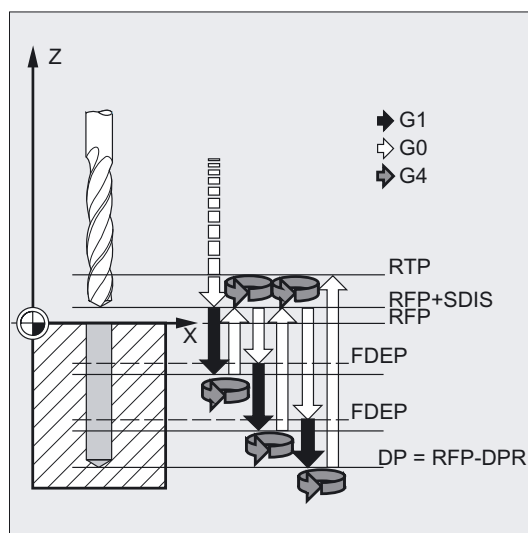


Figure 10-8 Deep hole drilling with swarf removal (VARI=1)

### 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

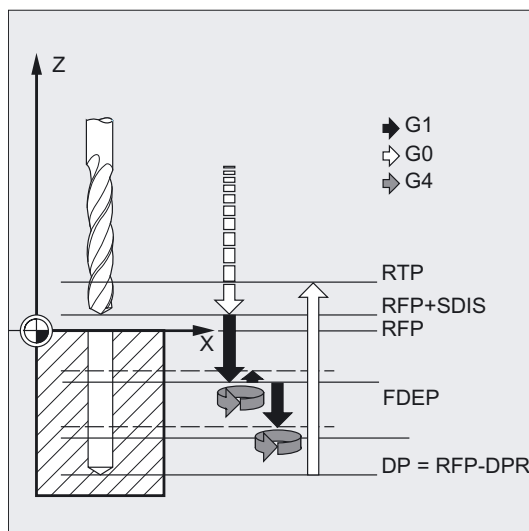


Figure 10-9 Deep-hole drilling with chip breakage (VARI=0)

### Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

### Interrelation of the DP (or DPR), FDEP (or FDPR) and DAM parameters

The intermediate drilling depth is 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 specify 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 breakage. 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  $\text{drilling depth} / 50$  is used (maximum value 7 mm).
-

### Programming example: Deep-hole drilling

This program executes the cycle CYCLE83 at the positions X80 Y120 and X80 Y60 in the XY plane. 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. In the second cycle call, a dwell time of 1 s is programmed. Machining type chip removal is selected, the final drilling depth is relative to the reference plane. The drilling axis in both cases is the Z axis.

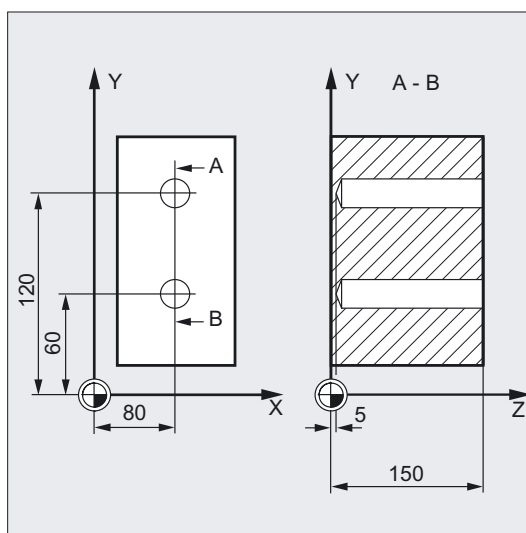


Figure 10-10 CYCLE83 example

```

N10 G0 G17 G90 F50 S500 M4           ; Specification of technology
                                       values
N20 D1 T12                           ; Approach retraction plane
N30 Z155
N40 X80 Y120                          ; Approach first drilling
                                       position
N50 CYCLE83(155, 150, 1, 5, 0, 100, , 20, 0, 0,
1, 0) ; Call of cycle; depth parameters
                                       with absolute values
N60 X80 Y60                          ; Approach next drilling position
N70 CYCLE83(155, 150, 1, , 145, , 50, 20, 1, 1,
0.5, 1) ; Cycle call with relative data
                                       for final drilling depth and
                                       first drilling depth; the safety
                                       clearance is 1 mm and the
                                       feedrate factor is 0.5
N80 M02                               ; End of program

```

## 10.4.6 Rigid tapping - CYCLE84

### Programming

CYCLE84(RTP, RFP, SDIS, DP, DPR, DTB, SDAC, MPIT, PIT, POSS, SST, SST1)

### Parameter

Table 10- 5 CYCLE84 parameters

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)
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) ... 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. For tapping with compensating chuck, a separate cycle CYCLE840 is provided.

### 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.

## 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 the 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

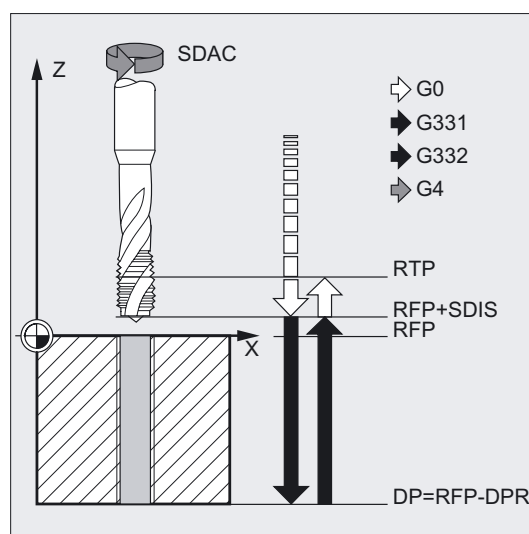


Figure 10-11 Explanation of the CYCLE84 parameters

### 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)

Under SDAC, the direction of rotation after end of cycle is programmed.

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). Any parameters not required are omitted in the call or 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)**

Before tapping, the spindle is stopped with orientation in the cycle by using the command SPOS and switched to position control.

The spindle position for this spindle stop is programmed under POSS.

### **SST (speed)**

Parameter SST contains the spindle speed for the tapping block with G331.

### **SST1 (retraction speed)**

The speed for retraction from the tapped hole is programmed under SST1.

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

A thread is tapped without compensating chuck at position X30 Y35 in the XY plane; the tapping 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.

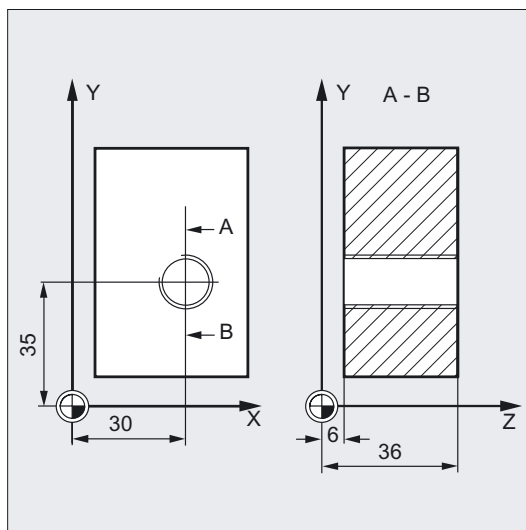


Figure 10-12 CYCLE84 example

<pre>N10 G0 G90 T11 D1 N20 G17 X30 Y35 Z40 N30 CYCLE84(40, 36, 2, , 30, , 3, 5, , 90, 200, 500) N40 M02</pre>	<pre>; Specification of technology values ; Approach drilling position Cycle call; parameter PIT has been 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 ; End of program</pre>
---	---

### 10.4.7 Tapping with compensating chuck - CYCLE840

#### Programming

CYCLE840(RTP, RFP, SDIS, DP, DPR, DTB, SDR, SDAC, ENC, MPIT, PIT, AXN)

#### Parameter

Table 10- 6 CYCLE840 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 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) ... 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, third axis of the plane

#### Function

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

This cycle is used to program 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:

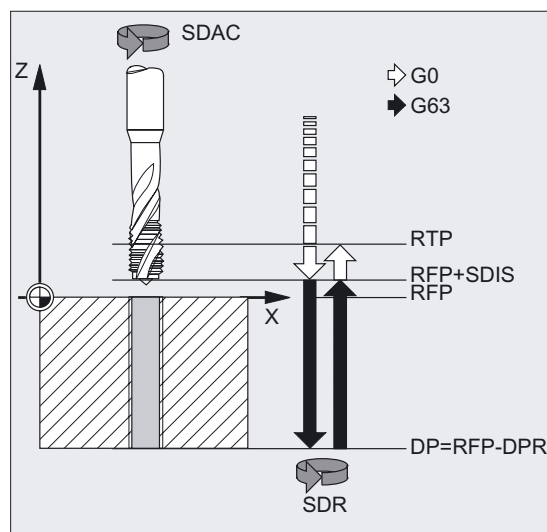


Figure 10-13 Tapping with compensating chuck without CYCLE840 encoder

- 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

### 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:

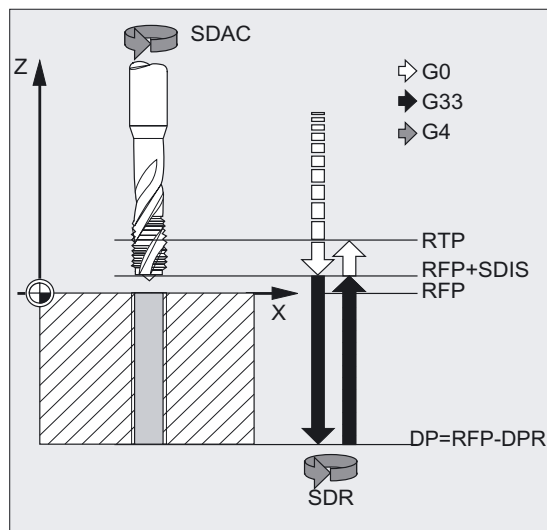


Figure 10-14 Tapping with compensating chuck with CYCLE840 encoder

- 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

### 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.

#### 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\_ENC 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 "Graphical cycle support" section), 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). Any parameters not required are omitted in the call or 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 aborted.

---

**Note**

Depending on the settings in machine data MD30200 \$MA\_NUM\_ENC\_S, 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)

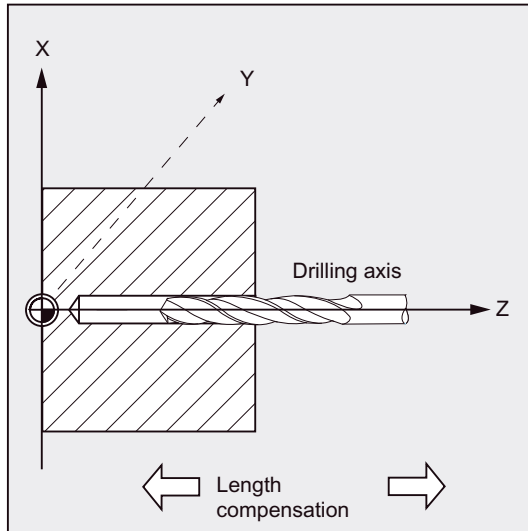


Figure 10-15 AXN (tool axis)

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

In this program, a thread is tapped without encoder at position X35 Y35 in the XY plane; the tapping 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.

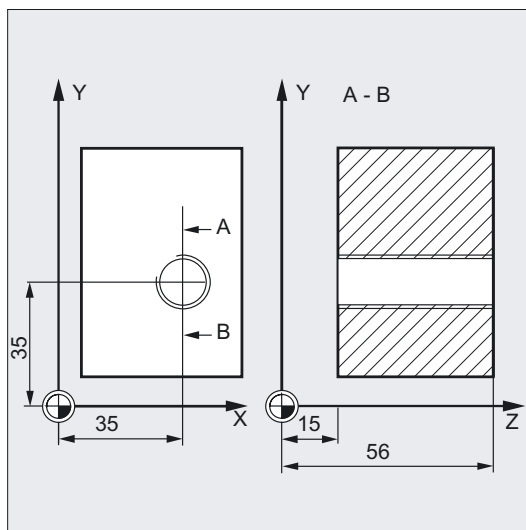


Figure 10-16 Thread without CYCLE840 encoder example

<pre>N10 G90 G0 T11 D1 S500 M3 N20 G17 X35 Y35 Z60 N30 G1 F200 N40 CYCLE840(59, 56, , 15, 0, 1, 4, 3, 1, , ) N50 M02</pre>	<pre>; Specification of technology values ; Approach drilling position ; Setting the path feedrate Cycle call, dwell time 1 s, direction of rotation for retraction M4, direction of rotation after cycle M3, no safety clearance, parameters MPIT and PIT have been omitted ; End of program</pre>
--	---

**Programming example: Tapping with encoder**

In this program, a thread is tapped with encoder at position X35 Y35 in the XY plane. 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.

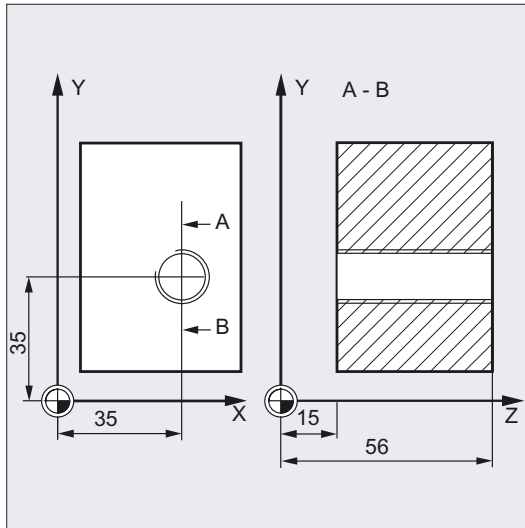


Figure 10-17 Thread with CYCLE840 encoder example

```

N10 G90 G0 T11 D1 S500 M4 ; Specification of technology
                               values
N20 G17 X35 Y35 Z60 ; Approach drilling position
N30 CYCLE840(59, 56, , 15, 0, 0, 4, 3, 0, 0,
3.5) ; Cycle call, without safety
clearance, with absolute depth
specification
N40 M02 ; End of program
    
```

## 10.4.8 Reaming 1 (Drilling 1) – CYCLE85

### Programming

CYCLE85 (RTP, RFP, SDIS, DP, DPR, DTB, FFR, RFF)

### Parameter

Table 10- 7 CYCLE85 parameters

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)
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.

### 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 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

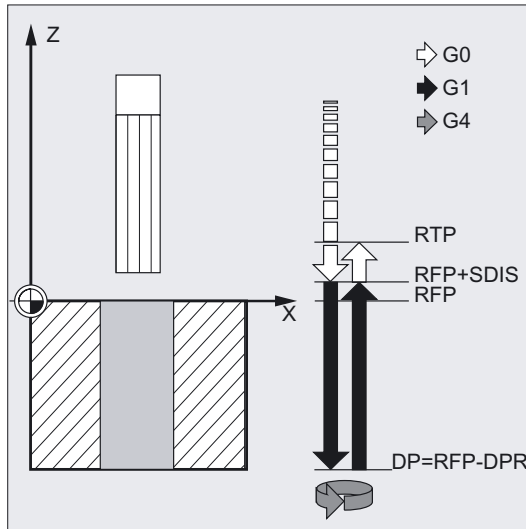


Figure 10-18 Explanation of the CYCLE85 parameters

**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 drilling

CYCLE85 is called at position Z70 X50 in the ZX plane. The drilling axis is the Y 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 Y102.

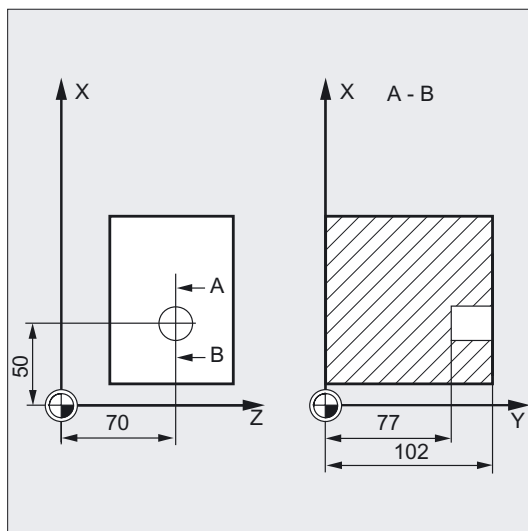


Figure 10-19 CYCLE85 example

<pre> N10 T11 D1 G1 F200 M3 S200 N20 G18 Z70 X50 Y105 N30 CYCLE85(105, 102, 2, , 25, , 300, 450) N40 M02 </pre>	<pre> ; Approach drilling position ; Cycle call, no dwell time programmed ; End of program </pre>
---	---

## 10.4.9 Boring (Drilling 2) – CYCLE86

### Programming

CYCLE86 (RTP, RFP, SDIS, DP, DPR, DTB, SDIR, RPA, RPO, RPAP, POSS)

### Parameter

Table 10- 8 CYCLE86 parameters

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)
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 first axis of the plane (incremental, enter with sign)
RPO	REAL	Retraction path along the second axis of the plane (incremental, enter with sign)
RPAP	REAL	Retraction path along the drilling 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 drilling 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.

### 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 drilling axis to the reference plane brought forward by the safety distance 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

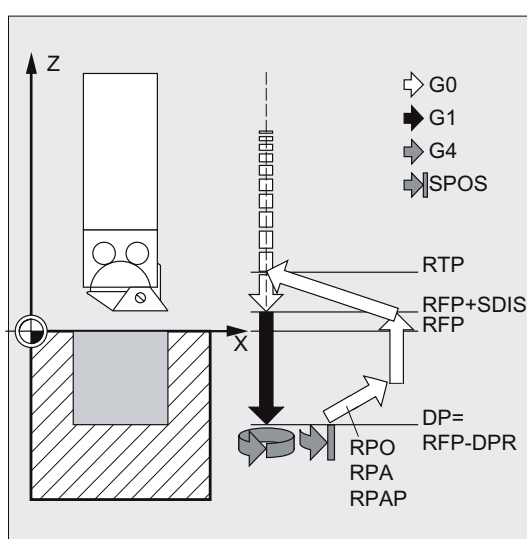


Figure 10-20 Explanation of the CYCLE86 parameters

### 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 first axis)

Use this parameter to define a retraction movement along the first axis (abscissa), which is executed after the final drilling depth has been reached and oriented spindle stop has been performed.

**RPO (retraction path along the second axis)**

Use this parameter to define a retraction movement along the second axis (ordinate), which is executed after the final drilling depth has been reached and oriented spindle stop has been performed.

**RPAP (retraction path along the drilling 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.

CYCLE86 can be used if the spindle to be used for the drilling operation is technically able to execute the SPOS command.

---

### Programming example: Second drilling

CYCLE86 is called at position X70 Y50 in the XY plane. 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 Z110. In the cycle, the spindle is to rotate with M3 and to stop at 45 degrees.

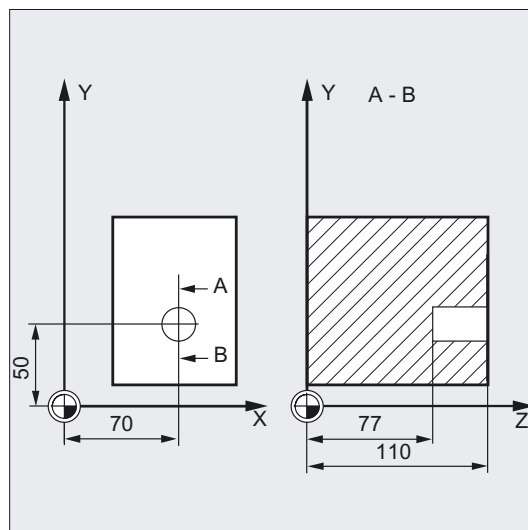


Figure 10-21 CYCLE86 example

<pre>N10 G0 G17 G90 F200 S300 M3 N20 T11 D1 Z112 N30 X70 Y50 N40 CYCLE86(112, 110, , 77, 0, 2, 3, -1, -1, 1, 45) N50 M02</pre>	<pre>; Specification of technology values ; Approach retraction plane ; Approach drilling position ; Cycle call with absolute drilling depth ; End of program</pre>
--	---

### 10.4.10 Boring with stop 1 (boring 3) – CYCLE87

#### Programming

CYCLE87 (RTP, RFP, SDIS, DP, DPR, SDIR)

#### Parameter

Table 10-9 CYCLE87 parameters

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)
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 drilling 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 the <NC START> button
- Retraction to the retraction plane with G0

### Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

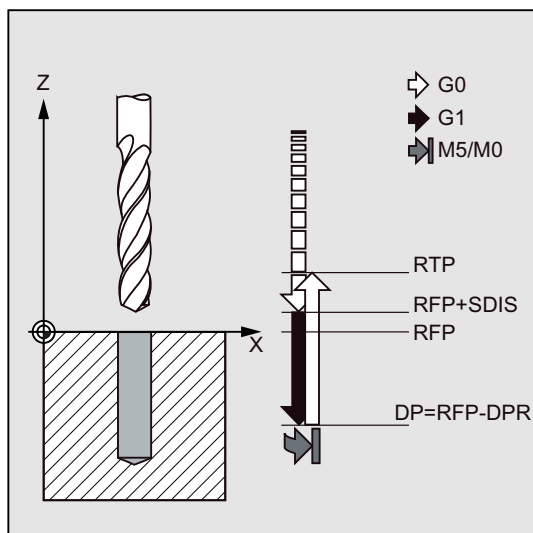


Figure 10-22 Explanation of the CYCLE87 parameters

### 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 drilling**

CYCLE87 is called at position X70 Y50 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.

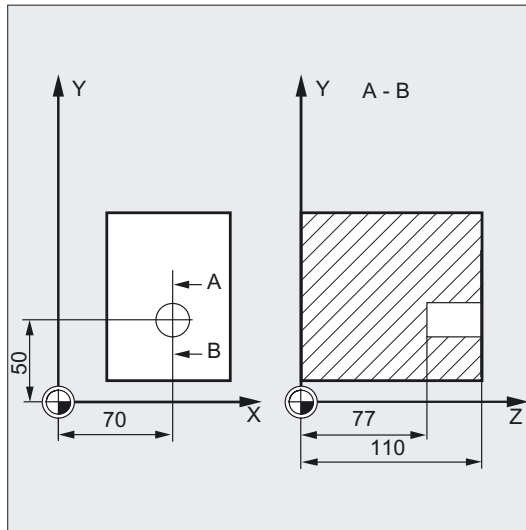


Figure 10-23 CYCLE87 example

```

DEF REAL DP, SDIS                                ; Definition of parameters
N10 DP=77 SDIS=2                                  ; Value assignments
N20 G0 G17 G90 F200 S300                          ; Specification of technology
                                                    values
N30 D3 T3 Z113                                     ; Approach retraction plane
N40 X70 Y50                                        ; Approach drilling position
N50 CYCLE87 (113, 110, SDIS, DP, , 3)             ; Cycle call with programmed
                                                    direction of rotation of spindle
M3
N60 M02                                           ; End of program

```



## 10.4.11 Drilling with stop 2 (Drilling 4) - CYCLE88

### Programming

CYCLE88 (RTP, RFP, SDIS, DP, DPR, DTB, SDIR)

### Parameter

Table 10- 10 CYCLE88 parameters

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)
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. When drilling with stop, 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 the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

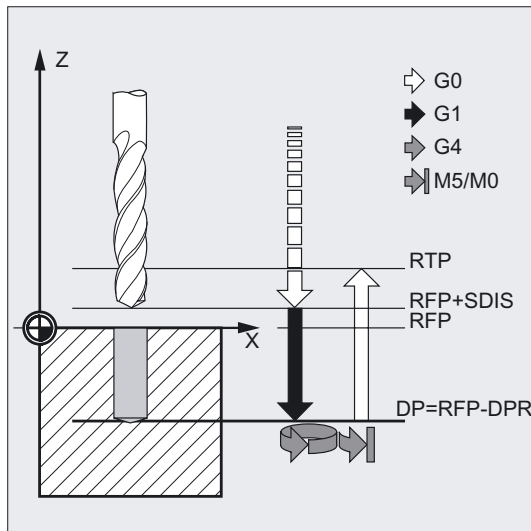


Figure 10-24 Explanation of the CYCLE88 parameters

### 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 drilling

CYCLE88 is called at position X80 Y90 in the XY plane. 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 G90 F100 S450           ; Specification of technology
                                values
N20 G0 X80 Y90 Z105           ; Approach drilling position
N30 CYCLE88 (105, 102, 3, , 72, 3, 4) ; Cycle call with programmed
                                spindle direction M4
N40 M02                       ; End of program

```

## 10.4.12 Reaming 2 (Drilling 5) – CYCLE89

### Programming

CYCLE89 (RTP, RFP, SDIS, DP, DPR, DTB)

### Parameter

Table 10- 11 Parameter CYCLE89

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)
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, the programmed dwell time is active.

### 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 the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

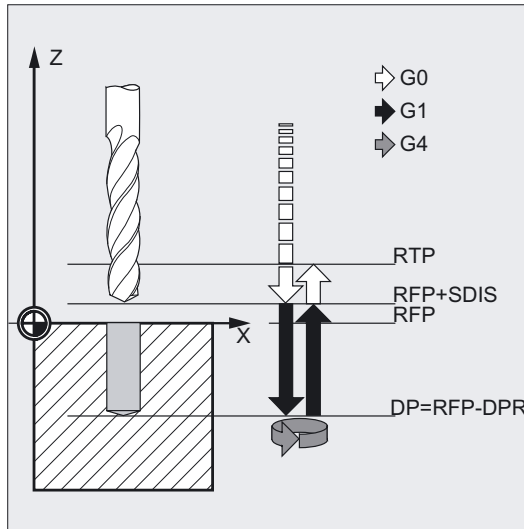


Figure 10-25 Explanation of the CYCLE89 parameters

### DTB (dwell time)

The dwell time to the final drilling depth (chip breakage) is programmed under DTB in seconds.

### Programming example: Fifth drilling

At X80 Y90 in the XY plane, the drilling cycle CYCLE89 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.

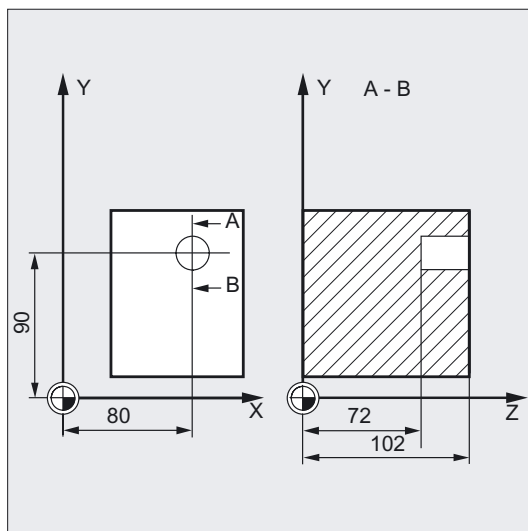


Figure 10-26 CYCLE89 example

<pre> DEF REAL RFP, RTP, DP, DTB RFP=102 RTP=107 DP=72 DTB=3 N10 G90 G17 F100 S450 M4 N20 G0 X80 Y90 Z107 N30 CYCLE89 (RTP, RFP, 5, DP, , DTB) N40 M02 </pre>	<pre> ; Definition of parameters ; Value assignments ; Specification of technology values ; Approach drilling position ; Cycle call ; End of program </pre>
---	---

## 10.5 Drilling pattern cycles

The drilling pattern cycles only describe the geometry of an arrangement of drilling holes in the plane. The link to a drilling process is established via the modal call of this drilling cycle before the drilling pattern cycle is programmed.

### 10.5.1 Requirements

#### Drilling pattern cycles without drilling cycle call

Drilling pattern cycles can also be used for other applications without prior modal call of a drilling cycle because the drilling pattern cycles can be parameterized without reference to the drilling cycle used.

If there was no modal call of the subroutine prior to calling the drilling pattern cycle, error message 62100 "No drilling cycle active" appears.

You can acknowledge this error message by pressing the error acknowledgment key and continue the program execution by pressing NC START. The drilling pattern cycle will then approach each of the positions calculated from the input data one after the other without calling a subroutine at these points.

#### Behavior when quantity parameter is zero

The number of holes in a drilling pattern must be parameterized. If the value of the quantity parameter is zero when the cycle is called (or if this parameter is omitted from the parameter list), alarm 61103 "Number of holes is zero" is issued and the cycle is aborted.

#### Checks in case of limited ranges of input values

Generally, there are no plausibility checks for defining parameters in the drilling pattern cycles.

## 10.5.2 Row of holes - HOLES1

### Programming

HOLES1 (SPCA, SPCO, STA1, FDIS, DBH, NUM)

### Parameter

Table 10- 12 HOLES1 parameters

Parameter	Data type	Description
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 first axis of the plane (abscissa) Range of values: $-180 < STA1 \leq 180$ 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 holes

### Function

This cycle can be used to produce a row of holes, i.e. a number of holes arranged along a straight line, or a grid of holes. The type of hole is determined by the drilling cycle that has already been called modally.

### 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.

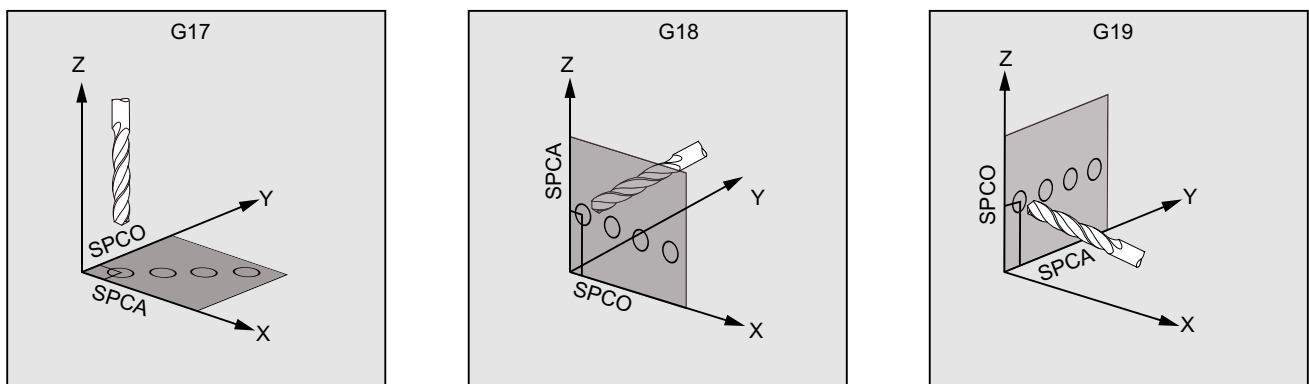


Figure 10-27 HOLES1 sequence

### Explanation of the parameters

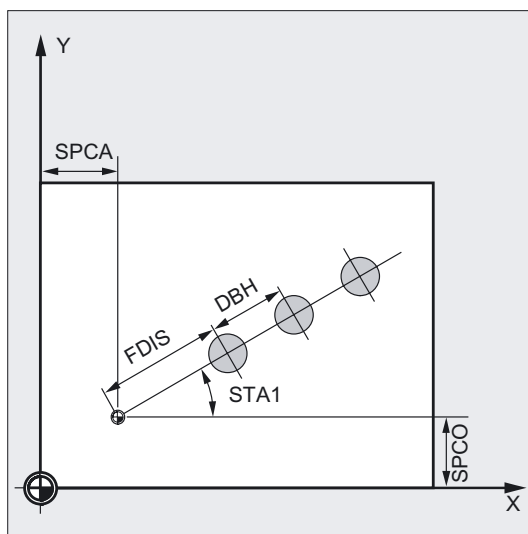


Figure 10-28 Explanation of the HOLES1 parameters

#### SPCA and SPCO (reference point on the first axis of the plane and of the second 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 first 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 and the reference point defined under SPCA and SPCO is programmed with FDIS. The parameter DBH contains the distance between any two holes.

#### NUM (number)

The NUM parameter is used to define the number of holes.

#### Programming example: Row of holes

Use this program to machine a row of holes consisting of five tapped holes arranged parallel to the Z axis of the ZX plane and which have a distance of 20 mm one to another. The starting point of the row of holes is at Z20 and X30 whereby the first hole has a distance of 10 mm from this point. 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





**Programming example: Grid of holes**

Use this program to machine a grid of holes consisting of five rows with five holes each, which are arranged in the XY plane, with a spacing of 10 mm between them. The starting point of the grid is at X30 Y20.

The example uses R parameters as transfer parameters for the cycle.

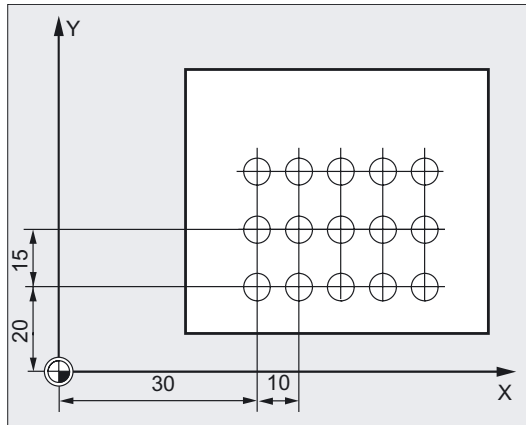


Figure 10-30 Grid of holes - HOLES1 example

```

R10=102          ; Reference plane
R11=105          ; Retraction plane
R12=2            ; Safety clearance
R13=75           ; Drilling depth
R14=30           ; Reference point for the row of holes in the first axis of the
                  plane
R15=20           ; Reference point for the row of holes in the second axis of the
                  plane
R16=0            ; Starting angle
R17=10           ; Distance from first hole to reference point
R18=10           ; Distance between the holes
R19=5            ; Number of holes per row
R20=5            ; Number of rows
R21=0            ; Row counter
R22=10           ; Distance between the rows

N10 G90 F300 S500 M3 T10 D1          : Specification of the
                                      technological values
N20 G17 G0 X=R14 Y=R15 Z105          ; Approach starting position
N30 MCALL CYCLE82(R11, R10, R12, R13, 0, 1) ; Modal call of drilling cycle
N40 LABEL1:                          ; Call of row of holes cycle
N41 HOLES1(R14, R15, R16, R17, R18, R19)
N50 R15=R15+R22                      ; Calculate y value for the next
                                      line
N60 R21=R21+1                        ; Increment line counter
N70 IF R21<R20 GOTOB LABEL1          ; Return to LABEL1 if the
                                      condition is fulfilled

```

```

N80 MCALL ; Deselect modal call
N90 G90 G0 X30 Y20 Z105 ; Approach starting position
N100 M02 ; End of program

```

### 10.5.3 Circle of holes - HOLES2

#### Programming

HOLES2 (CPA, CPO, RAD, STA1, INDA, NUM)

#### Parameter

Table 10- 13 HOLES2 parameters

Parameter	Data type	Description
CPA	REAL	Center point of circle of holes (absolute), first axis of the plane
CPO	REAL	Center point of circle of holes (absolute), second axis of the plane
RAD	REAL	Radius of circle of holes (enter without sign)
STA1	REAL	Starting angle Range of values: $-180 < STA1 \leq 180$ degrees
INDA	REAL	Incrementing angle
NUM	INT	Number of 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 hole is determined by the drilling cycle that has already been called modally.

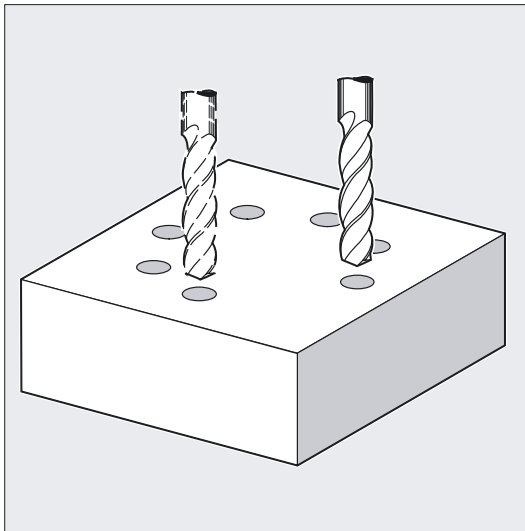


Figure 10-31 Row of holes - HOLES2

**Sequence**

In the cycle, the drilling positions are approached one after the other in the plane with G0.

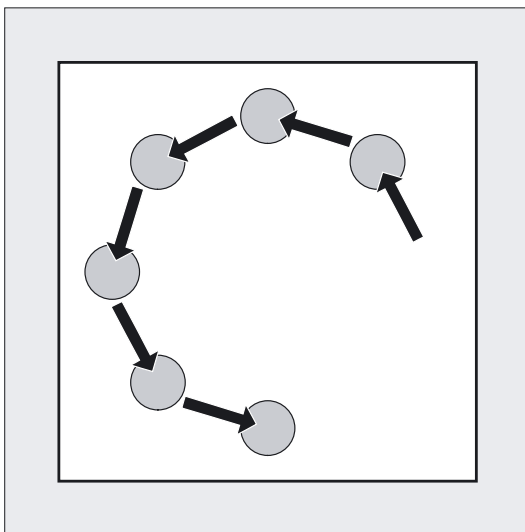


Figure 10-32 HOLES2 sequence

## Explanation of the parameters

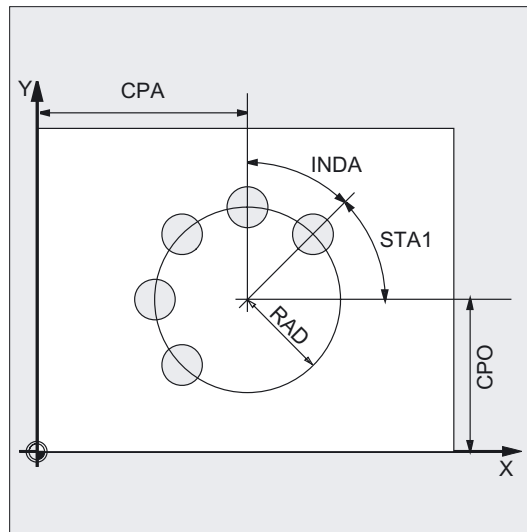


Figure 10-33 Explanation of the HOLES2 parameters

### 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.

The STA1 parameter defines the angle of rotation between the positive direction of the first axis (abscissa) in the workpiece coordinate system active before the cycle was called and the first hole. The INDA parameter 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 having a depth of 30 mm. The final drilling depth is specified as a relative value to the reference plane. The circle is defined by the center point X70 Y60 and the radius 42 mm in the XY plane. The starting angle is 33 degrees. The safety clearance in drilling axis Z is 2 mm.

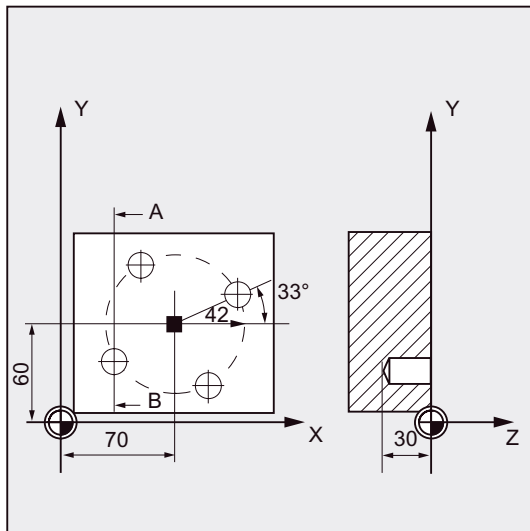


Figure 10-34 Circle of holes - HOLES2 example

N10 G90 F140 S170 M3 T10 D1	; Specification of technology values
N20 G17 G0 X50 Y45 Z2	; Approach starting position
N30 MCALL CYCLE82(2, 0, 2, , 30, 0)	; Modal call of the drilling cycle, without dwell time, DP is not programmed
N40 HOLES2 (70, 60, 42, 33, 0, 4)	; Call of the circle-of-holes cycle; the incremental angle is calculated in the cycle since the parameter INDA has been omitted
N50 MCALL	; Deselect modal call
N60 M02	; End of program

## 10.6 Milling cycles

### 10.6.1 Requirements

#### Call and return conditions

Milling cycles are programmed independently of the particular axis name.

Before you call the milling cycles, a tool compensation must be activated.

The appropriate values for feedrate, spindle speed and direction of rotation of spindle must be programmed in the part program if the appropriate parameters are not provided in the milling cycle.

The center point coordinates for the milling pattern or the pocket to be machined are programmed in a rectangular coordinate system.

The G functions active prior to the cycle call and the current programmable frame remain active beyond the cycle.

#### Plane definition

Milling cycles generally assume that the current workpiece coordinate system has been defined by selecting a plane (G17, G18 or G19) and activating a programmable frame (if necessary). The infeed axis is always the third axis of this coordinate system.

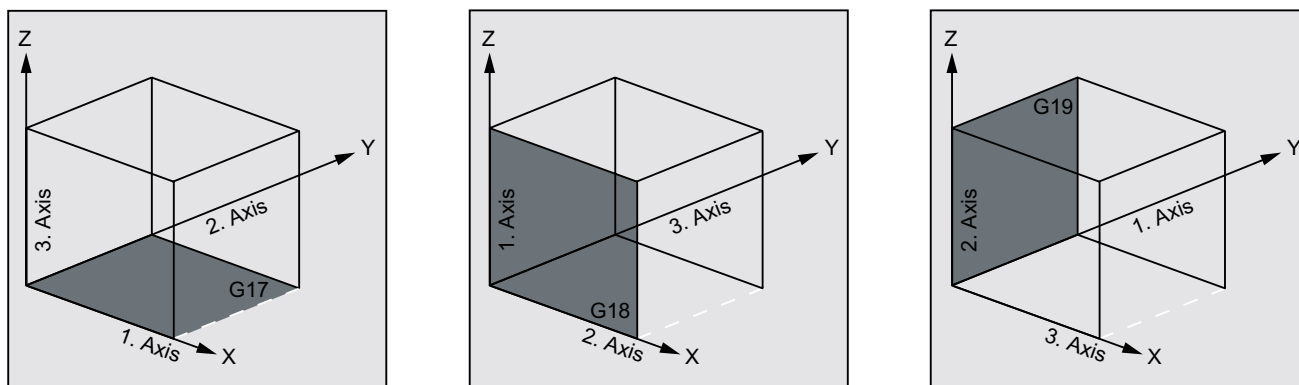


Figure 10-35 Plane and axis assignment

### Messages with regard to the machining state

During the execution of the milling cycles, various messages that refer to the machining status are displayed on the screen. The following messages are possible:

- "Elongated hole <No.>(first figure) being machined"
- "Slot <No.>(other figure) being machined"
- "Circumferential slot <No.>(last figure) being machined"

In each case, <No.> stands for the number of the figure that is currently being machined.

These message do not interrupt the program execution and continue to be displayed until the next message is displayed or the cycle is completed.

## 10.6.2 Face milling - CYCLE71

### Programming

CYCLE71 (RTP, RFP, SDIS, DP, PA, PO, LENG, WID, STA, MID, MIDA, FDP, FALD, FFP1, VARI, FDP1)

### Parameter

Table 10- 14 CYCLE71 parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (to be added to the reference plane; enter without sign)
DP	REAL	Depth (absolute)
PA	REAL	Starting point (absolute), first axis of the plane
PO	REAL	Starting point (absolute), second axis of the plane
LENG	REAL	Rectangle length along the first axis, incremental. The corner from which the dimension starts results from the sign.
WID	REAL	Rectangle length along the second axis, incremental. The corner from which the dimension starts results from the sign.
STA	REAL	Angle between the longitudinal axis of the rectangle and the first axis of the plane (abscissa, enter without sign). Range of values: $0^\circ \leq STA < 180^\circ$
MID	REAL	Maximum infeed depth (enter without sign)
MIDA	REAL	Maximum infeed width during solid machining in the plane as a value (enter without sign)
FDP	REAL	Retraction travel in the finishing direction (incremental, enter without sign)
FALD	REAL	Finishing dimension in the depth (incremental, enter without sign)
FFP1	REAL	Feedrate for surface machining



Parameter	Data type	Description
VARI	INT	Machining type (enter without sign) UNITS DIGIT Values: 1 roughing, 2 finishing TENS DIGIT: Values: 1 parallel to the first axis of the plane, in one direction, 2 parallel to the second axis of the plane, in one direction, 3 parallel to the first axis of the plane, with alternating direction 4 parallel to the second axis of the plane, with alternating direction
FDP1	REAL	Overrun travel in the direction of the plane infeed (incremental, enter without sign)

## Function

Use CYCLE71 to face mill any rectangular surface. The cycle differentiates between roughing (machining the surface in several steps until reaching the final machining allowance) and finishing (end milling the surface in one step). The maximum infeed in width and depth can be specified.

The cycle operates without cutter radius compensation. The depth infeed is performed in the open.

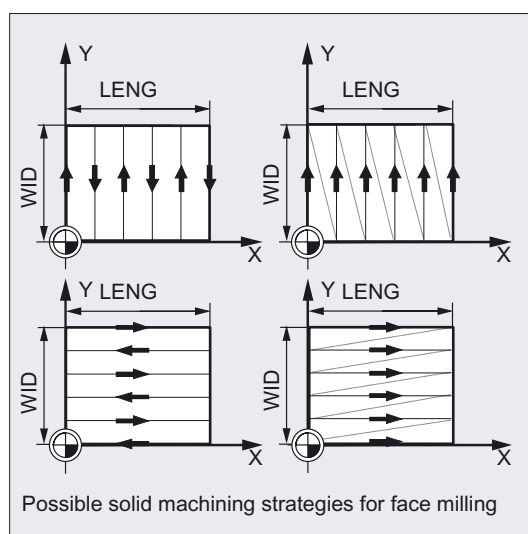


Figure 10-36 Possible solid machining strategies for face milling

## Sequence

### Position reached prior to cycle start:

Starting position is any position from which the infeed point can be approached at the height of the retraction plane without collision.

**The cycle creates the following sequence of motions:**

- G0 is applied to approach the infeed point at the current position level. The reference plane, brought forward by the safety distance, is then also approached with G0 to this position. Then, also with G0, feeding to the machining plane. G0 is possible since infeed in the open is possible.

There are several roughing strategies (paraxial in one direction or back and forth).

- Sequence of motions when roughing:

Face milling can be performed in several planes based on the programmed values DP, MID and FALD. Machining is carried out from the top downward, i.e. one plane each is removed and then the next depth infeed is carried out in the open (FDP parameters). The traversing paths for solid machining in the plane depend on the values of the parameters LENG, WID, MIDA, FDP, FDP1 and the cutter radius of the active tool.

The first path to be milled is always traversed such that the infeed depth exactly corresponds to MIDA, ensuring that no width infeed larger than the maximum possible width infeed occurs. The tool center point therefore does not always travel exactly on the edge (only if MIDA = cutter radius). The dimension by which the tool traverses outside the edge is always equal to the cutter diameter - MIDA even if only one surface cut is performed, i. e. area width + overrun is less than MIDA. The other paths for width infeed are calculated internally so as to produce a uniform path width ( $\leq$ MIDA).

- Sequence of motions when finishing:

When finishing, the surface is milled in the plane once. This means that the finishing allowance when roughing has to be selected also such that the residual depth can be removed with the finishing tool in one step.

After each surface milling pass in the plane, the tool will retract. The retraction travel is programmed under the parameter FDP.

Machining in one direction stops at the final machining allowance + safety distance and the next starting point is approached in rapid traverse.

When roughing in one direction, the tool will retract by the calculated infeed depth + safety clearance. The depth infeed is performed at the same point as in roughing.

After finishing has been completed, the tool retracts from the last position reached to the retraction plane RTP.

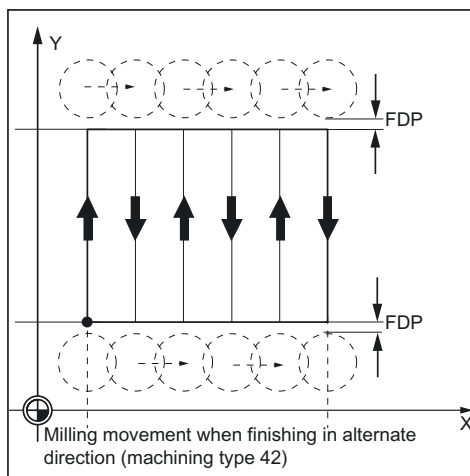


Figure 10-37 Milling motion when finishing in one direction

### Explanation of the parameters

For parameters RTP, RFP, SDIS - see CYCLE81

For the parameters STA, MID, FFP1 - see POCKET3.

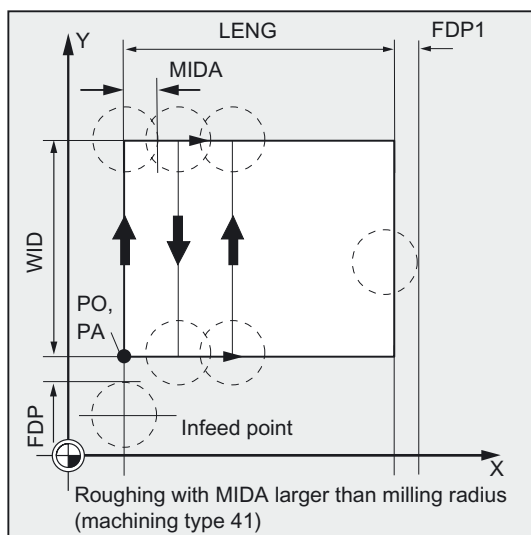


Figure 10-38 Roughing with MIDA larger than milling radius

### DP (depth)

The depth can be specified as an absolute value (DP) to the reference plane.

### PA, PO (starting point)

Use the parameters PA and PO to define the starting point of the area in the axes of the plane.

### LENG, WID (length)

Use the parameters LENG and WID to define the length and width of a rectangle in the plane. The position of the rectangle, with reference to PA and PO, results from the sign.

### MIDA (max. infeed width)

Use this parameter to define the maximum infeed width when machining in a plane. Analogously to the known calculation method for the infeed depth (equal distribution of the total depth with maximum possible value) the width is distributed equally, maximally with the value programmed under MIDA.

If this parameter is not programmed or has value 0, the cycle will internally use 80% of the milling tool diameter as the maximum infeed width.

**FDP (retraction travel)**

Use this parameter to define the dimension for the retraction travel in the plane. This parameter should reasonably always have a value greater than zero.

**FDP1 (overrun travel)**

Use this parameter to specify an overrun travel in the direction of the plane infeed (MIDA). Thus, it is possible to compensate the difference between the current cutter radius and the tool nose radius (e.g. cutter radius or cutting tips arranged at an angle). The last milling cutter center point path therefore always results as  $\_LENG$  (or  $\_WID$ ) +  $\_FDP1$  - tool radius (from the compensation table).

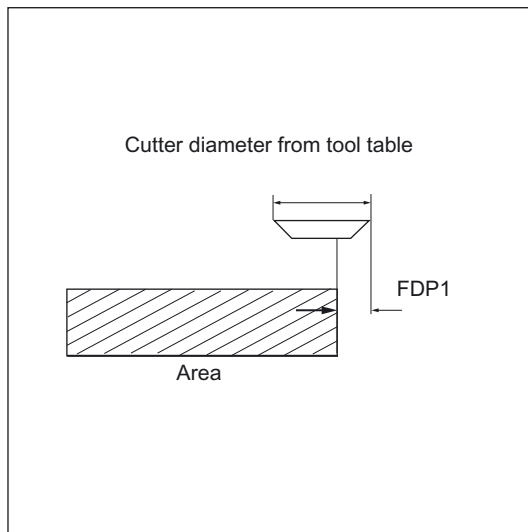


Figure 10-39 FDP1 overrun travel

**FALD (finishing allowance)**

When roughing, a finishing allowance in the depth is taken into account which is programmed under this parameter.

The residual material remained as the finishing allowance must always be specified for finishing to ensure that the tool can be retracted and then fed to the starting point of the next cut without collision.

If  $> 0$ , the parameter is ignored for finishing.

## VARI (machining type)

Use the parameter VARI to define the machining type.

Possible values are:

- Units digit:
  - 1=roughing to finishing allowance
  - 2=finishing
- Tens digit:
  - 1=parallel to the first axis of the plane; unidirectional
  - 2=parallel to the second axis of the plane; unidirectional
  - 3=parallel to the first axis of the plane; with alternating direction
  - 4=parallel to the second axis of the plane; with alternating direction

If a different value is programmed for the parameter VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

---

### Note

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

---

## Programming example: Face milling

### Parameters for the cycle call:

- Retraction plane: 10 mm
- Reference plane: 0 mm
- Safety distance: 2 mm
- Milling depth: -11 mm
- Starting point of the rectangle: X = 100 mm, Y = 100 mm
- Rectangle dimensions: X = +60 mm, Y = +40 mm
- Angle of rotation in the plane: 10 degrees
- Max. infeed depth: 6 mm
- Max. infeed width: 10 mm
- Retraction travel at the end of the milling path: 5 mm
- No final machining allowance: -
- Feedrate for surface machining: 4000 mm/min
- Machining type: Roughing parallel to the X axis with alternating direction
- Overrun on last cut as determined by the cutting edge geometry: mm

A milling cutter with 10 mm radius is used.

```

N10 T2 D2
N20 G17 G0 G90 G54 G94 F2000 X0 Y0 Z20           ; Approach start position
N30 CYCLE71(10, 0, 2, -11, 100, 100, 60, 40, 10,   ; Cycle call
6, 10, 5, 0, 4000, 31, 2)
N40 G0 G90 X0 Y0
N50 M02                                           ; End of program

```

### 10.6.3 Contour milling - CYCLE72

#### Programming

CYCLE72 (KNAME, RTP, RFP, SDIS, DP, MID, FAL, FALD, FFP1, FFD, VARI, RL, AS1, LP1, FF3, AS2, LP2)

#### Parameter

Table 10- 15 CYCLE72 parameters

Parameter	Data type	Description
KNAME	STRING	Name of contour subroutine
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (to be added to the reference plane; enter without sign)
DP	REAL	Depth (absolute)
MID	REAL	Maximum infeed depth (incremental; enter without sign)
FAL	REAL	Finishing allowance at the edge contour (enter without sign)
FALD	REAL	Finishing allowance at the base (incremental, enter without sign)
FFP1	REAL	Feedrate for surface machining
FFD	REAL	Feedrate for depth infeed (enter without sign)

Parameter	Data type	Description
VARI	INT	Machining type (enter without sign) UNITS DIGIT Values: 1 roughing, 2 finishing TENS DIGIT: Values: 0 intermediate travel with G0, 1 intermediate travel with G1 HUNDREDS DIGIT Values: 0 Retraction at the end of contour to RTP 1 Retraction at the end of contour to RFP + SDIS 2 Retraction by SDIS at the end of contour 3 No retraction at the end of contour
RL	INT	Traveling around the contour either centrally, to the right or to the left (with G40, G41 or G42; enter without sign) Values: 40...G40 (approach and return, straight line only) 41...G41 42...G42
AS1	INT	Specification of the approach direction/path: (enter without sign) UNITS DIGIT: Values: 1...Straight tangential line 2...Quadrant 3...Semi-circle TENS DIGIT: Values: 0...Approach to the contour in the plane 1...Approach to the contour in a spatial path
LP1	REAL	Length of the approach travel (with straight-line) or radius of the approach arc (with circle) (enter without sign)

The other parameters can be selected as options.

FF3	REAL	Retraction feedrate and feedrate for intermediate positions in the plane (in the open)
AS2	INT	Specification of the retraction direction/path: (enter without sign) UNITS DIGIT: Values: 1...Straight tangential line 2...Quadrant 3...Semi-circle TENS DIGIT: Values: 0...Retraction from the contour in the plane 1...Retraction from the contour in a spatial path
LP2	REAL	Length of the retraction travel (with straight-line) or radius of the retraction arc (with circle) (enter without sign)

## Function

Use CYCLE72 to mill along any contour defined in a subroutine. The cycle operates with or without cutter radius compensation.

It is not imperative that the contour is closed. Internal or external machining is defined via the position of the cutter radius compensation (centrally, left or right to the contour).

The contour must be programmed in the direction as it is to be milled and must consist of a minimum of two contour blocks (start and end point), since the contour subroutine is called directly internally in the cycle.

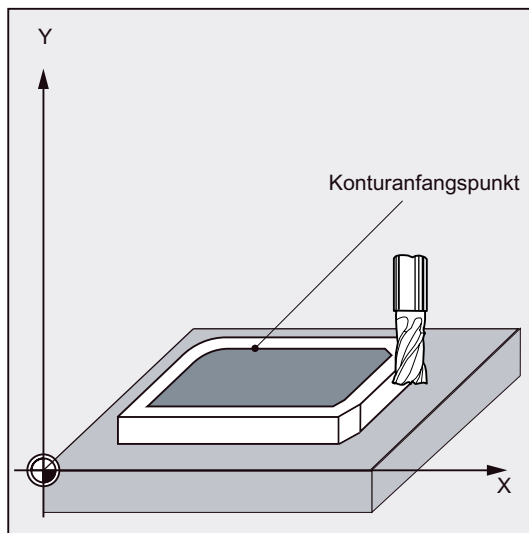


Figure 10-40 Path milling 1



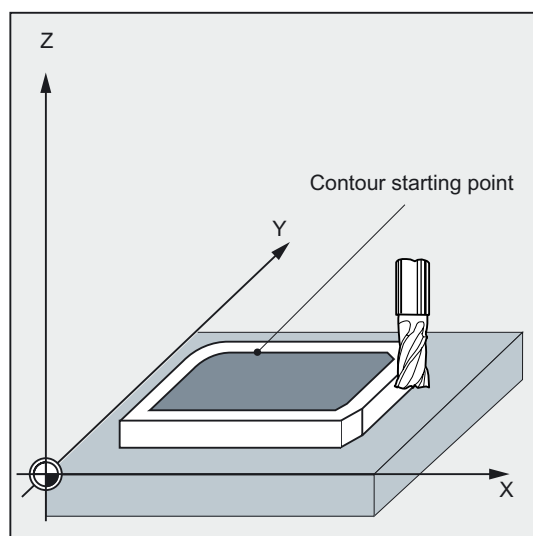


Figure 10-41 Path milling 2

### Functions of the cycle

- Selection of roughing (single-pass traversing parallel to contour, taking into account a finishing allowance, if necessary at several depths until the finishing allowance is reached) and finishing (single-pass traversing along the final contour if necessary at several depths)
- Smooth approach to and retraction from the contour either tangentially or radially (quadrant or semi-circle)
- Programmable depth infeeds

Intermediate motions either at rapid traverse rate or at feedrate

### Sequence

#### Position reached prior to cycle start:

Starting position is any position from which the contour starting point can be approached at the height of the retraction plane without collision.

The cycle generates the following sequence of motions when roughing:

The depth infeeds are distributed equally with the maximum possible value of the specified parameters.

- Traversing to the starting point for first milling with G0/G1 (and FF3). This point is calculated internally in the control system and depends on
  - the contour starting point (first point in the subroutine),
  - the direction of the contour at the starting point,
  - the approach mode and its parameters and
  - the tool radius.  
The cutter radius compensation is activated in this block.
- Depth infeed to the first or next machining depth plus programmed safety clearance with G0/G1. The first machining depth results from:
  - The total depth,
  - The finishing allowance and
  - The maximum possible depth infeed.
- Approach of the contour vertically with depth infeed and then in the plane at the programmed feedrate or 3D with the feedrate programmed under FAD according to the programming for smooth approach.
- Milling along the contour with G40/G41/G42.
- Smooth retraction from the contour with G1 while continuing feed for the surface machining by the retraction amount.
- Retraction with G0 /G1 (and feedrate for intermediate paths FF3), depending on the programming.
- Retraction to the depth infeed point with G0/G1 (and FF3).
- This sequence is repeated on the next machining plane up to finishing allowance in the depth.

A completion of roughing, the tool stands above the point (calculated internally in the control system) of retraction from the contour at the height of the retraction plane.

**The cycle generates the following sequence of motions when finishing:**

During finishing, milling is performed at the relevant infeed along the base of the contour until the final dimension is reached.

Smooth approach and retraction of the contour is carried out according to the existing parameters. The appropriate path is calculated internally in the control system.

At the end of the cycle, the tool is positioned at the contour retraction point at the height of the retraction level.

---

#### Note

#### Contour programming

When programming the contour, please observe the following:

- No programmable offset must be selected in the subroutine prior to the first programmed position.
  - The first block of the contour subroutine is a straight line block containing G90 / G0 or G90 / G1 and defines the start of the contour.
  - The starting condition of the contour is the first position in the machining plane which is programmed in the contour subroutine.
  - The cutter radius compensation is selected/deselected by the higher-level cycle; therefore, no G40, G41, G42 is programmed in the contour subroutine.
- 

#### Explanation of the parameters

For parameters RTP, RFP, SDIS - see CYCLE81

For parameters MID, FAL, FALD, FFP1, FFD, DP - see POCKET3.

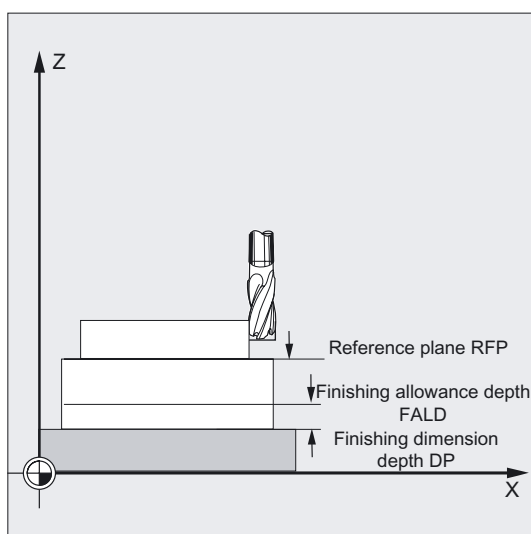


Figure 10-42 Explanation of the CYCLE72 parameters

**KNAME (name)**

The contour to be milled is programmed completely in a subroutine. KNAME defines the name of the contour subroutine.

1. The contour can be defined as a subroutine:

KNAME=name of subroutine

The name of the contour subroutine is subject to all name conventions described in the Programming Guide.

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:

KNAME=name of 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 the "Contour append" softkey.

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.

Examples:

```
KNAME="KONTUR1"
```

The milling contour is the complete program Kontur\_1.

```
KNAME="ANFANG:ENDE"
```

The milling contour is defined as a section in the calling program, which starts from the block containing label ANFANG (START) to the block containing label ENDE (END).

### LP1, LP2 (length, radius)

Use parameter `_LP1` to program the approach travel or approach radius (distance from the tool external edge to the contour starting point), and `_LP2` to program the retraction travel or retraction radius (distance from the tool external edge to the contour end point).

The values LP1, LP2 must be set to >0. In the case of zero, error 61116 "Approach or retraction path=0" is output.

---

#### Note

When using G40, the approach or retraction travel is the distance from the tool center point to the start or end point of the contour.

---

### VARI (machining type)

Use the parameter VARI to define the machining type. Possible values are:

- UNITS DIGIT

Values:

- 1 Roughing
- 2 Finishing

- TENS DIGIT:

Values:

- 0 Intermediate travel with G0
- 1 Intermediate travel with G1

- HUNDREDS DIGIT

Values:

- 0 Retraction at the end of contour to RTP
- 1 Retraction at the end of contour to RFP + SDIS
- 2 Retraction by SDIS at the end of contour
- 3 No retraction at the end of contour

If a different value is programmed for the parameter VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

### RL (bypassing the contour)

With the parameter RL, you program the traveling around the contour centrally, to the right or to the left with G40, G41 or G42. For possible values, see "Parameters for CYCLE72".

### AS1, AS2 (approach direction/path, retraction direction/path)

Use the parameter AS1 to program the specification of the approach path and AS2 to program that of the retraction path. For possible values, see "Parameters for CYCLE72". If AS2 is not programmed, then the behavior of the retraction path is analogous to that of the approach path.

Smooth approach of the contour along a spatial path (helix or straight line) should only be programmed if the tool is not yet being used or is suitable for this type of approach.

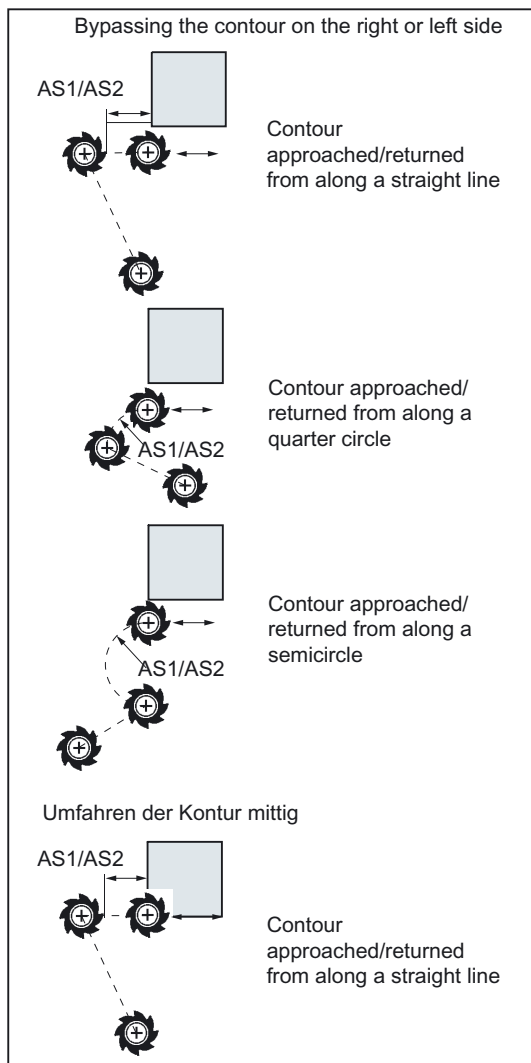


Figure 10-43 Traversing of the contour

In the case of central (G40), approach and retraction is only possible along a straight line.

### FF3 (retraction feedrate)

Use parameter FF3 to define a retraction feedrate for intermediate positions in the plane (in the open) if the intermediate motions are to be carried out with feedrate (G01). If no feedrate value is programmed, the intermediate motions with G01 are carried out at surface feedrate.

---

#### Note

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

---

### Programming example 1: Milling around a closed contour externally

This program is used to mill the contour shown in the diagram below.

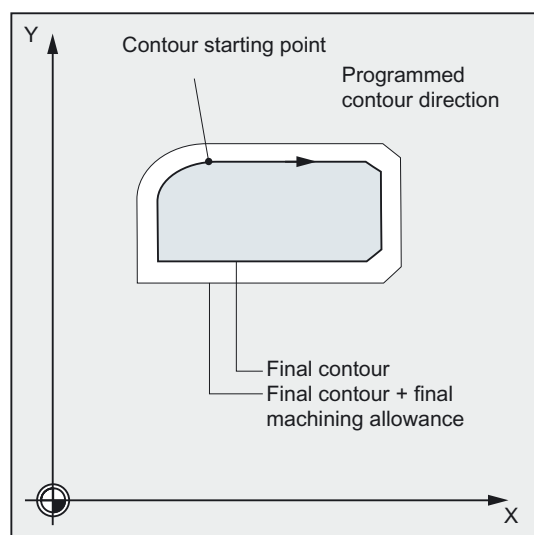


Figure 10-44 Milling around a closed contour - CYCLE72 example

Parameters for the cycle call:

- Retraction plane: 250 mm
- Reference plane: 200
- Safety distance: 3 mm
- Depth: 175 mm
- Maximum depth infeed: 10 mm
- Finishing allowance in the depth: 1.5 mm
- Depth infeed feedrate: 400 mm/min
- Finishing allowance in the plane: 1 mm
- Feedrate in the plane: 800 mm/min
- Machining: Roughing up to finishing allowance; intermediate paths with G1, for intermediate paths retraction in Z to RFP + SDIS

Parameters for approach:

- G41 - left of the contour, i.e. external machining
- Approach and retraction in a quadrant in the plane: 20 mm radius
- Retraction feedrate: 1000 mm/min

```

N10 T3 D1 ; T3: Milling cutter with radius
          7
N20 S500 M3 F3000 ; Program feedrate and spindle
                  speed
N30 G17 G0 G90 X100 Y200 Z250 G94 ; Approach start position
N40 CYCLE72("EX72CONTOUR", 250, 200, 3, 175, ; Cycle call
          10,1, 1.5, 800, 400, 111, 41, 2, 20, 1000, 2,
          20)
N50 X100 Y200
N60 M2 ; End of program
%NEX72CONTOURSPF ; Subroutine for contour milling
                  (for example)
N100 G1 G90 X150 Y160 ; Starting point of contour
N110 X230 CHF=10
N120 Y80 CHF=10
N130 X125
N140 Y135
N150 G2 X150 Y160 CR=25
N160 M2
N170 M02

```



## Programming example - 2

Milling around a closed contour externally, as in programming example 1, with contour programming in the calling program

```
N10 T3 D1 ; T3: Milling cutter with radius  
7  
N20 S500 M3 F3000 ; Program feedrate and spindle  
speed  
N30 G17 G0 G90 X100 Y200 Z250 G94 ; Approach start position  
N40 CYCLE72 ( "PIECE245:PIECE245E", 250, 200, 3, ; Cycle call  
175, 10,1, 1.5, 800, 400, 11, 41, 2, 20, 1000,  
2, 20)  
N50 X100 Y200  
N60 M2  
N70 PIECE245: ; Contour  
N80 G1 G90 X150 Y160  
N90 X230 CHF=10  
N100 Y80 CHF=10  
N110 X125  
N120 Y135  
N130 G2 X150 Y160 CR=25  
N140 PIECE245E: ; End of contour  
N150 M2
```

### 10.6.4 Rectangular spigot milling - CYCLE76

#### Programming

CYCLE76 (RTP, RFP, SDIS, DP, DPR, LENG, WID, CRAD, PA, PO, STA, MID, FAL, FALD, FFP1, FFD, CDIR, VARI, AP1, AP2)

#### Parameter

Table 10- 16 Parameters for CYCLE76

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)
LENG	REAL	Spigot length
WID	REAL	Spigot width
CRAD	REAL	Spigot corner radius (enter without sign)
PA	REAL	Reference point of spigot, abscissa (absolute)
PO	REAL	Reference point of spigot, ordinate (absolute)
STA	REAL	Angle between longitudinal axis and first axis of plane
MID	REAL	Maximum depth infeed (incremental; enter without sign)
FAL	REAL	Final machining allowance at the margin contour (incremental)
FALD	REAL	Finishing allowance at the base (incremental, enter without sign)
FFP1	REAL	Feedrate on contour
FFD	REAL	Feedrate for depth infeed
CDIR	INT	Milling direction (enter without sign) Values: 0 Down-cut milling 1 Conventional milling 2 With G2 (independent of spindle direction) 3 With G3
VARI	INT	Machining type Values: 1 Roughing to final machining allowance 2 Finishing (allowance X/Y/Z=0)
AP1	REAL	Length of blank spigot
AP2	REAL	Width of blank spigot

## Function

Use this cycle to machine rectangular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always carried out in the position upstream of the semi-circle style approach to the contour.

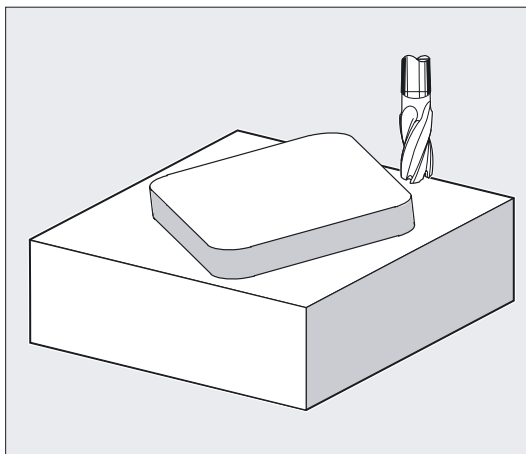


Figure 10-45 Rectangular spigot milling - CYCLE76

## Sequence

### Position reached prior to cycle start:

The starting point is a position in the positive range of the abscissa with the approach semi-circle and the programmed raw dimension on the abscissa end taken into account.

### Motion sequence when roughing (VARI=1):

*Approach/retraction from contour:*

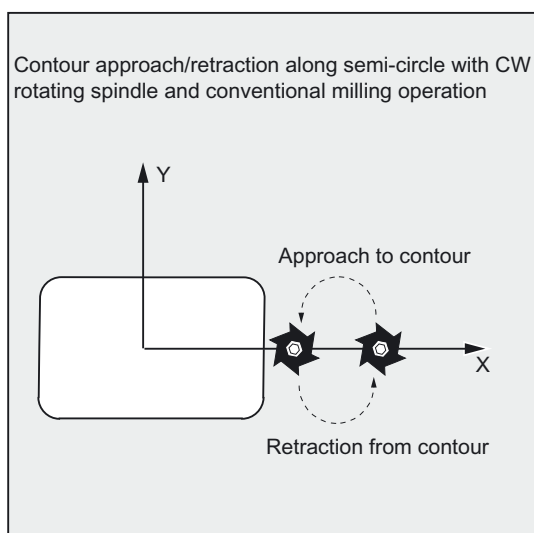


Figure 10-46 Sequence of motions when roughing

The retraction plane (RTP) is approached at rapid traverse rate to then be able to position to the starting point in the machining plane at this height. The starting point is defined with reference to 0 degrees of the abscissa.

The tool is fed to the safety clearance (SDIS) at rapid traverse with subsequent traversing to the machining depth at feedrate. To approach the spigot contour, the tool travels along a semi-circular path.

The milling direction can be determined either as up-cut milling or down-cut milling with reference to the spindle direction.

If the spigot is bypassed once, the contour is left along a semi-circle in the plane, and the tool is fed to the next machining depth.

The contour is then reapproached along a semi-circle and the spigot traversed once. This process is repeated until the programmed spigot depth is reached. Then, the retraction plane (RTP) is approached at rapid traverse rate.

*Depth infeed:*

- Feeding to the safety clearance
- Insertion to machining depth

The first machining depth is calculated from:

- The total depth,
- The finishing allowance and
- The maximum possible depth infeed.

**Sequence of motions when finishing (VARI=2):**

According to the set parameters FAL and FALD, finishing is either carried out at the surface contour or at the base or both together. The approach strategy corresponds to the motions in the plane as with roughing.

**Explanation of the parameters**

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81.

For parameters MID, FAL, FALD, FFP1, FFD - see POCKET3.

### LENG, WID and CRAD (spigot length, spigot width and corner radius)

Use the parameters LENG, WID and CRAD to define the form of a slot in the plane.

The spigot is always dimensioned from the center. The length (LENG) always refers to the abscissa (with a plane angle of 0 degrees).

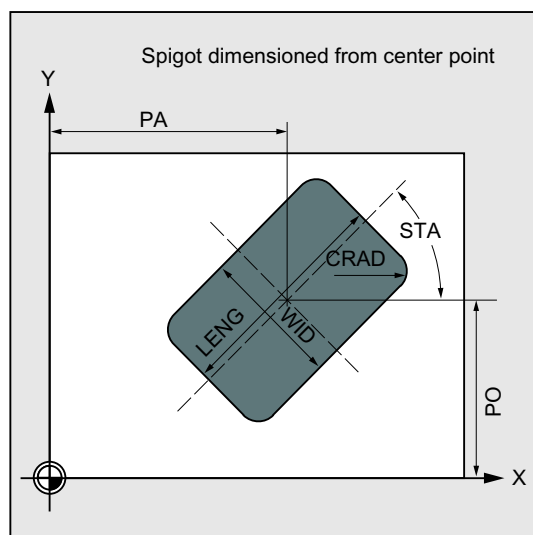


Figure 10-47 Spigot dimensioned from the center point

### PA, PO (reference point)

Use the parameters PA and PO to define the reference point of the spigot along the abscissa and the ordinate.

This is the spigot center point.

### STA (angle)

STA specifies the angle between the first axis of the plane (abscissa) and the longitudinal axis of the spigot.

### CDIR (milling direction)

Use this parameter to specify the machining direction for the spigot.

Using the CDIR parameter, the milling direction

- can be programmed directly with "2 for G2" and "3 for G3" or,
- alternatively with "synchronous milling" or "conventional milling".

Down-cut and up-cut milling are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

#### Down-cut and up-cut milling:

M3 → G3, M3 → G2

M4 → G2, M4 → G3

**VARI (machining type)**

Use the parameter VARI to define the machining type.

Possible values are:

- 1=roughing
- 2=finishing

**AP1, AP2 (blank dimensions)**

When machining the spigot, it is possible to take into account blank dimensions (e.g. when machining precast parts).

The basic sizes for the length and width (AP1 and AP2) are programmed without sign and their symmetrical positions around the spigot center are computed in the cycle. The internally calculated radius of the approach semi-circle depends on this dimension.

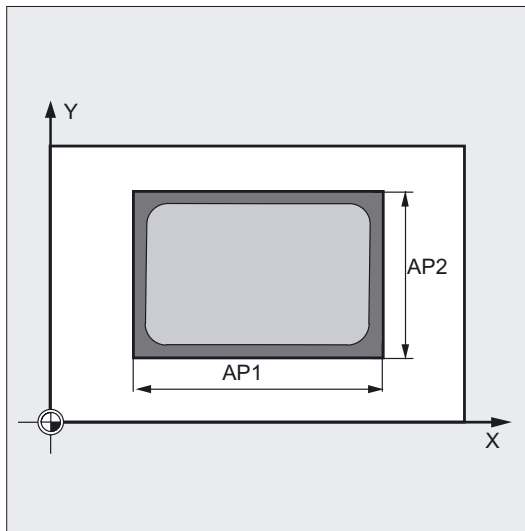


Figure 10-48 Blank dimensions (AP1/AP2)

---

**Note**

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is canceled and alarm 61009 "Active tool number=0" is output.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original coordinate system is active again.

---

### Programming example: Spigot

Use this program to machine in the XY plane a spigot that is 60 mm long, 40 mm wide and has 15 mm corner radius. The spigot has an angle of 10 degrees relative to the X axis and is premanufactured with a length allowance of 80 mm and a width allowance of 50 mm.

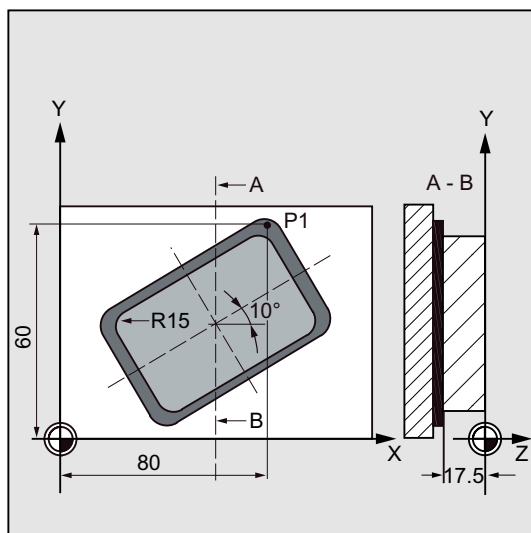


Figure 10-49 Example of spigot milling - CYCLE76

```

N10 G90 G0 G17 X100 Y100 T20 D1 S3000 M3           ; Specification of technology
                                                    values
N11 M6
N30 CYCLE76 (10, 0, 2, -17.5, , 60, 40, 15, 80,    ; Cycle call
60, 10, 11, , , 900, 800, 0, 1, 80, 50)
N40 M30                                           ; End of program
    
```

### 10.6.5 Circular spigot milling - CYCLE77

#### Programming

CYCLE77 (RTP, RFP, SDIS, DP, DPR, PRAD, PA, PO, MID, FAL, FALD, FFP1, FFD, CDIR, VARI, AP1)

#### Parameter

The following input parameters are always required:

Table 10- 17 Parameters for CYCLE77

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Depth (absolute)
DPR	REAL	Depth relative to the reference plane (enter without sign)
PRAD	REAL	Spigot diameter (enter without sign)
PA	REAL	Center point of spigot, abscissa (absolute)
PO	REAL	Center point of spigot, ordinate (absolute)
MID	REAL	Maximum depth infeed (incremental; enter without sign)
FAL	REAL	Final machining allowance at the margin contour (incremental)
FALD	REAL	Finishing allowance at the base (incremental, enter without sign)
FFP1	REAL	Feedrate on contour
FFD	REAL	Feedrate for depth infeed (or spatial infeed)
CDIR	INT	Milling direction (enter without sign) Values: 0 Down-cut milling 1 Conventional milling 2 With G2 (independent of spindle direction) 3 With G3
VARI	INT	Machining type Values: 1 Roughing to final machining allowance 2 Finishing (allowance X/Y/Z=0)
AP1	REAL	Length of blank spigot



## Function

Use this cycle to machine circular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always performed in the position before the semi-circular approach to the contour.

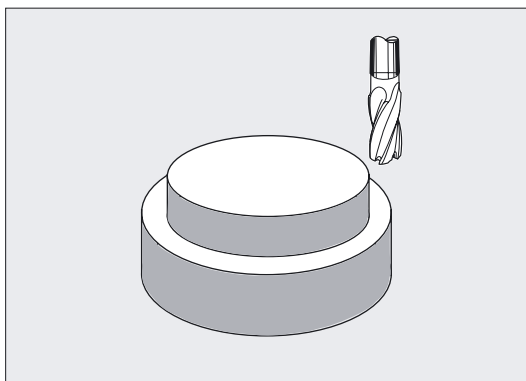


Figure 10-50 Circular spigot milling - CYCLE77

## Sequence

### Position reached prior to cycle start:

The starting point is a position in the positive range of the abscissa with the approach semi-circle and the programmed raw dimension taken into account.

### Motion sequence when roughing (VARI=1):

*Approach/retraction from contour:*

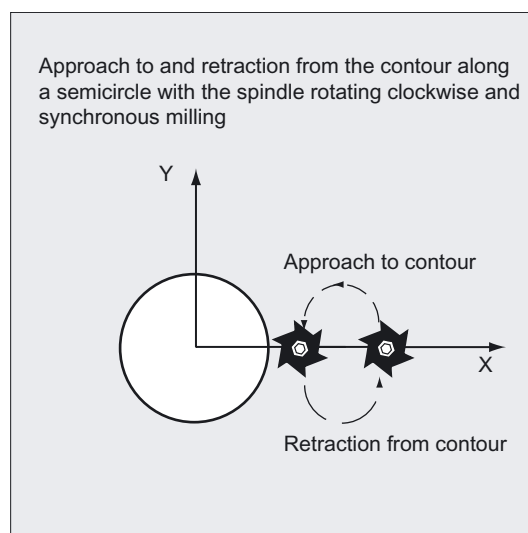


Figure 10-51 Sequence of motions for roughing

The retraction plane (RTP) is approached at rapid traverse rate to then be able to position at this height to the starting point in the machining plane. The starting point is defined with reference to 0 degrees of the axis of the abscissa.

The tool is fed to the safety clearance (SDIS) at rapid traverse with subsequent traversing to the machining depth at feedrate. To approach the spigot contour, the tool is approached along a semi-circular path using the programmed blank spigot.

The milling direction can be determined either as up-cut milling or down-cut milling with reference to the spindle direction.

If the spigot is bypassed once, the contour is left along a semi-circle in the plane, and the tool is fed to the next machining depth.

The contour is then reapproached along a semi-circle and the spigot traversed once. This process is repeated until the programmed spigot depth is reached.

Then, the retraction plane (RTP) is approached at rapid traverse rate.

*Depth infeed:*

- Feeding to the safety clearance
- Insertion to machining depth

The first machining depth is calculated from:

- The total depth,
- The finishing allowance and
- The maximum possible depth infeed.

**Sequence of motions when finishing (VARI=2):**

According to the set parameters FAL and FALD, either finishing is carried out at the surface contour or at the base or both together. The approach strategy corresponds to the motions in the plane as with roughing.

### Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81.

For parameters MID, FAL, FALD, FFP1, FFD - see POCKET3.

### PRAD (diameter of spigot)

Enter the diameter without sign.

### PA, PO (spigot center point)

Use the parameters PA and PO to define the reference point of the spigot.

### CDIR (milling direction)

Use this parameter to specify the machining direction for the spigot. Using the parameter CDIR, the milling direction

- can be programmed directly with "2 for G2" and "3 for G3" or,
- alternatively with "synchronous milling" or "conventional milling".

Down-cut and up-cut milling are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

**Down-cut and up-cut milling:**

M3 → G3, M3 → G2

M4 → G2, M4 → G3

**VARI (machining type)**

Use the parameter VARI to define the machining type. Possible values are:

- 1=roughing
- 2=finishing

**AP1 (diameter of blank spigot)**

Use this parameter to define the blank dimension of the spigot (without sign). The internally calculated radius of the approach semi-circle depends on this dimension.

---

**Note**

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is canceled and alarm 61009 "Active tool number=0" is output. Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original coordinate system is active again.

---

**Programming example: Circular spigot**

Machining a spigot from a blank with a diameter of 55 mm and a maximum infeed of 10 mm per cut; specification of a final machining allowance for subsequent finishing of the spigot surface. The whole machining is performed with reverse rotation.

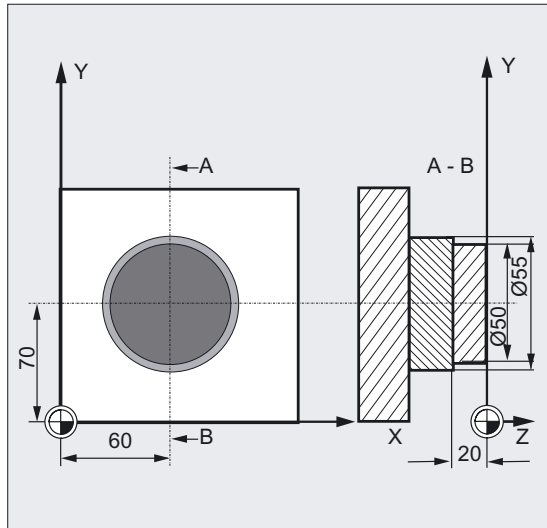


Figure 10-52 Circular spigot milling - CYCLE77 example

```

N10 G90 G17 G0 S1800 M3 D1 T1 ; Specification of technology
                                values
N11 M6
N20 CYCLE77 (10, 0, 3, -20, ,50, 60, 70, 10, ; Roughing cycle call
0.5, 0, 900, 800, 1, 1, 55)
N30 D1 T2 M6 ; Change tool
N40 S2400 M3 ; Specification of technology
                                values
N50 CYCLE77 (10, 0, 3, -20, , 50, 60, 70, 10, 0, ; Call finishing cycle
0, 800, 800, 1, 2, 55)
N40 M30 ; End of program

```

## 10.6.6 Long holes located on a circle - LONGHOLE

### Programming

LONGHOLE (RTP, RFP, SDIS, DP, DPR, NUM, LENG, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID)

### Parameter

Table 10- 18 LONGHOLE parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Slot depth (absolute)
DPR	REAL	Slot depth relative to the reference plane (enter without sign)
NUM	INT	Number of slots
LENG	REAL	Slot length (enter without sign)
CPA	REAL	Center point of circle (absolute), first axis of the plane
CPO	REAL	Center point of circle (absolute), second axis of the plane
RAD	REAL	Radius of the circle (enter without sign)
STA1	REAL	Starting angle
INDA	REAL	Incrementing angle
FFD	REAL	Feedrate for depth infeed
FFP1	REAL	Feedrate for surface machining
MID	REAL	Maximum infeed depth for one infeed (enter without sign)

### Note

The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

### Function

Use this cycle to machine long holes located on a circle. The longitudinal axis of the long holes is aligned radially.

In contrast to the slot, the width of the long hole is determined by the tool diameter.

Internally in the cycle, an optimum traversing path of the tool is determined, ruling out unnecessary idle passes. If several depth infeeds are required to machine a slot, the infeed is carried out alternately at the end points. The path to be traversed along the longitudinal axis of the long hole changes its direction after each infeed. The cycle searches for the shortest path when changing to the next long hole.

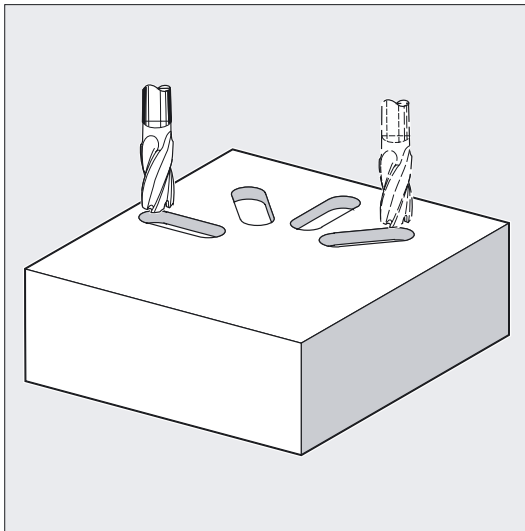


Figure 10-53 Long holes located on a circle - LONGHOLE

## Sequence

### Position reached prior to cycle start:

The starting position is any position from which each of the long holes can be approached without collision.

### The cycle creates the following sequence of motions:

- Using G0, the starting position for the cycle is approached. In both axes of the current plane, the next end point of the first slot to be machined is approached at the height of the retraction plane in this applicator, and then the applicator is lowered to the reference plane brought forward by the safety clearance.
- Each long hole is milled in a reciprocating motion. The machining in the plane is performed using G1 and the feedrate programmed under FFP1. The infeed to the next machining depth calculated using G1 internally in the cycle and using feedrate is performed at each reversal point until the final depth is reached.

- Retraction to the retraction plane using G0 and approach to the next long hole on the shortest path.
- After the last long hole has been machined, the tool is moved with G0 to the position in the machining plane, which was reached last and which is specified in the diagram below, and the cycle is ended.

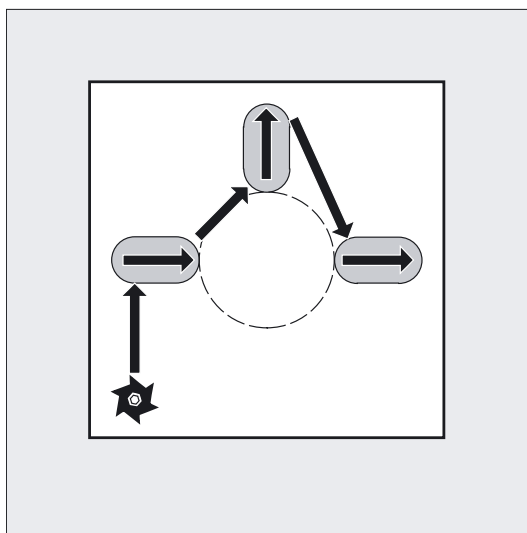


Figure 10-54 LONGHOLE sequence

### Explanation of the parameters

For parameters RTP, RFP, SDIS - see CYCLE81.

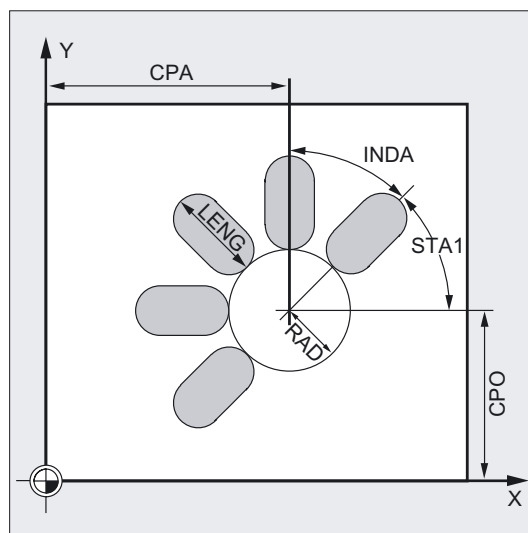


Figure 10-55 Explanation of the LONGHOLE parameters

### **DP and DPR (long hole depth)**

The depth of the long hole can be specified either absolute (DP) or relative (DPR) to the reference plane.

With relative specification, the cycle calculates the resulting depth automatically using the positions of reference and retraction planes.

### **NUM (number)**

Use the parameter NUM to specify the number of long holes.

### **LENG (long hole length)**

The length of the long hole is programmed under LENG.

If it is detected in the cycle that this length is smaller than the milling diameter, the cycle is aborted with alarm 61105 "Milling radius is too large".

### **MID (infeed depth)**

Use this parameter to define the maximum infeed depth.

The depth infeed is performed by the cycle in equally-sized infeed steps.

Using MID and the total depth, the cycle automatically calculates this infeed which lies between 0.5 x maximum infeed depth and the maximum infeed depth. The minimum possible number of infeed steps is used as the basis. MID=0 means that the cut to pocket depth is made with one feed.

The depth infeed starts from the reference plane brought forward by the safety clearance (depending on `_ZSD[1]`).

### **FFD and FFP1 (feedrate for depth and surface)**

The feedrate FFP1 is active for all movements in the plane traversed at feedrate. FFD acts for infeeds vertically to this plane.

### **CPA, CPO and RAD (center point and radius)**

You define the position of the circle in the machining plane by the center point (CPA, CPO) and the radius (RAD). Only positive values are permitted for the radius.



### **STA1 and INDA (starting and incremental angle)**

The arrangement of the long holes on the circle is defined by these parameters.

If INDA=0, the indexing angle is calculated from the number of long holes, so that they are equally distributed around the circle.

---

#### **Note**

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

If mutual contour violations of the slots result from incorrect values of the parameters that determine the arrangement and the size of the slots, the cycle will not start the machining. The cycle is aborted and the error message 61104 "Contour violation of slots/elongated holes" is output.

During the cycle, the workpiece coordinate system is offset and rotated. The values in the workpiece coordinate system are shown on the actual value display such that the longitudinal axis of the long hole being machined is positioned on the first axis of the current machining plane.

After the cycle has been completed, the workpiece coordinate system is in the same position again as it was before the cycle was called.

---

**Programming example: Machining slots**

By using this program, you can machine four slots of the length 30 mm and the relative depth 23 mm (difference between the reference plane and the slot root), which are arranged on a circle with the center point Y40 Z45 and the radius 20 mm in the YZ plane. The starting angle is 45 degrees, the incremental angle is 90 degrees. The maximum infeed depth is 6 mm, the safety clearance 1 mm.

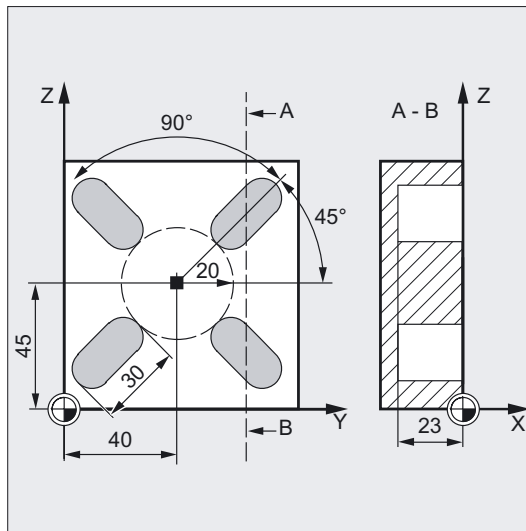


Figure 10-56 LONGHOLE example

```

N10 G19 G90 D9 T10 S600 M3 ; Specification of the
                                technological values
N20 G0 Y50 Z25 X5 ; Approach starting position
N30 LONGHOLE (5, 0, 1, , 23, 4, 30, 40, 45, 20, ; Cycle call
45, 90, 100 , 320, 6)
N40 M02 ; End of program
    
```

## 10.6.7 Slots on a circle - SLOT1

### Programming

SLOT1(RTP, RFP, SDIS, DP, DPR, NUM, LENG, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF)

### Parameter

Table 10- 19 SLOT1 parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Slot depth (absolute)
DPR	REAL	Slot depth relative to the reference plane (enter without sign)
NUM	INT	Number of slots
LENG	REAL	Slot length (enter without sign)
WID	REAL	Slot width (enter without sign)
CPA	REAL	Center point of circle (absolute), first axis of the plane
CPO	REAL	Center point of circle (absolute), second axis of the plane
RAD	REAL	Radius of the circle (enter without sign)
STA1	REAL	Starting angle
INDA	REAL	Incrementing angle
FFD	REAL	Feedrate for depth infeed
FFP1	REAL	Feedrate for surface machining
MID	REAL	Maximum infeed depth for one infeed (enter without sign)
CDIR	INT	Milling direction for machining the slot Values: 2 (for G2), 3 (for G3)
FAL	REAL	Finishing allowance at the slot edge (enter without sign)
VARI	INT	Machining type Values: 0 = complete machining, 1 = roughing, 2 = finishing
MIDF	REAL	Maximum infeed depth for finishing
FFP2	REAL	Feedrate for finishing
SSF	REAL	Speed when finishing

### Note

The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

## Function

The cycle SLOT1 is a combined roughing-finishing cycle.

Use this cycle to machine slots arranged on a circle. The longitudinal axis of the slots is aligned radially. In contrast to the long hole, a value is defined for the slot width.

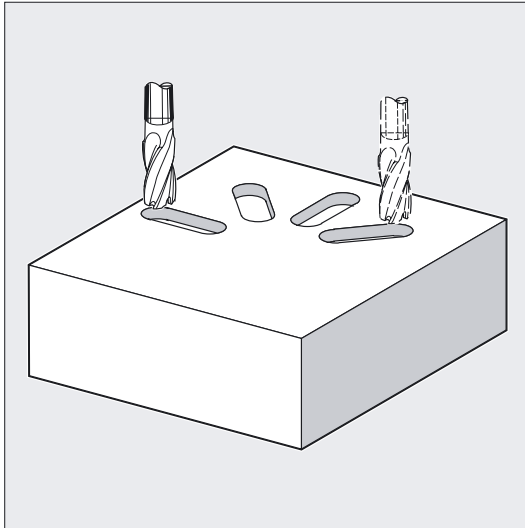


Figure 10-57 Slots on a circle - SLOT1

## Sequence

### Position reached prior to cycle start:

The starting position can be any position from which each of the slots can be approached without collision.

### The cycle creates the following sequence of motions:

- Approach of the position at the beginning of the cycle indicated in the "SLOT1 sequence" figure with G0.
- Complete machining of a slot is carried out in the following steps:
  - Approach of the reference plane brought forward by the safety clearance by using G0
  - Infeed to the next machining depth with G1 and with feedrate value FFD
  - Solid machining of the slot to the finishing allowance at the slot edge with feedrate value FFP1. Then finishing with feedrate value FFP2 and spindle speed SSF along the contour according to the machining direction programmed under CDIR.
  - The depth infeed is always carried out at the same position in the machining plane until the end depth of the slot is reached.

- Retract tool to the retraction plane and move to the next slot with G0.
- After the last slot has been machined, the tool is moved with G0 to the end position in the machining plane, which is specified in the diagram below, and the cycle is ended.

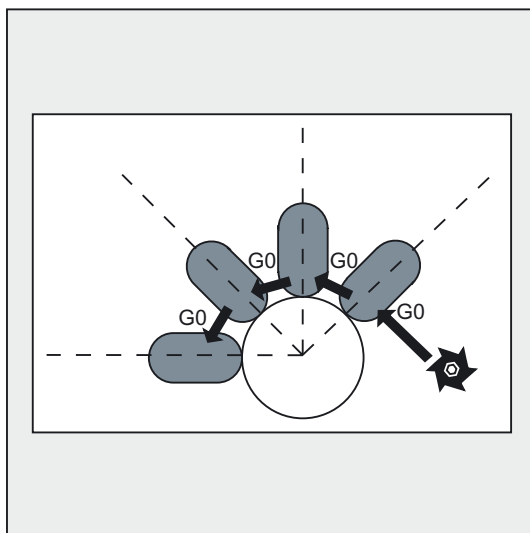


Figure 10-58 SLOT1 sequence

### Explanation of the parameters

For parameters RTP, RFP, SDIS - see CYCLE81.

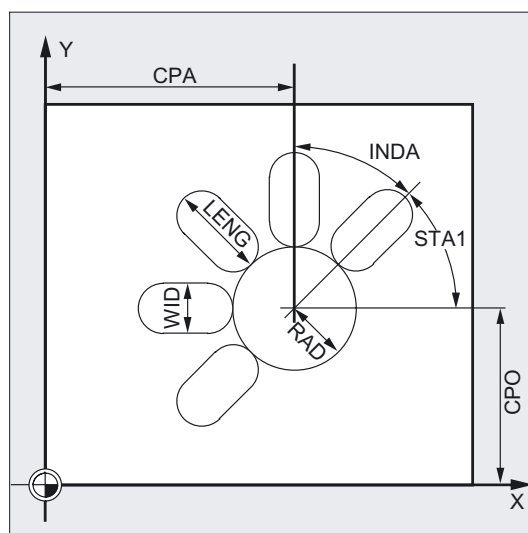


Figure 10-59 Explanation of the SLOT1 parameters

### DP and DPR (slot depth)

The slot depth can be specified either absolute (DP) or relative (DPR) to the reference plane.

With relative specification, the cycle calculates the resulting depth automatically using the positions of reference and retraction planes.

### **NUM (number)**

Use the parameter NUM to specify the number of slots.

### **LENG and WID (slot length and slot width)**

Use the parameters LENG and WID to define the form of a slot in the plane. The milling cutter diameter must be smaller than the slot width. Otherwise, alarm 61105 "Cutter radius too large" will be activated and the cycle aborted.

The milling cutter diameter must not be smaller than half of the groove width. This is not checked.

### **CPA, CPO and RAD (center point and radius)**

You define the position of the circle in the machining plane by the center point (CPA, CPO) and the radius (RAD). Only positive values are permitted for the radius.

### **STA1 and INDA (starting and incremental angle)**

The arrangement of the slot on the circle is defined by these parameters.

STA1 defines the angle between the positive direction of the first axis (abscissa) of the workpiece coordinate system active before the cycle was called and the first groove. Parameter INDA contains the angle from one slot to the next.

If INDA=0, the incrementing angle is calculated from the number of slots so that they are arranged equally around the circle.

### **FFD and FFP1 (feedrate for depth and surface)**

The feedrate FFD is active for all infeed movements perpendicular to the machining plane.

The feedrate FFP1 is active for all movements in the plane traversed at feedrate when roughing.

### **MID (infeed depth)**

Use this parameter to define the maximum infeed depth.

The depth infeed is performed by the cycle in equally-sized infeed steps.

Using MID and the total depth, the cycle automatically calculates this infeed which lies between 0.5 x maximum infeed depth and the maximum infeed depth. The minimum possible number of infeed steps is used as the basis. MID=0 means that the cut to slot depth is made with one feed.

The depth infeed commences at the reference plane moved forward by the safety clearance.

### **CDIR (milling direction)**

Use this parameter to specify the machining direction for the groove. Possible values are:

- "2" for G2
- "3" for G3

If the parameter is set to an illegal value, then the message "Wrong milling direction, G3 will be generated" will be displayed in the message line. In this case, the cycle is continued and G3 is automatically generated.

### **FAL (finishing allowance)**

Use this parameter to program a finishing allowance at the slot edge. FAL does not influence the depth infeed.

If the value of FAL is greater than allowed for the specified width and the milling cutter used, FAL is automatically reduced to the maximum possible value. In the case of roughing, milling is performed with a reciprocating movement and depth infeed at both end points of the slot.

### **VARI, MIDF, FFP2 and SSF (machining type, infeed depth, feedrate and speed)**

Use the parameter VARI to define the machining type.

Possible values are:

- 0=complete machining in two parts
  - Solid machining of the slot (SLOT1, SLOT2) to the finishing allowance is performed at the spindle speed programmed before the cycle was called and with feedrate FFP1. Depth infeed is defined with MID.
  - Solid machining of the remaining finishing allowance is carried out at the spindle speed defined via SSF and the feedrate FFP2. Depth infeed is defined with MIDF.  
If MIDF=0, the infeed is performed right to the final depth.
  - If FFP2 is not programmed, feedrate FFP1 is active. This also applies analogously if SSF is not specified, i.e. the speed programmed prior to the cycle call will apply.
- 1=Roughing

The groove (SLOT1, SLOT2) is solid-machined up to the finishing allowance at the speed programmed before the cycle call and at the feedrate FFP1. The depth infeed is programmed via MID.
- 2=Finishing

The cycle requires that the slot (SLOT1, SLOT2) is already machined to a residual finishing allowance and that it is only necessary to machine the final finishing allowance. If FFP2 and SSF are not programmed, the feedrate FFP1 or the speed programmed before the cycle call is active. Depth infeed is defined with MIDF.

If a different value is programmed for the parameter VARI, the cycle is aborted after output of alarm 61102 "Machining type defined incorrectly".

---

**Note**

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

If incorrect values are assigned to the parameters that determine the arrangement and size of the slots and thus cause mutual contour violation of the slots, the cycle is not started. The cycle is aborted and the error message 61104 "Contour violation of slots/elongated holes" is output.

During the cycle, the workpiece coordinate system is offset and rotated. The values in the workpiece coordinate system displayed on the actual value display are such that the longitudinal axis of the slot that has just been machined corresponds to the first axis of the current machining plane.

After the cycle has been completed, the workpiece coordinate system is in the same position again as it was before the cycle was called.

---

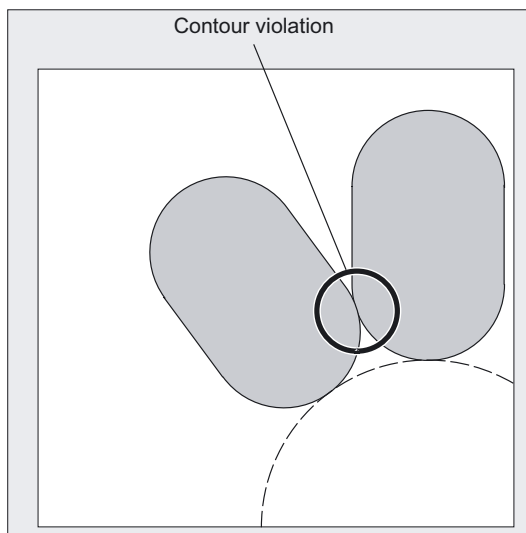


Figure 10-60 SLOT1 contour violation



### Programming example: Grooves

Four slots are milled.

The slots have the following dimensions: Length 30 mm, width 15 mm and depth 23 mm. The safety clearance is 1 mm, the final machining allowance is 0.5 mm, the milling direction is G2, the maximum infeed in the depth is 6 mm.

The slot is to be machined completely. Infeed during finishing is to be performed directly to the pocket depth and the same feedrate and speed are to be used.

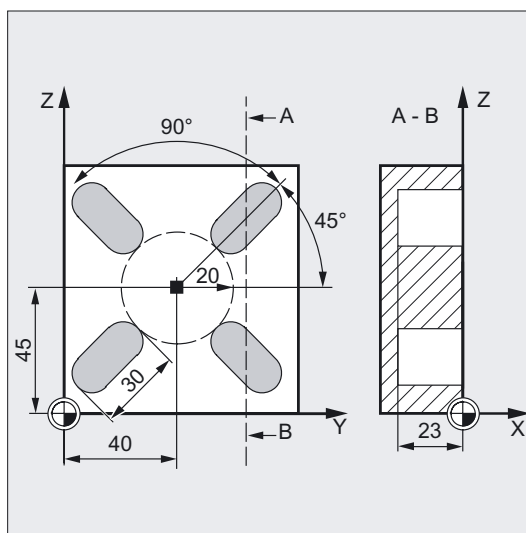


Figure 10-61 SLOT1 example

```

N10 G17 G90 T1 D1 S600 M3 ; Specification of technology
                               values
N20 G0 X20 Y50 Z5 ; Approach starting position
N30 SLOT1(5, 0, 1, -23, , 4, 30, 15, 40, 45, 20, ; Cycle call, VARI, MIDF, FFP2
45, 90, 100, 320, 6, 2, 0.5, 0, , 0, ) and SSF parameters omitted
N40 M02 ; End of program

```

## 10.6.8 Circumferential slot - SLOT2

### Programming

SLOT2(RTP, RFP, SDIS, DP, DPR, NUM, AFSL, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF)

### Parameter

Table 10- 20 SLOT2 parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Slot depth (absolute)
DPR	REAL	Slot depth relative to the reference plane (enter without sign)
NUM	INT	Number of slots
AFSL	REAL	Angle for the slot length (enter without sign)
WID	REAL	Circumferential slot width (enter without sign)
CPA	REAL	Center point of circle (absolute), first axis of the plane
CPO	REAL	Center point of circle (absolute), second axis of the plane
RAD	REAL	Radius of the circle (enter without sign)
STA1	REAL	Starting angle
INDA	REAL	Incrementing angle
FFD	REAL	Feedrate for depth infeed
FFP1	REAL	Feedrate for surface machining
MID	REAL	Maximum infeed depth for one infeed (enter without sign)
CDIR	INT	Milling direction for machining the circumferential slot Values: 2 (for G2), 3 (for G3)
FAL	REAL	Finishing allowance at the slot edge (enter without sign)
VARI	INT	Machining type Values: 0 = complete machining, 1 = roughing, 2 = finishing
MIDF	REAL	Maximum infeed depth for finishing
FFP2	REAL	Feedrate for finishing
SSF	REAL	Speed when finishing

### Note

The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

## Function

The cycle SLOT2 is a combined roughing-finishing cycle.

Use this cycle to machine circumferential slots arranged on a circle.

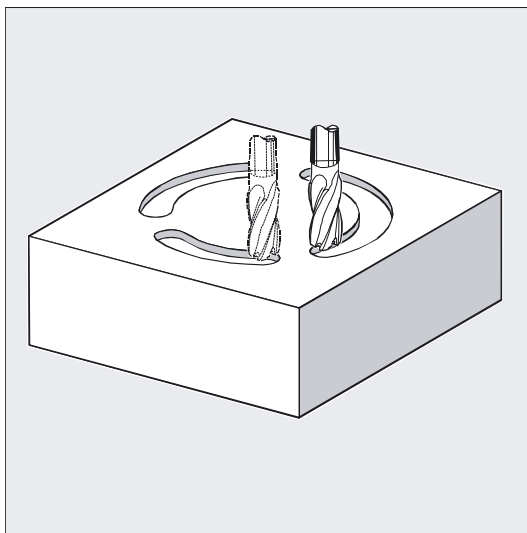


Figure 10-62 Circumferential slot - SLOTT2

## Sequence

### Position reached prior to cycle start:

The starting position can be any position from which each of the slots can be approached without collision.

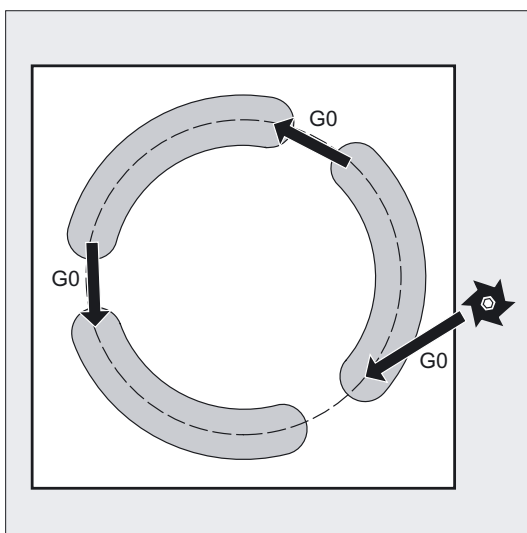


Figure 10-63 SLOTT2 sequence

**The cycle creates the following sequence of motions:**

- G0 is used to approach the position specified in the diagram below at cycle start.
- The steps when machining a circumferential slot are the same as when machining an elongated hole.
- After a circumferential slot is machined completely, the tool is retracted to the retraction plane and the next slot is machined with G0.
- After the last slot has been machined, the tool is moved with G0 to the end position in the machining plane, which is specified in the diagram below, and the cycle is ended.

**Explanation of the parameters**

For parameters RTP, RFP, SDIS - see CYCLE81.

For the parameters DP, DPR, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF - see SLOT1.

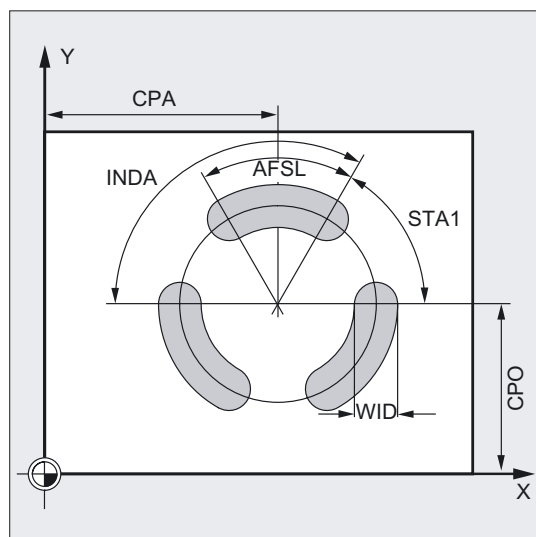


Figure 10-64 Explanation of the SLOT2 parameters

**NUM (number)**

Use the parameter NUM to specify the number of slots.

**AFSL and WID (angle and circumferential slot width)**

Use the parameters AFSL and WID to define the form of a slot in the plane. The cycle checks whether the slot width is violated with the active tool. Otherwise, alarm 61105 "Cutter radius too large" will be activated and the cycle aborted.

**CPA, CPO and RAD (center point and radius)**

You define the position of the circle in the machining plane by the center point (CPA, CPO) and the radius (RAD). Only positive values are permitted for the radius.

### STA1 and INDA (starting and incremental angle)

The arrangement of the circumferential slots on the circle is defined by these parameters.

STA1 defines the angle between the positive direction of the first axis (abscissa) of the workpiece coordinate system active before the cycle was called and the first slot.

The INDA parameter contains the angle from one circumferential slot to the next.

If INDA=0, the incremental angle is calculated from the number of circumferential slots so that they are arranged equally around the circle.

---

#### Note

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

If incorrect values are assigned to the parameters that determine the arrangement and size of the slots and thus cause mutual contour violation of the slots, the cycle is not started.

The cycle is aborted and the error message 61104 "Contour violation of slots/elongated holes" is output.

During the cycle, the workpiece coordinate system is offset and rotated. The actual value display in the workpiece coordinate system is always shown such that the circumferential slot currently being machined starts on the first axis of the current processing level and the zero point of the workpiece coordinate system is in the center of the circle.

After the cycle has been completed, the workpiece coordinate system is in the same position again as it was before the cycle was called.

---

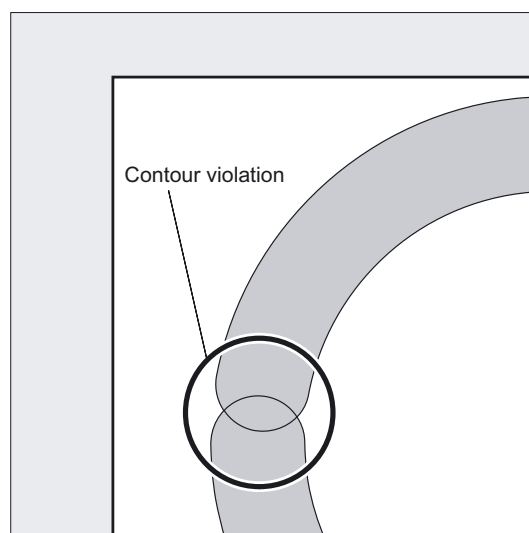


Figure 10-65 SLOT2 contour violation

### Programming example: Slots2

Use this program to machine three circumferential slots arranged at a circle with center point X60 Y60 and radius 42 mm in the XY plane. The circumferential slots have the following dimensions: Width 15 mm, angle for slot length 70 degrees, depth 23 mm. The initial angle is 0 degrees, the incremental angle is 120 degrees. The slot contours are machined to a final machining allowance of 0.5 mm, the safety clearance in infeed axis Z is 2 mm, the maximum depth infeed is 6 mm. The slots are to be completely machined. Speed and feedrate are to be the same when finishing. The infeed when finishing is to be performed to slot depth.

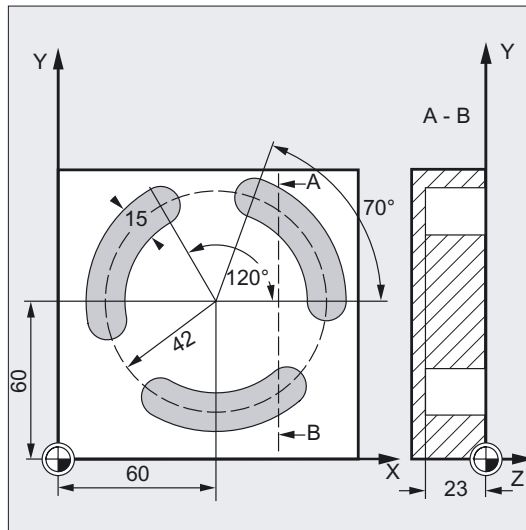


Figure 10-66 SLOT2 example

```

N10 G17 G90 T1 D1 S600 M3 ; Specification of technology
                               values
N20 G0 X60 Y60 Z5 ; Approach starting position
N30 SLOT2(2, 0, 2, -23, , 3, 70, 15, 60, 60, 42, ; Cycle call
, 120, 100, 300, 6, 2, 0.5, 0, , 0, )
                               Reference plane+SDIS=retraction
                               plane means: Lowering in the
                               infeed axis with G0 to reference
                               plane+SDIS no longer applicable,
                               parameters VAR, MIDF, FFP2 and
                               SSF omitted
N40 M02 ; End of program

```

## 10.6.9 Milling a rectangular pocket - POCKET3

### Programming

POCKET3 (RTP, RFP, SDIS, DP, LENG, WID, CRAD, PA, PO, STA, MID, FAL, FALD, FFP1, FFD, CDIR, VARI, MIDA, AP1, AP2, AD, RAD1, DP1)

### Parameter

Table 10- 21 POCKET3 parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Pocket depth (absolute)
LENG	REAL	Pocket length, for dimensioning from the corner with sign
WID	REAL	Pocket width, for dimensioning from the corner with sign
CRAD	REAL	Pocket corner radius (enter without sign)
PA	REAL	Reference point for the pocket (absolute), first axis of the plane
PO	REAL	Reference point for the pocket (absolute), second axis of the plane
STA	REAL	Angle between the pocket longitudinal axis and the first axis of the plane (enter without sign); Range of values: $0^\circ \leq STA < 180^\circ$
MID	REAL	Maximum infeed depth (enter without sign)
FAL	REAL	Finishing allowance at the pocket edge (enter without sign)
FALD	REAL	Finishing allowance at the base (enter without sign)
FFP1	REAL	Feedrate for surface machining
FFD	REAL	Feedrate for depth infeed

Parameter	Data type	Description
CDIR	INT	Milling direction: (enter without sign) Values: 0 Down-cut milling (in the spindle direction) 1 Conventional milling 2 With G2 (independent of spindle direction) 3 With G3
VARI	INT	Machining type UNITS DIGIT Values: 1 roughing, 2 finishing TENS DIGIT: Values: 0 Perpendicular to the pocket center with G0 1 Perpendicular to the pocket center with G1 2 Along a helix 3 Oscillation along the pocket longitudinal axis

The other parameters can be selected as options. Specify the plunge-cut strategy and the overlap for solid machining (to be entered without sign):

MIDA	REAL	Maximum infeed width as a value in solid machining in the plane
AP1	REAL	Blank dimension of pocket length
AP2	REAL	Blank dimension of pocket width
AD	REAL	Blank pocket depth dimension from reference plane
RAD1	REAL	Radius of the helical path on insertion (relative to the tool center point path) or maximum insertion angle for reciprocating motion
DP1	REAL	Insertion depth per 360° revolution on insertion along helical path

## Function

The cycle can be used for roughing and finishing. For finishing, a face cutter is required.

The depth infeed will always start at the pocket center point and be performed vertically from there; thus it is practical to predrill at this position.

- The milling direction can be determined either using a G command (G2/G3) or from the spindle direction as synchronous or up-cut milling.
- For solid machining, the maximum infeed width in the plane can be programmed.
- Finishing allowance also for the pocket base
- There are three different insertion strategies:
  - vertically to the pocket center
  - along a helical path around the pocket center
  - oscillating at the pocket central axis



- Shorter approach paths in the plane for finishing
- Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).

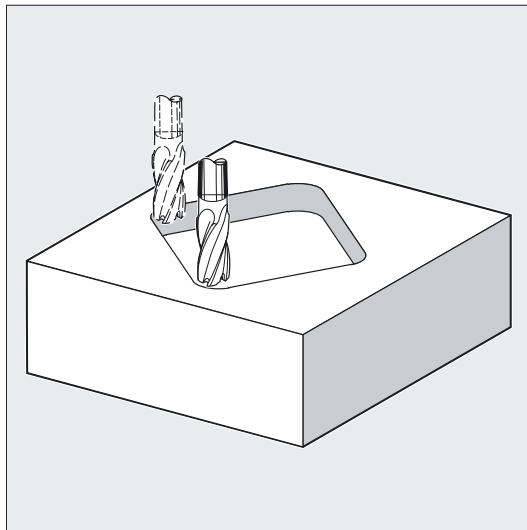


Figure 10-67 Milling a rectangular pocket - POCKET3

## Sequence

### Position reached prior to cycle start:

Starting position is any position from which the pocket center point can be approached at the height of the retraction plane without collision.

### Sequence of motions when roughing:

With G0, the pocket center point is approached at the retraction level, and then, from this position, with G0, too, the reference plane brought forward by the safety clearance is approached. The machining of the pocket is then carried out according to the selected insertion strategy, taking into account the programmed blank dimensions.

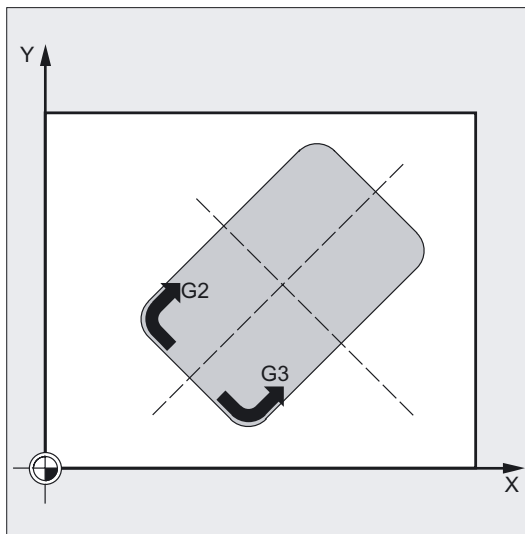


Figure 10-68 Sequence of motions when roughing

#### Sequence of motions when finishing:

Finishing is performed in the order from the edge until the finishing allowance on the base is reached, and then the base is finished. If one of the finishing allowances is equal to zero, this part of the finishing process is skipped.

- Finishing on the edge

While finishing on the edge, the tool traverses around the pocket contour only once.

For finishing on the edge, the path includes one quadrant reaching the corner radius. The radius of this path is normally 2 mm or, if "less space" is provided, equals to the difference between the corner radius and the mill radius.

If the final machining allowance on the edge is larger than 2 mm, the approach radius is increased accordingly.

The depth infeed is performed with G0 in the open towards the pocket center, and the starting point of the approach path is also reached with G0.

- Finishing on the base

During finishing on the base, the machine performs G0 towards the pocket center until reaching a distance equal to pocket depth + finishing allowance + safety clearance. From this point onwards, the tool is always fed in **vertically** at the depth (since a tool with a front cutting edge is used for base finishing).

The base surface of the pocket is machined once.

## Insertion strategies

- Inserting vertically to the pocket center means that the current infeed depth calculated internally in the cycle ( $\leq$  maximum infeed depth programmed under MID) is executed in a block containing G0 or G1.
- Insertion at a helical path means that the cutter center point traverses along the helical path determined by the radius RAD1 and the depth per revolution DP1. The feedrate is also programmed under FFD. The direction of rotation of this helical path corresponds to the direction of rotation with which the pocket will be machined.

The insertion depth programmed under DP1 is taken into account as the maximum depth and is always calculated as an integer number of revolutions of the helical path.

If the current depth required for an infeed (this can be several revolutions on the helical path) is reached, a full circle is still executed to eliminate the inclined path of insertion.

Pocket solid machining then starts in this plane and continues until it reaches the final machining allowance.

The starting point of the described helical path is at the longitudinal axis of the pocket in "plus direction" and is approached with G1.

- Insertion with oscillation to the central axis of the pocket means that the cutter center point is inserted oscillating on a straight line until it reaches the next current depth. The maximum immersion angle is programmed under RAD1, and the length of the oscillation travel is calculated in the cycle. If the current depth is reached, the travel is executed once more without depth infeed in order to eliminate the inclined insertion path. The feedrate is programmed under FFD.

## Taking into account the blank dimensions

During solid machining of the pockets, it is possible to take into account blank dimensions (e.g. when machining precast parts).

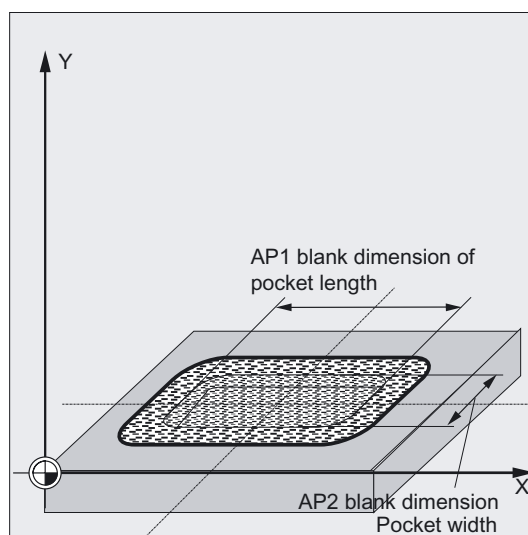


Figure 10-69 Taking into account the blank dimensions

The basic sizes for the length and width (AP1 and AP2) are programmed without sign and their symmetrical positions around the pocket center point are computed in the cycle. You define the part of the pocket which is no longer to be machined by solid machining. The

blank dimension for the depth (AD) is also programmed without sign and taken into account by the reference plane in the direction of the pocket depth.

The depth infeed when taking into account blank dimensions is carried out according to the programmed type (helical path, reciprocating, vertically). If the cycle detects that there is space enough in the pocket center because of the given blank contour and the radius of the active tool, the infeed is carried out vertically to the pocket center point as long as it is possible in order not to traverse extensive insertion paths in the open.

Solid machining of the pocket is carried out starting from the top downwards.

### Explanation of the parameters

For parameters RTP, RFP, SDIS - see CYCLE81.

For the DP parameter, see LONGHOLE.

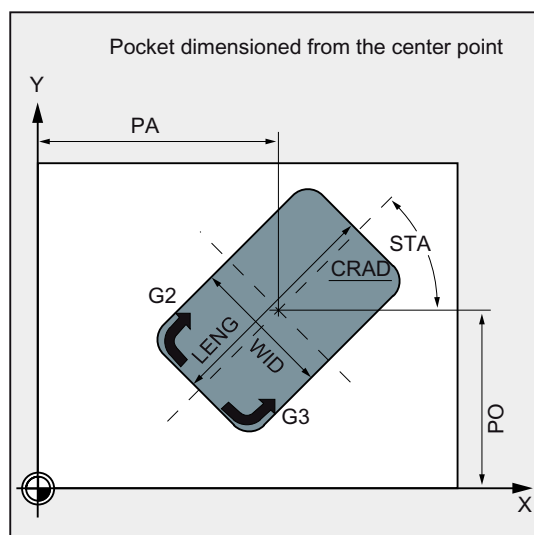


Figure 10-70 Explanation of the POCKET3 parameters

### LENG, WID and CRAD (pocket length, pocket width and corner radius)

Use the parameters LENG, WID and CRAD to define the form of a pocket in the plane.

If you cannot traverse the programmed corner radius with the active tool since its radius is larger, then the corner radius of the machine pocket corresponds to the tool radius.

If the milling tool radius is larger than half of the length or width of the pocket, then the cycle will be aborted and alarm 61105 "Cutter radius too large" is output.

### PA, PO (reference point)

Use the parameters PA and PO to define the reference point of the pocket in the axes of the plane. This is the pocket center point.

### STA (angle)

STA indicates the angle between the first axis of the plane (abscissa) and the longitudinal axis of the pocket.

### MID (infeed depth)

Use this parameter to define the maximum infeed depth when roughing.

The depth infeed is performed by the cycle in equally-sized infeed steps.

By using MID and the entire depth, the cycle calculates this infeed automatically. The minimum possible number of infeed steps is used as the basis.

MID=0 means that the cut to pocket depth is made with one feed.

### FAL (finishing allowance at edge)

The finishing allowance only affects the machining of the pocket in the plane on the edge.

If the final machining allowance  $\geq$  tool diameter, the pocket will not necessarily be machined completely. The message "Caution: final machining allowance  $\geq$  tool diameter" appears; the cycle, however, is continued.

### FALD (finishing allowance at the base)

When roughing, a separate finishing allowance is taken into account at the base.

### FFD and FFP1 (feedrate for depth and surface)

The feedrate FFD is effective when inserting into the material.

The feedrate FFP1 is active for all movements in the plane traversed at feedrate when machining.

### CDIR (milling direction)

Use this parameter to specify the machining direction for the pocket.

Using the parameter CDIR, the milling direction

- can be programmed directly with "2 for G2" and "3 for G3" or,
- alternatively with "synchronous milling" or "conventional milling".

Synchronized operation or reverse rotation are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

#### **Down-cut and up-cut milling:**

M3 → G3, M3 → G2

M4 → G2, M4 → G3

### VARI (machining type)

Use the parameter VARI to define the machining type.

Possible values are:

Units digit:

- 1=roughing
- 2=finishing

Tens digit (infeed):

- 0=vertically to pocket center with G0
- 1=vertically to pocket center with G1
- 2=along a helical path
- 3=oscillating to pocket length axis

If a different value is programmed for the parameter VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

### MIDA (max. infeed width)

Use this parameter to define the maximum infeed width when solid machining in a plane. Analogously to the known calculation method for the infeed depth (equal distribution of the total depth with maximum possible value) the width is distributed equally, maximally with the value programmed under MIDA.

If this parameter is not programmed or has value 0, the cycle will internally use 80% of the milling tool diameter as the maximum infeed width.

---

#### Note

Applies if the calculated width infeed from edge machining is recalculated when reaching the full pocket in the depth; otherwise the width infeed calculated at the beginning is kept for the whole cycle.

---

### AP1, AP2, AD (blank dimensions)

Use the parameters AP1, AP2 and AD to define the blank dimensions (incremental) of the pocket in the plane and in the depth.

### RAD1 (radius)

Use the RAD1 parameter to define the radius of the helical path (relative to the tool center point path) or the maximum insertion angle for the reciprocating motion.

### DP1 (insertion depth)

Use the parameter DP1 to define the infeed depth when inserting to the helical path.

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point. At the end of the cycle, the original coordinate system is active again.

### Programming example: Pocket

Use this program to machine a pocket in the XY plane which is 60 mm in length, 40 mm in width, and which has a corner radius of 8 mm and is 17.5 mm in depth. The pocket has an angle of 0 degrees to the X axis. The final machining allowance of the pocket edges is 0.75 mm, 0.2 mm at the base, the safety clearance in the Z axis, which is added to the reference plane, is 0.5 mm. The center point of the pocket lies at X60 and Y40, the maximum depth infeed is 4 mm.

The machining direction results from the direction of rotation of the spindle in the case of down-cut milling. A milling cutter with 5 mm radius is used.

Merely a rough machining operation is to be carried out.

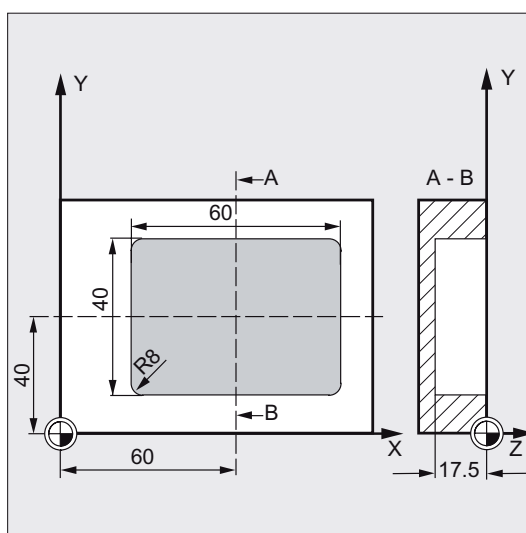


Figure 10-71 POCKET3 example

```

N10 G90 T1 D1 S600 M4 ; Specification of technology
                           values
N20 G17 G0 X60 Y40 Z5 ; Approach starting position
N30 POCKET3(5, 0, 0.5, -17.5, 60, 40, 8, 60, 40, ; Cycle call
0, 4, 0.75, 0.2, 1000, 750, 0, 11, 5, , , , )
N40 M02 ; End of program

```

## 10.6.10 Milling a circular pocket - POCKET4

### Programming

POCKET4 (RTP, RFP, SDIS, DP, PRAD, PA, PO, MID, FAL, FALD, FFP1, FFD, CDIR, VARI, MIDA, AP1, AD, RAD1, DP1)

### Parameter

Table 10- 22 POCKET4 parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (to be added to the reference plane; enter without sign)
DP	REAL	Pocket depth (absolute)
PRAD	REAL	Pocket radius
PA	REAL	Starting point (absolute), first axis of the plane
PO	REAL	Starting point (absolute), second axis of the plane
MID	REAL	Maximum infeed depth (enter without sign)
FAL	REAL	Finishing allowance at the pocket edge (enter without sign)
FALD	REAL	Finishing allowance at the base (enter without sign)
FFP1	REAL	Feedrate for surface machining
FFD	REAL	Feedrate for depth infeed
CDIR	INT	Milling direction: (enter without sign) Values: 0 Down-cut milling (in the spindle direction) 1 Conventional milling 2 With G2 (independent of spindle direction) 3 With G3
VARI	INT	Machining type UNITS DIGIT Values: 1 roughing, 2 finishing TENS DIGIT: Values: 0 Perpendicular to the pocket center with G0 1 Perpendicular to the pocket center with G1 2 Along a helix

The other parameters can be selected as options. Specify the plunge-cut strategy and the overlap for solid machining (to be entered without sign):



MIDA	REAL	Maximum infeed width as a value in solid machining in the plane
AP1	REAL	Pocket radius blank dimension
AD	REAL	Blank pocket depth dimension from reference plane
RAD1	REAL	Radius of the helical path during insertion (relative to the tool center point path)
DP1	REAL	Insertion depth per 360° revolution on insertion along helical path

## Function

Use this cycle to machine circular pockets in the machining plane. For finishing, a face cutter is required.

The depth infeed always starts at the pocket center point and be performed vertically from there; thus it is practical to predrill at this position.

- The milling direction can be determined either using a G command (G2/G3) or from the spindle direction as synchronous or up-cut milling.
- For solid machining, the maximum infeed width in the plane can be programmed.
- Finishing allowance also for the pocket base.
- Two different insertion strategies:
  - vertically to the pocket center
  - along a helical path around the pocket center
- Shorter approach paths in the plane for finishing
- Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).
- MIDA is recalculated during edge machining.

## Sequence

### Position reached prior to cycle start:

Starting position is any position from which the pocket center point can be approached at the height of the retraction plane without collision.

Motion sequence when roughing (VARI=X1):

With G0, the pocket center point is approached at the retraction level, and then, from this position, with G0, too, the reference plane brought forward by the safety clearance is approached. The machining of the pocket is then carried out according to the selected insertion strategy, taking into account the programmed blank dimensions.

Sequence of motions when finishing:

Finishing is performed in the order from the edge until the finishing allowance on the base is reached, and then the base is finished. If one of the finishing allowances is equal to zero, this part of the finishing process is skipped.

- Finishing on the edge

While finishing on the edge, the tool traverses around the pocket contour only once.

For finishing on the edge, the path includes one quadrant reaching the pocket radius. The radius of this path is 2 mm as the maximum or, if "less space" is provided, equals to the difference between the pocket radius and the milling radius.

The depth infeed is performed with G0 in the open towards the pocket center, and the starting point of the approach path is also reached with G0.

- Finishing on the base

During finishing on the base, the machine performs G0 towards the pocket center until reaching a distance equal to pocket depth + finishing allowance + safety clearance. From this point onwards, the tool is always fed in **vertically** at the depth (since a tool with a front cutting edge is used for base finishing).

The base surface of the pocket is machined once.

### Insertion strategies

see Section "POCKET3"

### Taking into account the blank dimensions

During solid machining of the pockets, it is possible to take into account blank dimensions (e.g. when machining precast parts).

With circular pockets, the blank dimension AP1 is also a circle (with a smaller radius than the pocket radius).

For further explanations, see POCKET3.

## Explanation of the parameters

For parameters RTP, RFP, SDIS - see CYCLE81

For parameters DP, MID, FAL, FALD, FFP1, FFD, CDIR, MIDA, AP1, AD, RAD1, DP1 - see POCKET3.

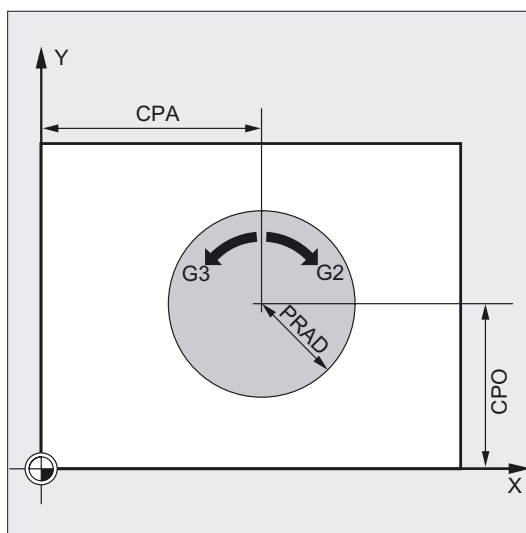


Figure 10-72 Explanation of the POCKET4 parameters

### PRAD (pocket radius)

The form of the circular pocket is determined solely by its radius.

If this is smaller than the tool radius of the active tool, then the cycle is aborted and alarm 61105 "Cutter radius too large" is output.

### PA, PO (pocket center point)

Use the parameters PA and PO to define the pocket center point. Circular pockets are always dimensioned across the center.

### VARI (machining type)

Use the parameter VARI to define the machining type.

Possible values are:

Units digit:

- 1=roughing
- 2=finishing

Tens digit (infeed):

- 0=vertically to pocket center with G0
- 1=vertically to pocket center with G1
- 2=along a helical path

If a different value is programmed for the parameter VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

### Note

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original coordinate system is active again.

### Programming example: Circular pocket

With this program, you can machine a circular pocket in the YZ plane. The center point is determined by Y50 Z50. The infeed axis for the depth infeed is the X axis. Neither finishing dimension nor safety clearance is specified. The pocket is machined with down-cut milling. Infeed is performed along a helical path.

A milling cutter with 10 mm radius is used.

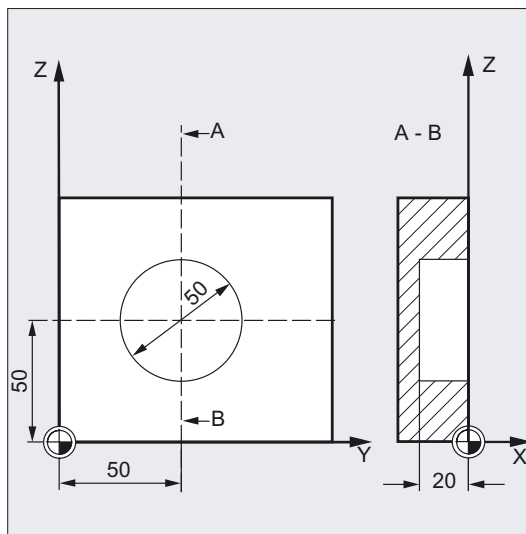


Figure 10-73 POCKET4 example

```

N10 G17 G90 G0 S650 M3 T1 D1           ; Specification of technology
                                         values
N20 X50 Y50                             ; Approach starting position
N30 POCKET4(3, 0, 0, -20, 25, 50, 60, 6, 0, 0, 0, 0, 0, 0, 2, 3) ; Cycle call
                                         Parameters FAL and FALD are
                                         omitted
N40 M02                                   ; End of program

```

## 10.6.11 Thread milling - CYCLE90

### Programming

CYCLE90 (RTP, RFP, SDIS, DP, DPR, DIATH, KDIAM, PIT, FFR, CDIR, TYPTH, CPA, CPO)

### Parameter

Table 10- 23 Parameters for CYCLE90

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)
DIATH	REAL	Nominal diameter, outer diameter of the thread
KDIAM	REAL	Core diameter, internal diameter of the thread
PST	REAL	Thread pitch; value range: 0.001 ... 2000.000 mm
FFR	REAL	Feedrate for thread milling (enter without sign)
CDIR	INT	Direction of rotation for thread milling Values: 2 (for thread milling with G2), 3 (for thread milling with G3)
TYPTH	INT	Thread type Values: 0=internal thread, 1=external thread
CPA	REAL	Center point of circle, abscissa (absolute)
CPO	REAL	Center point of circle, ordinate (absolute)

**Function**

By using the cycle CYCLE90, you can produce internal or external threads. The path when milling threads is based on a helix interpolation. All three geometry axes of the current plane, which you define before calling the cycle, are involved in this motion.

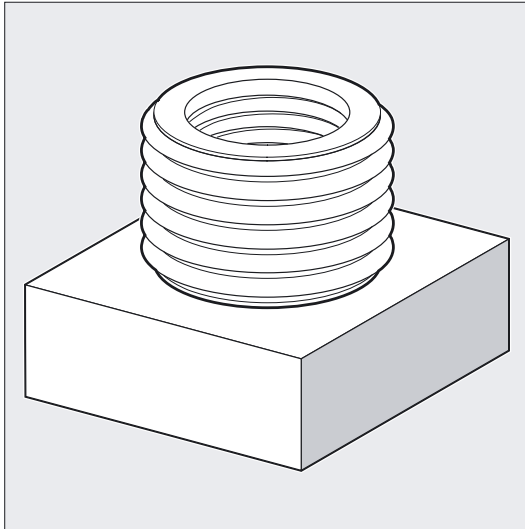


Figure 10-74 Thread milling - CYCLE90

**Sequence for external thread****Position reached prior to cycle start:**

The starting position is any position from which the starting position at the outside diameter of the thread at the height of the retraction plane can be reached without collision.

This start position for thread milling with G2 lies between the positive abscissa and the positive ordinate in the current level (i.e. in the first quadrant of the coordinate system). For thread milling with G3, the start position lies between the positive abscissa and the negative ordinate (namely in the fourth quadrant of the coordinate system).

The distance from the thread diameter depends on the size of the thread and the tool radius used.

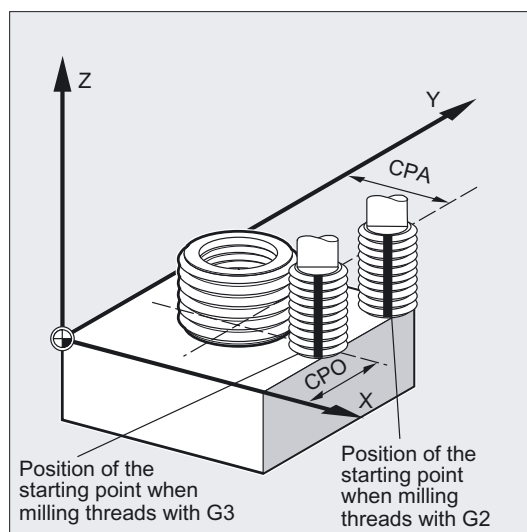


Figure 10-75 CYCLE90 sequence

**The cycle creates the following sequence of motions:**

- Positioning on the starting point using G0 at the height of the retraction plane in the applicator of the current plane
- Infeed to the reference plane brought forward by the safety clearance for swarf removal, using G0
- Approach motion to the thread diameter along a circle path opposite to the direction G2/G3 programmed under CDIR
- Thread milling along a helix path using G2/G3 and the feedrate value FFR
- Retraction motion along a circle path in the opposite direction of rotation G2/G3 at the reduced feedrate FFR
- Retraction to the retraction plane along the applicator using G0

**Sequence for internal thread**

**Position reached prior to cycle start:**

The starting position is any position from which the center point of the thread at the height of the retraction plane can be reached without collision.

**The cycle creates the following sequence of motions:**

- Positioning on the center point using G0 at the height of the retraction plane in the applicator of the current plane
- Infeed to the reference plane brought forward by the safety clearance for swarf removal, using G0
- Approach to an approach circle calculated internally in the cycle using G1 and the reduced feedrate FFR
- Approach motion to the thread diameter along a circle path according to the direction G2/G3 programmed under CDIR
- Thread milling along a helix path using G2/G3 and the feedrate value FFR

- Retraction motion along a circle path in the same direction of rotation at the reduced feedrate FFR
- Retraction to the center point of the thread using G0
- Retraction to the retraction plane along the applicator using G0

### Thread from bottom to top

For technological reasons, it can also be reasonable to machine a thread from bottom to top. In this case, the retraction plane RTP will be behind the thread depth DP.

This machining is possible, but the depth specifications must be programmed as absolute values and the retraction plane must be approached before calling the cycle or a position after the retraction plane must be approached.

### Programming example (thread from bottom to top)

A thread with a pitch of 3 mm is to start from -20 and to be milled to 0. The retraction plane is at 8.

```
N10 G17 X100 Y100 S300 M3 T1 D1 F1000
N20 Z8
N30 CYCLE90 (8, -20, 0, -60, 0, 46, 40, 3, 800,
3, 0, 50, 50)
N40 M2
```

The hole must have a depth of at least -21.5 (half pitch in excess).

### Overshooting in the direction of the thread length

For thread milling, the travel-in and travel-out movements occur along all three axes concerned. This means that the travel-out movement includes a further step in the vertical axis, beyond the programmed thread depth.

The overshoot is calculated as follows:

$$\Delta z = \frac{p}{4} * \frac{2 * WR + RDIFF}{DIATH}$$

$\Delta z$ : Overshoot, internal

p: Pitch

WR: Tool radius

DIATH: External diameter of the thread

RDIFF: Radius difference for travel-out circle

For internal threads,  $RDIFF = DIATH/2 - WR$ ; for external threads,  $RDIFF = DIATH/2 + WR$ .



## Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

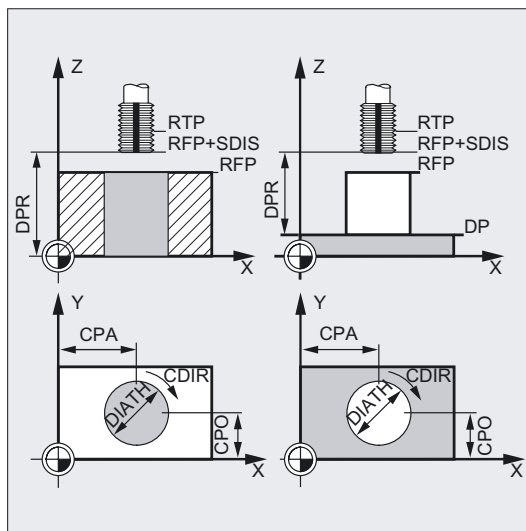


Figure 10-76 Explanation of the CYCLE90 parameters

## DIATH, KDIAM, and PIT (nominal diameter, core diameter, and thread pitch)

These parameters are used to determine the thread data nominal diameter, core diameter, and pitch. The parameter DIATH is the external, and KDIAM is the internal diameter of the thread. The travel-in / travel-out movements are created internally in the cycle, based on these parameters.

## FFR (feedrate)

The value of the parameter FFR is specified as the current feedrate value for thread milling. It is effective when thread milling on a helical path.

This value will be reduced in the cycle for the travel-in / travel-out movements. The retraction is performed outside the helix path using G0.

## CDIR (direction of rotation)

This parameter is used to specify the value for the machining direction of the thread.

If the parameter has an illegal value, the following message will appear:

"Wrong milling direction; G3 is generated".

In this case, the cycle is continued and G3 is automatically generated.

## TYPTH (thread type)

The parameter TYPTH is used to define whether you want to machine an external or an internal thread.

**CPA and CPO (center point)**

These parameters are used to define the center point of the drill hole or of the spigot on which the thread will be produced.

**Note**

The cutter radius is calculated internally in the cycle. Therefore, a tool compensation must be programmed before calling the cycle. Otherwise, the alarm 61000 "No tool compensation active" appears and the cycle is aborted.

If the tool radius=0 or negative, the cycle is also aborted and this alarm is issued.

With internal threads, the tool radius is monitored and alarm 61105 "Cutter radius too large" is output, and the cycle is aborted.

**Programming example: Internal thread**

By using this program, you can mill an internal thread at point X60 Y50 of the G17 plane.

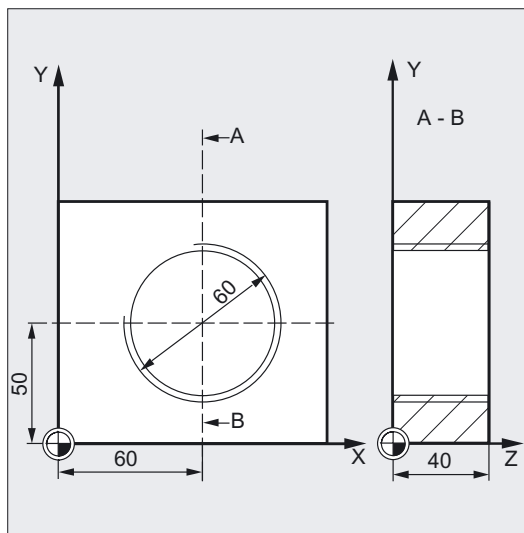


Figure 10-77 CYCLE90 example

```

DEF REAL RTP=48, RFP=40, SDIS=5, DPR=40,           ; Definition of the variable with
DIATH=60, KDIAM=50                                 value assignments
DEF REAL PIT=2, FFR=500, CPA=60,CPO=50
DEF INT CDIR=2, TYPTH=0
N10 G90 G0 G17 X0 Y0 Z80 S200 M3                   ; Approach starting position
N20 T5 D1                                           ; Specification of technology
                                                    values
N30 CYCLE90 (RTP, RFP, SDIS, DP, DPR, DIATH,       ; Cycle call
KDIAM, PIT, FFR, CDIR, TYPTH, CPA CPO)
N40 G0 G90 Z100                                     ; Approach position after cycle
N50 M02                                           ; End of program

```

## 10.7 Error messages and error handling

### 10.7.1 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 messages do 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.

### 10.7.2 Error handling in the cycles

If error conditions are detected in the cycles, an alarm is generated and the machining is aborted.

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.

### 10.7.3 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

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 number	Alarm text	Source	Explanation, Remedy
61000	"No tool offset active"	SLOT1 SLOT2 POCKET3 POCKET4 CYCLE71 CYCLE72	D offset must be programmed prior to cycle call.
61001	"Illegal thread lead"	CYCLE84 CYCLE840	Check the parameters for the thread size or the specifications for the lead (mutually conflicting).
61002	"Machining type defined incorrectly"	SLOT1 SLOT2 POCKET3 POCKET4 CYCLE71 CYCLE72	The value of parameters VARI for the machining type is specified incorrectly and must be changed.
61003	"No feedrate programmed in cycle"	CYCLE71 CYCLE72	The parameter for the feed is incorrectly specified and must be changed.
61009	"Active tool number = 0"	CYCLE71 CYCLE72	No tool (T) is programmed prior to the cycle call.
61010	"Final machining allowance too large"	CYCLE72	The finishing allowance at the root is larger than the overall depth; it must be reduced.
61011	"Scaling not allowed"	CYCLE71 CYCLE72	A scaling factor not permissible for this cycle is active.
61101	"Reference plane defined incorrectly"	CYCLE71 CYCLE72 CYCLE81 to CYCLE89 CYCLE840 SLOT1 SLOT2 POCKET3 POCKET4	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"	CYCLE86 CYCLE88 CYCLE840 POCKET3 POCKET4	The parameter SDIR (or SDR in CYCLE840) must be programmed.
61103	"Number of holes is zero"	HOLES1 HOLES2	No value for the number of holes programmed.

Alarm number	Alarm text	Source	Explanation, Remedy
61104	"Contour violation of the slots / elongated holes"	SLOT1 SLOT2	Faulty parameterization of the milling pattern in the parameters defining the position of the slots/elongated holes on the circle and their shape.
61105	"Cutter radius too large"	SLOT1 SLOT2 POCKET3 POCKET4	The diameter of the milling cutter being used is too large for the figure that is to be machined; either a tool with a smaller radius must be used or the contour must be changed.
61106	"Number or distance of circle elements"	HOLES2 SLOT1 SLOT2	Parameters NUM or INDA parameterized incorrectly; the arrangement of the circle elements within a full circle is not possible.
61107	"First drilling depth defined incorrectly"	CYCLE83	First drilling depth is opposite to total drilling depth.
61108	"No valid settings for parameters _RAD1 and _DP1"	POCKET3 POCKET4	The parameters _RAD1 and _DP for defining the path for depth infeed were incorrectly specified.
61109	"Parameter _CDIR incorrectly defined"	POCKET3 POCKET4	The value of the _CDIR parameter for the milling direction was specified incorrectly and must be changed.
61110	"Finishing allowance on base > infeed depth"	POCKET3 POCKET4	The final machining allowance on the base has been set to a higher value than the maximum depth infeed; either reduce final machining allowance or increase depth infeed.
61111	"Infeed width > tool diameter"	CYCLE71 POCKET3 POCKET4	The programmed infeed width is larger than the diameter of the active tool; it must be reduced.
61112	Tool radius negative"	CYCLE72	The radius of the active tool is negative; this is not permissible.
61113	"Parameter _CRAD for corner radius too large"	POCKET3	The parameter for the corner radius _CRAD was specified too large; it must be reduced.
61114	"Direction of machining G41/G42 incorrectly defined"	CYCLE72	The machining direction of the cutter radius path compensation G41/G42 was selected incorrectly.
61115	"Approach or retraction mode (straight line/circle/plane/space) defined incorrectly"	CYCLE72	The contour approach or return mode has been incorrectly defined; check the _AS1 or _AS2 parameters.
61116	"Approach or return path=0"	CYCLE72	The approach or return path is set to zero and must be increased; check the _LP1 or _LP2 parameters.
61117	"Active tool radius <= 0"	CYCLE71 POCKET3 POCKET4	The radius of the active tool is negative or zero; this is not permissible!
61118	"Length or width = 0"	CYCLE71	The length or width of the milling surface is not permissible; check the _LENG and _WID parameters.

Alarm number	Alarm text	Source	Explanation, Remedy
61124	"Infeed width is not programmed"	CYCLE71	With the simulation active without tool, a value for the infeed width _MIDA must always be programmed.
62100	"No drilling cycle active"	HOLES1 HOLES2	No drilling cycle has been called modally before the drilling cycle has been called.

#### 10.7.4 Messages in the cycles

The cycles display their messages in the message line of the control system. These messages do 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"	CYCLE81...CYCLE89, CYCLE840
"Slot is being processed"	SLOT1
"Circumferential slot is being processed"	SLOT2
"Wrong milling direction; G3 is generated".	SLOT1, SLOT2
First drilling depth: according to the value for the relative depth"	CYCLE83

In each case, <No.> stands for the number of the figure that is currently being machined.

# Network operation

## 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

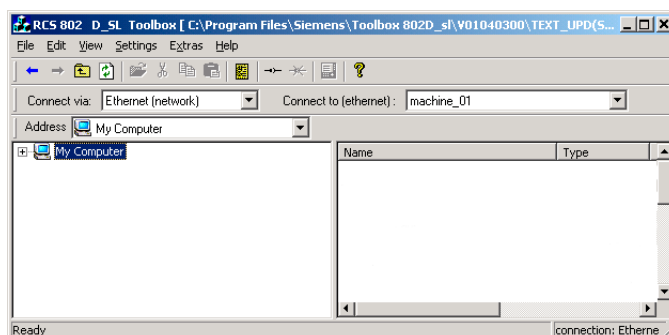


Figure 11-1 Explorer window of the RCS802 tool

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.

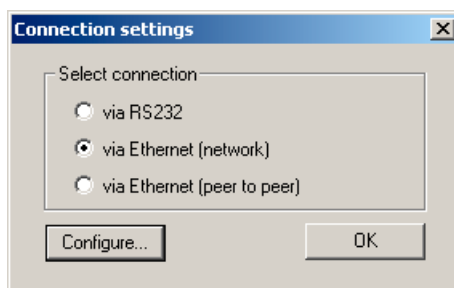


Figure 11-2 Connection Settings

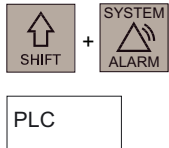
---

### 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



- You are now in the <SYSTEM> operating area.
- Press the "PLC" softkey.

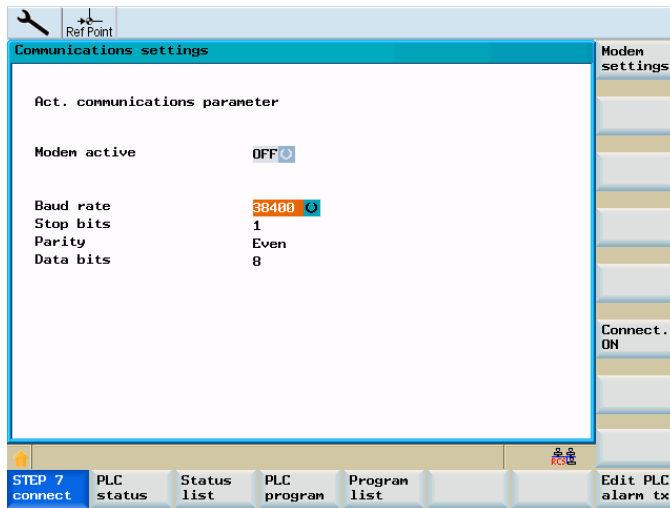
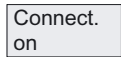


Figure 11-3 Communication settings RS232



- Set the parameters for communication in the "STEP 7 Connect" dialog.



- Activate the RS232 connection with the "Connect. ON" softkey.

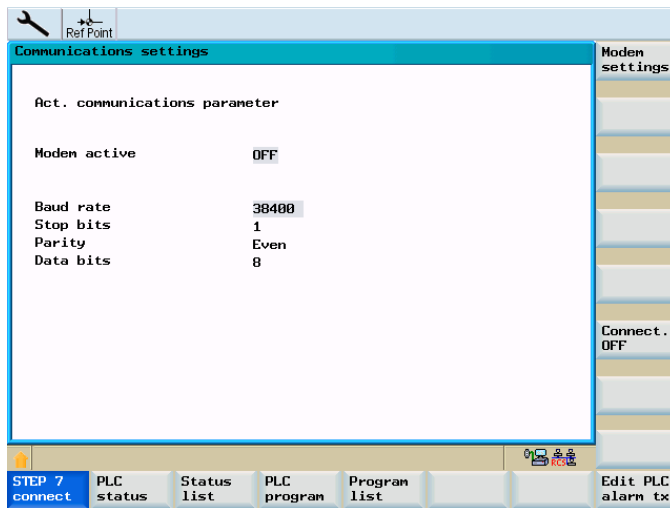


Figure 11-4 RS232 connection active

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



- You are now in the <SYSTEM> operating area.

Service display

- Press the softkeys "Service display" >"Service control".

Service control

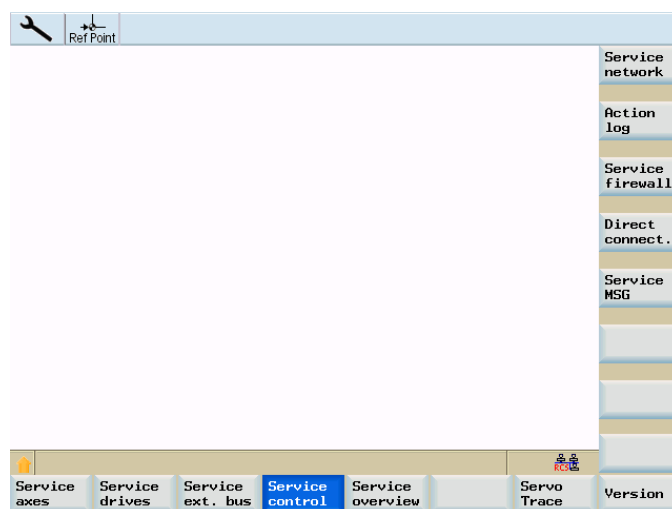


Figure 11-5 "Service control"

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

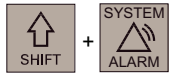
The IP address and subnet mask shown are fixed values.

These values cannot be changed.

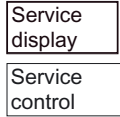
Direct connection

- 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



- You are now in the <SYSTEM> operating area.



- Press the softkeys "Service display" >"Service control".

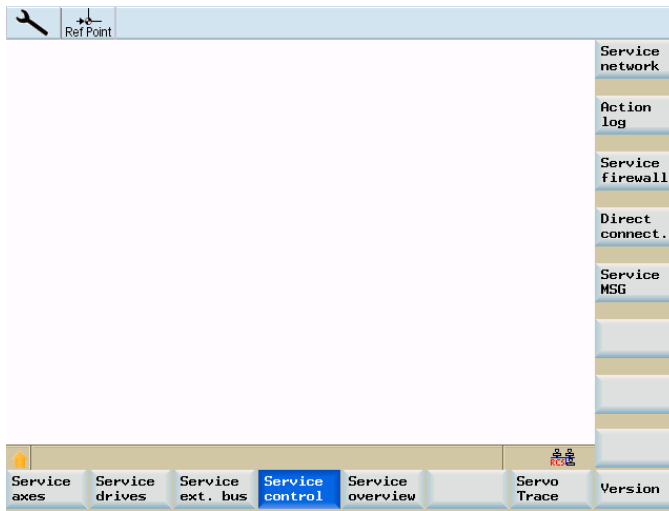
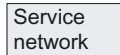


Figure 11-6 "Service control"



- 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.

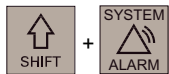
Screened network operation with encrypted data transfer is possible using an 802D specific transmission protocol. This protocol is used, e.g. for transmitting and executing part programs in conjunction with the RCS tool.

### 11.3.2 Configuring the network connection

#### Prerequisite

The control system is connected to the PC or the local network via the X5 interface.

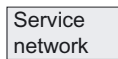
#### Entering network parameters



Switch to the the <SYSTEM> operating area.



Press the "Service display" "Service control system" softkeys.



Select the "Service network" softkey to display the network configuration window.

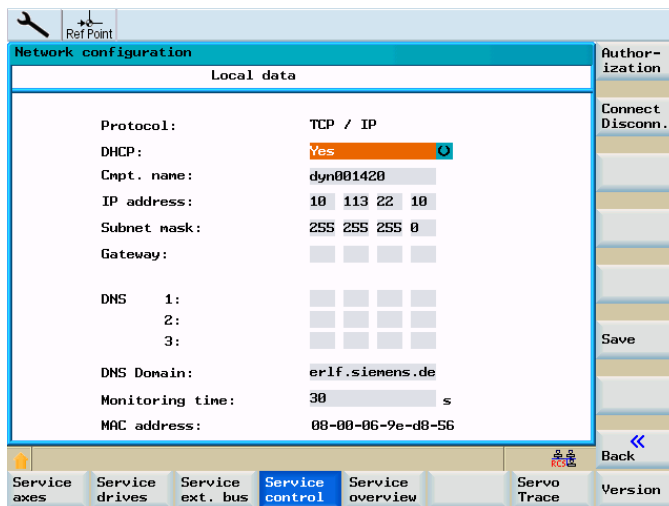


Figure 11-7 "Network configuration" start screen

Table 11- 3 Network configuration required

Parameter	Explanation
DHCP	<p>DHCP log: A DHCP server is needed in the network which dynamically distributes the IP addresses.</p> <p>When <b>No</b> is selected fixed network addresses will be assigned.</p> <p>When <b>Yes</b> is selected the network addresses are assigned dynamically. Input fields that are no longer needed will be hidden.</p> <p>If you selected "yes", the following steps are necessary to activate the fields for the computer name, IP address and Subnet mask:</p> <ol style="list-style-type: none"> <li>1. Press the vertical softkey "Save".</li> <li>2. Switch the control system off and on again.</li> </ol>
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.

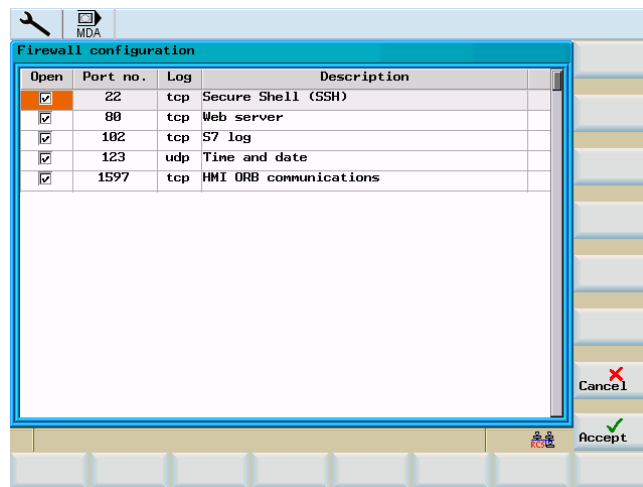


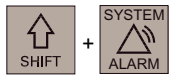
Figure 11-8 Firewall configuration

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



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.

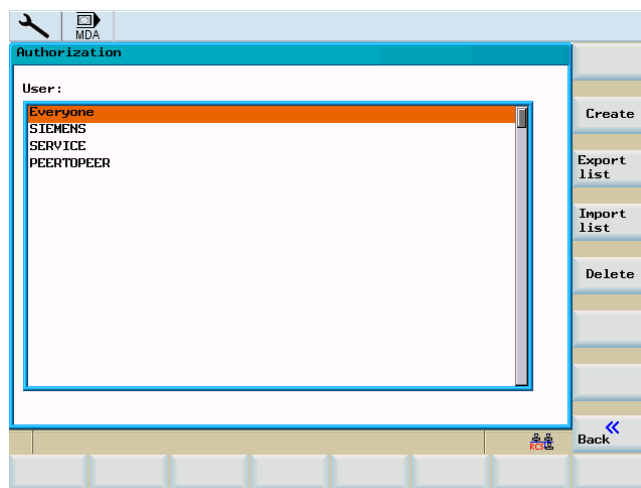


Figure 11-9 User accounts

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.

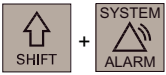
This password is required also, if the user want to communicate with the control system from the RCS tool.

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



RCS  
log-in

In the <SYSTEM> operating area, select the "RCS Connect" softkey. The user log-in input screen will appear.

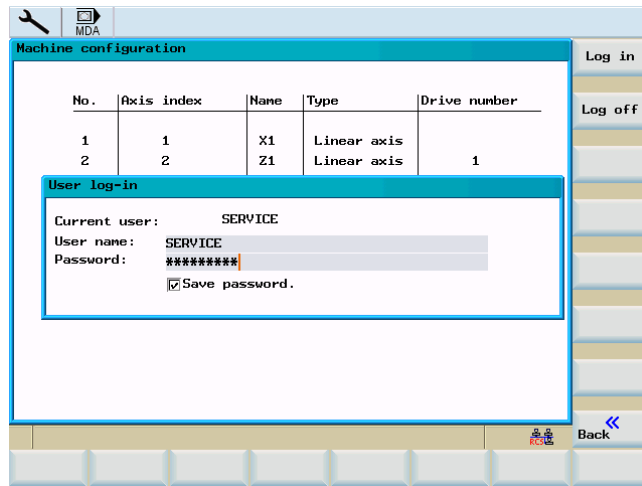


Figure 11-10 User log-in

#### 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 sharing 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.



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.

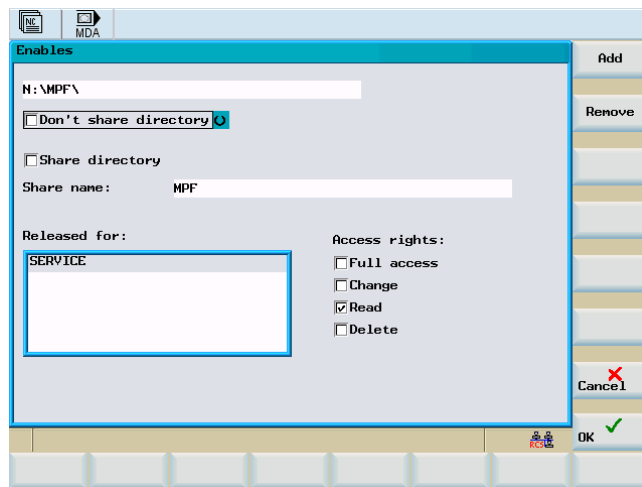
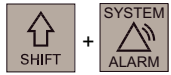


Figure 11-11 Sharing status

- 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



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.

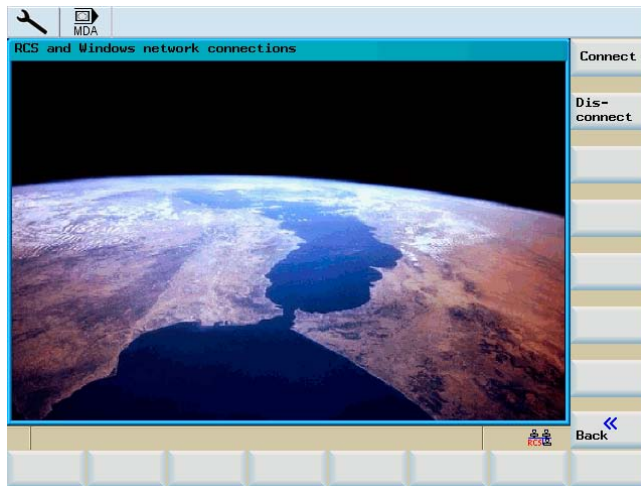


Figure 11-12 Network connections

## 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".

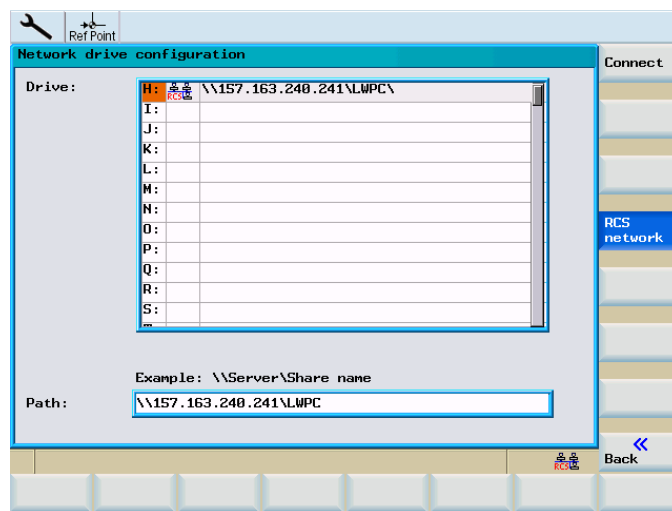


Figure 11-13 Connecting network drives

## 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

Dis-  
connect

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

### 12.1 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



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.

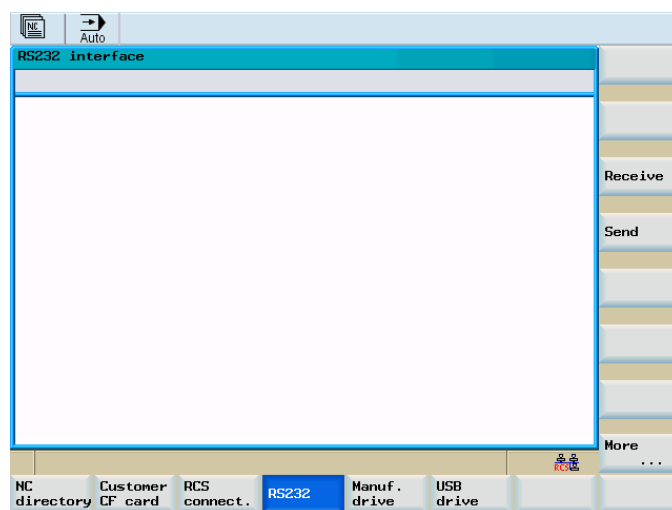


Figure 12-1 Reading out a program

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 <b>&lt;Stop&gt;</b> 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.

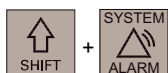


## 12.2 Creating / reading in / reading out a start-up archive

### References

SINUMERIK 802D sl Operating Instructions for Turning, Milling, Grinding, Nibbling; Data Backup and Series Start-Up

### Operating sequence



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.



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.

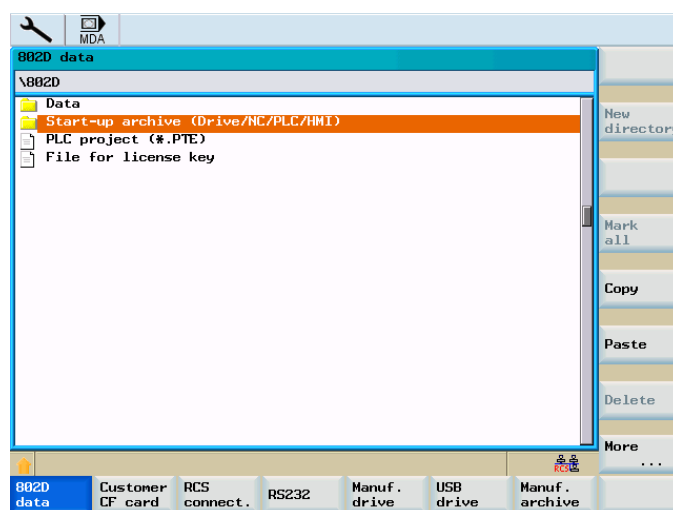


Figure 12-2 Copy entire start-up archive

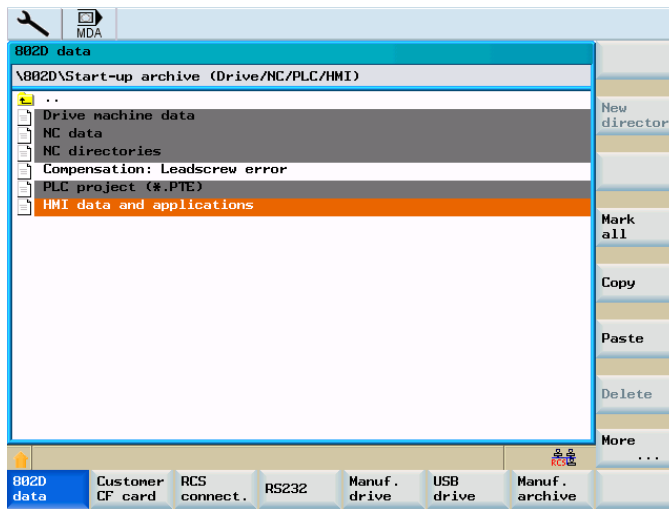


Figure 12-3 Contents of the start-up archive



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".

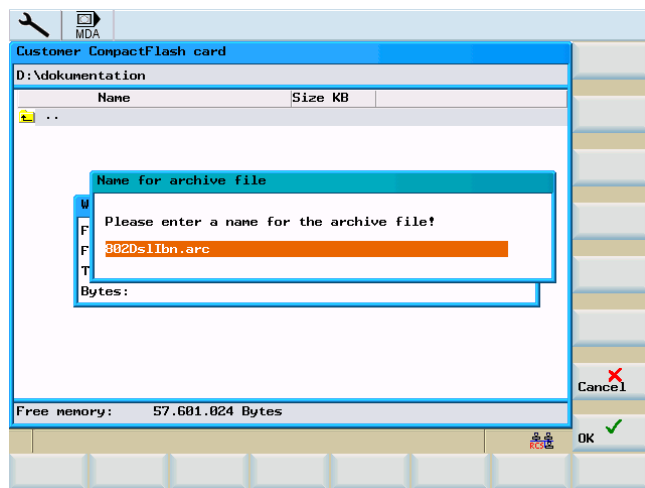


Figure 12-4 Insert files

### 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

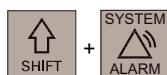
### 13.1 PLC diagnosis represented as a ladder diagram

#### 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



PLC

Press the "PLC" softkey in the <SYSTEM> operating area.

PLC  
program

Press "PLC program".

The project stored in the residual memory is opened.

## 13.2 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.

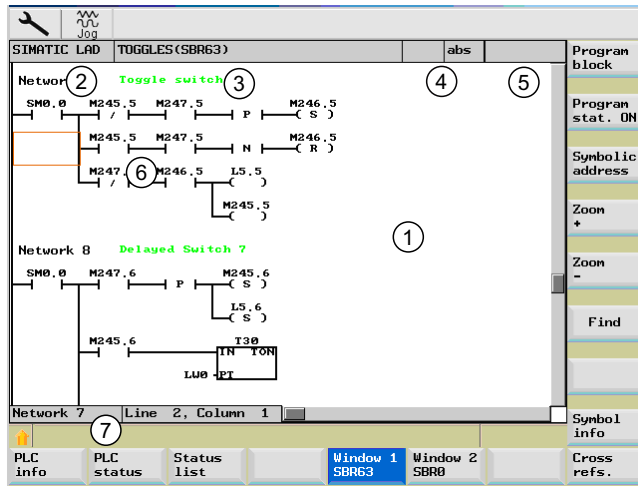


Figure 13-1 Screen layout

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
	<b>Status of the application area</b>	
	Sym	Symbolic representation
⑤	abs	Absolute representation
		<b>Display of the active keys</b>
⑥	<b>Focus</b>	Performs the tasks of the cursor
⑦	<b>Tip line</b>	contains notes for searching














## 13.3 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
 or  	To the first line of the row
 or  	To the last line of the row
	Up a screen
	Down a screen
	One field to the left
	One field to the right
	Up a field

Keystroke combination	Action
	Down a field
  or  	To the first field of the first network
  or  	To the last field of the first network
 	Opens the next program block in the same window
 and 	Opens the previous program block in the same window
	The function of the Select key depends on the position of the input focus. <ul style="list-style-type: none"> <li>• Table line: Displays the complete text line</li> <li>• Network title: Displays the network comment</li> <li>• Command: Displays the complete operands</li> </ul>
	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

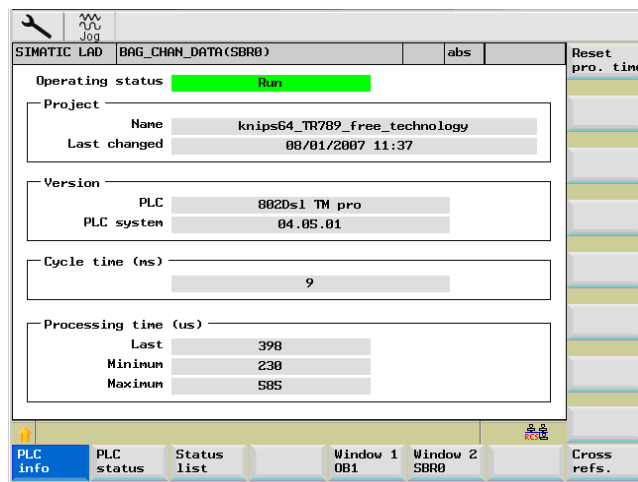


Figure 13-2 PLC info

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.

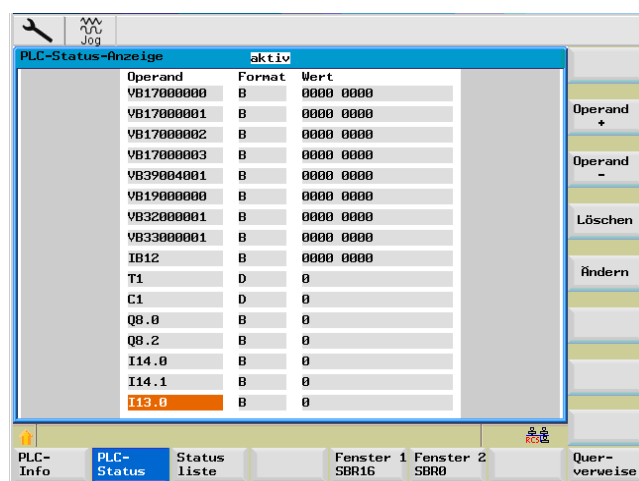


Figure 13-3 PLC status display

Status list

Use the "Status list" softkey to display and modify PLC signals.

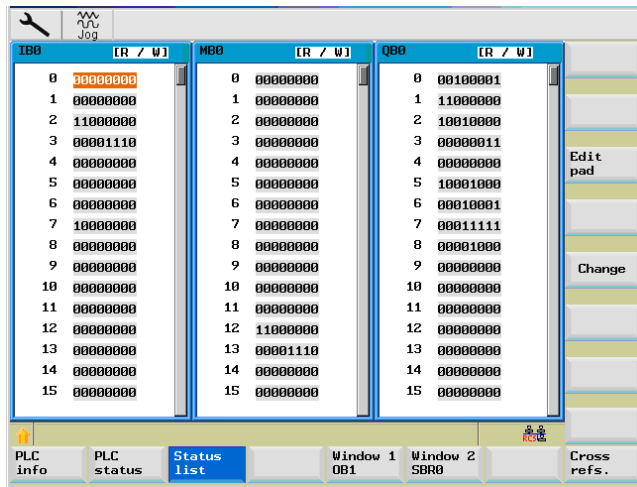


Figure 13-4 Status list

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

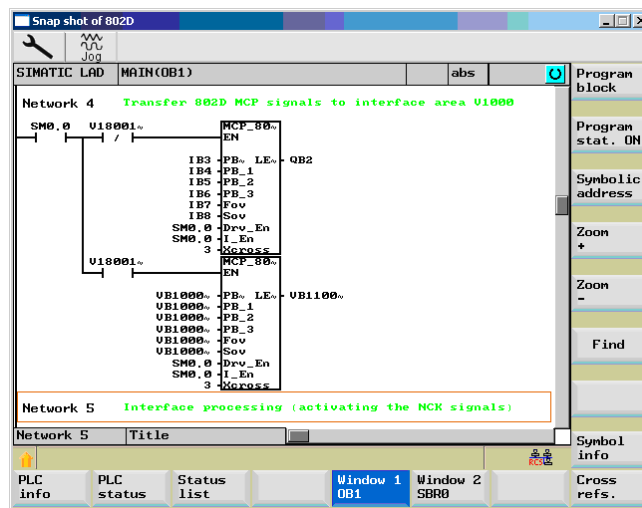


Figure 13-5 Window 1, OB1

Program block

This softkey can be used to select the list of the PLC program blocks.

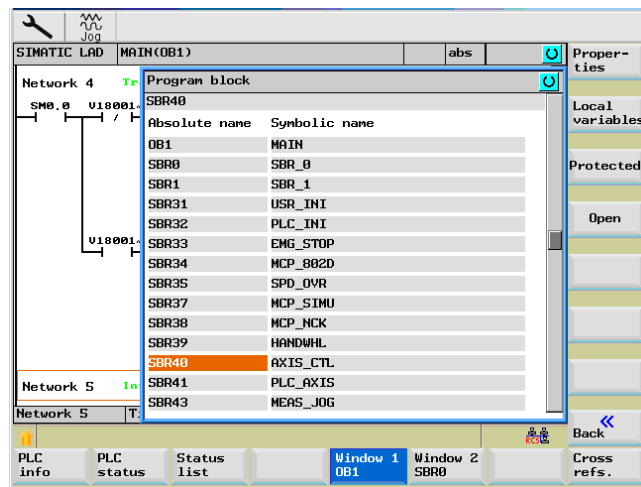


Figure 13-6 Select the PLC program block

Properties

Using this softkey will display the following properties of the selected program block:

- Symbolic name
- Author
- Comments

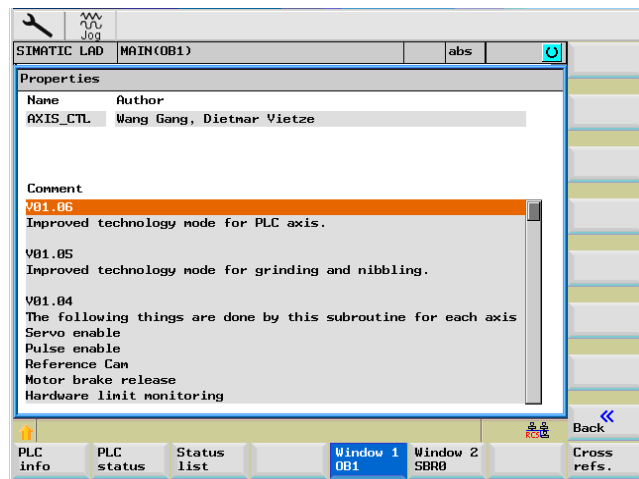


Figure 13-7 Properties of the selected PLC program block

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

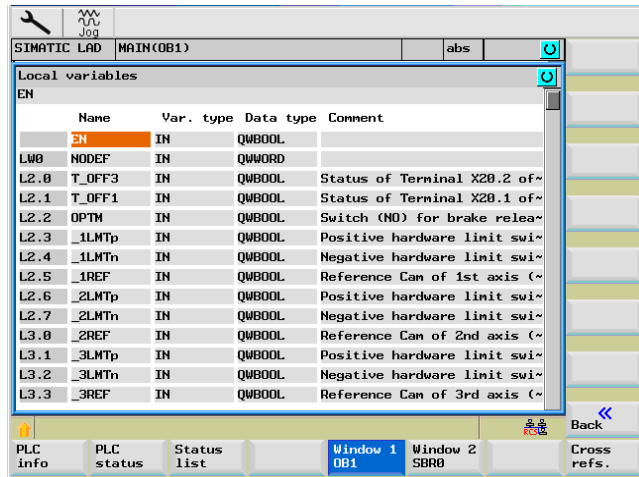


Figure 13-8 Table of local variables for the selected PLC program block

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.

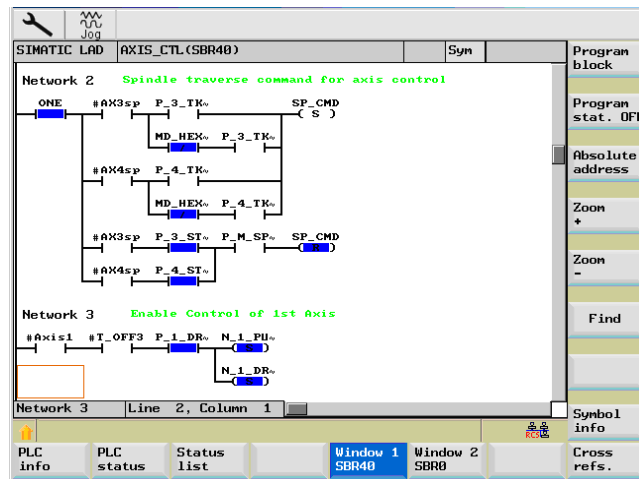


Figure 13-9 "Program status" ON – symbolic representation

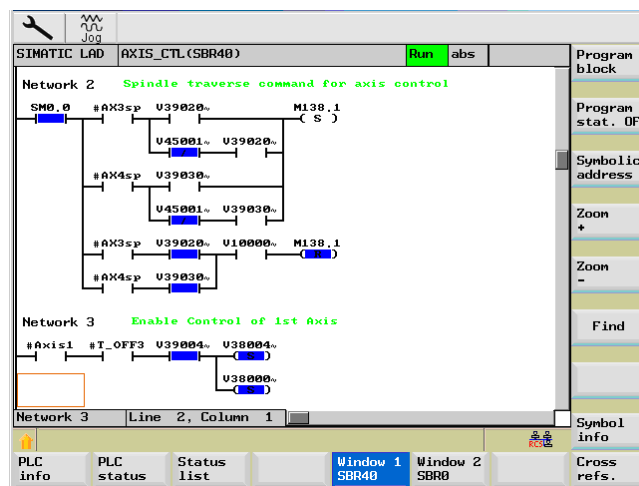


Figure 13-10 "Program status" ON – absolute representation

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  
+

The representation in the application area can be zoomed in or zoomed out step by step. The following zoom stages are provided:

Zoom  
-

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.

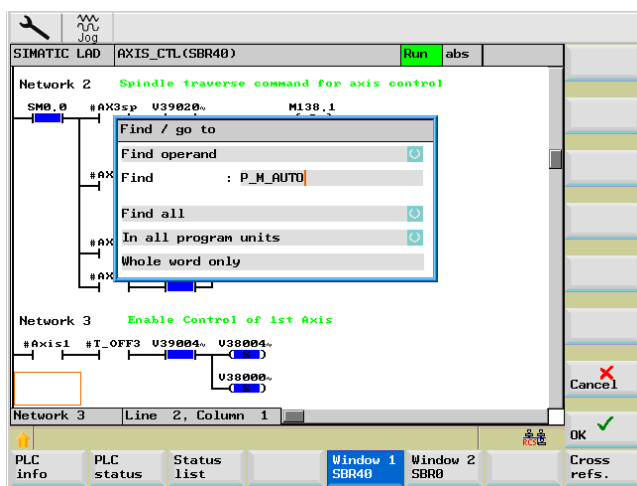


Figure 13-11 Search for symbolic operands



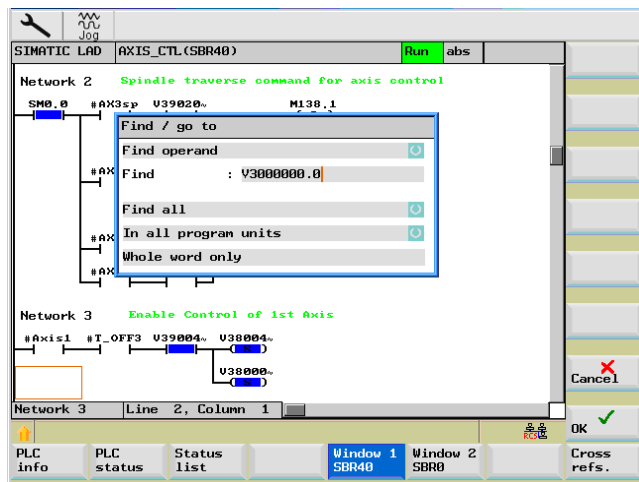


Figure 13-12 Search for absolute operands

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.

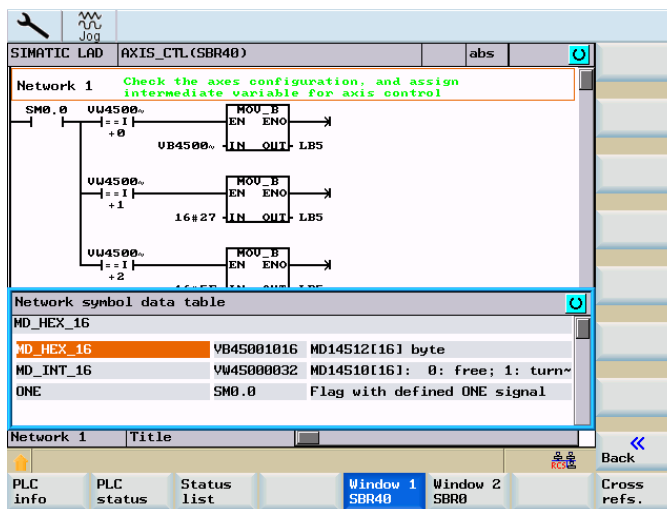


Figure 13-13 Network symbol information table

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.

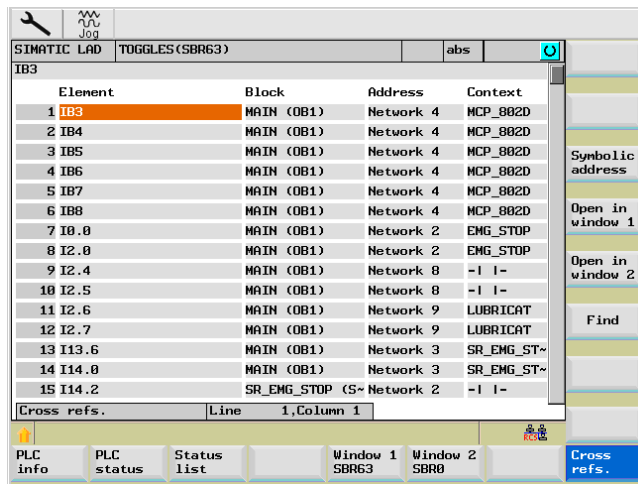


Figure 13-14 Cross references main menu (absolute)

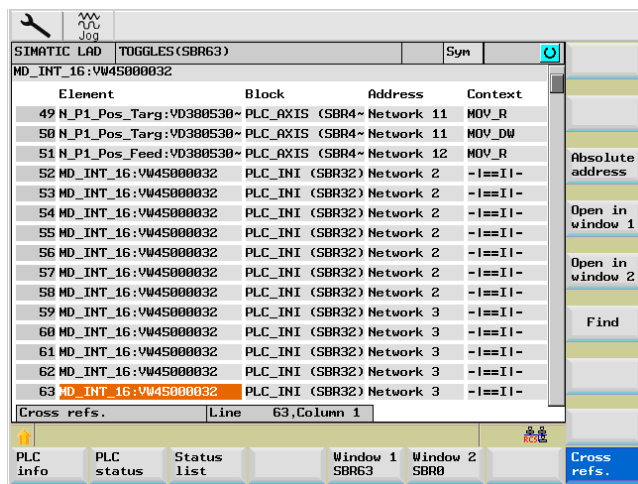


Figure 13-15 Cross references main menu (symbolic))

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.

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 18	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-

Figure 13-16 Cursor M251.0 in OB1 network 2

Open in window 1

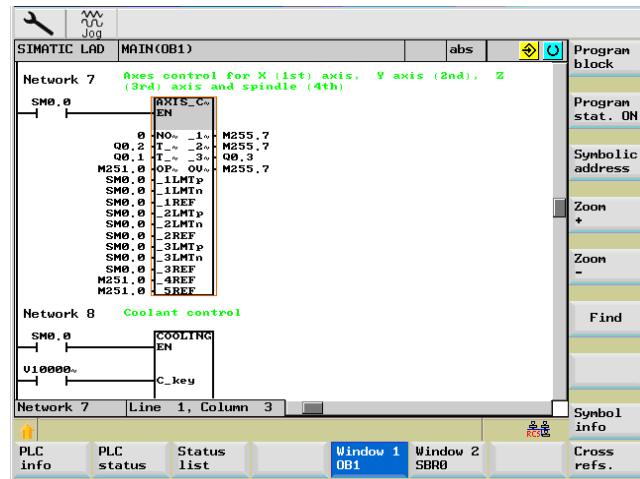


Figure 13-17 M251.0 in OB1 network 2 in window 1

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)

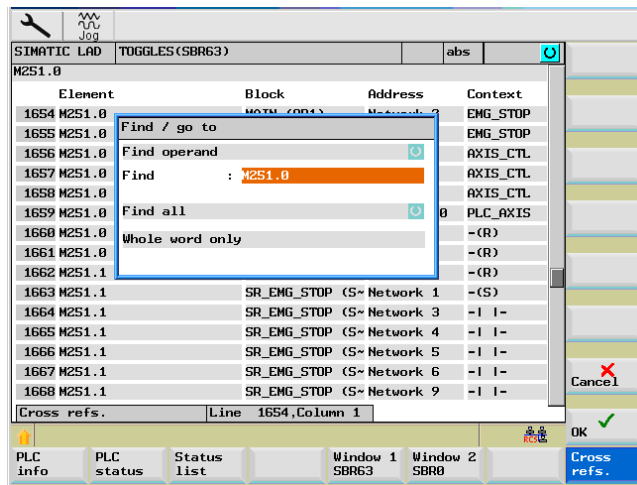


Figure 13-18 Searching for operands in cross references

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.1 Miscellaneous

#### A.1.1 Pocket calculator



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.

---

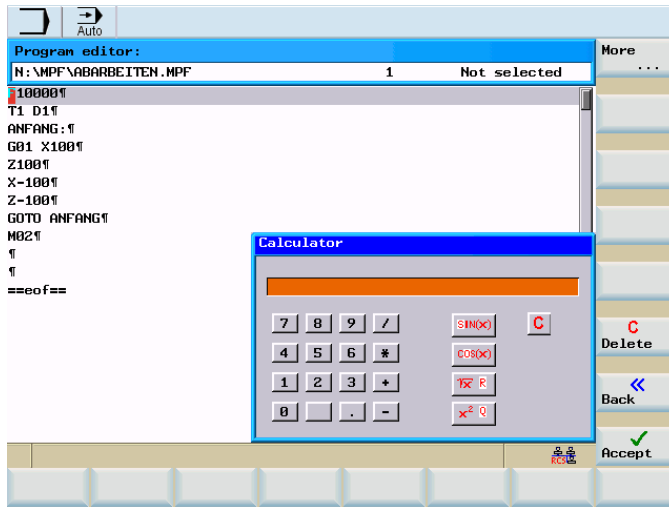


Figure A-1 Pocket calculator

**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<sup>2</sup> 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 <sup>2</sup>	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.

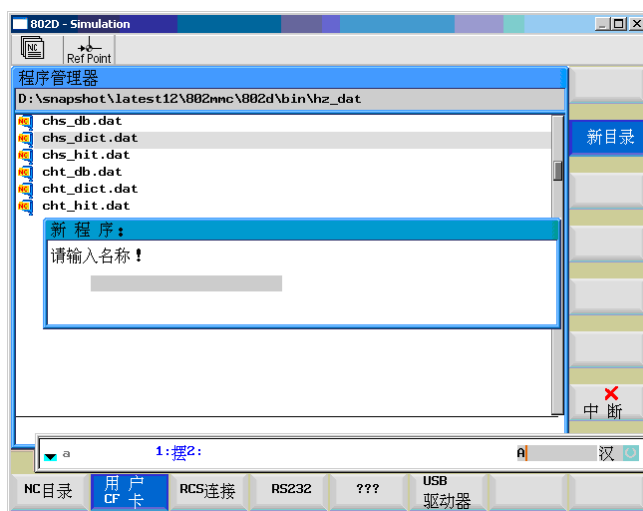


Figure A-2 Example of editing Simplified Chinese

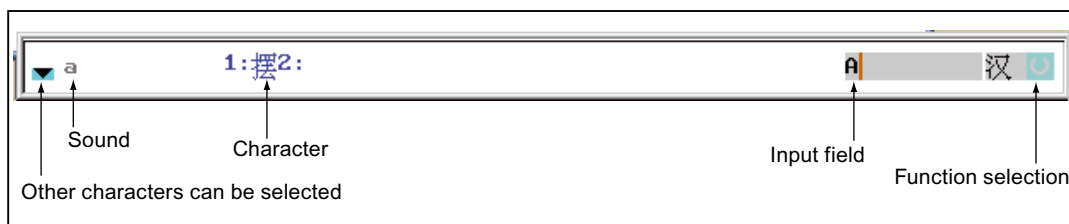


Figure A-3 Structure of editor

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.

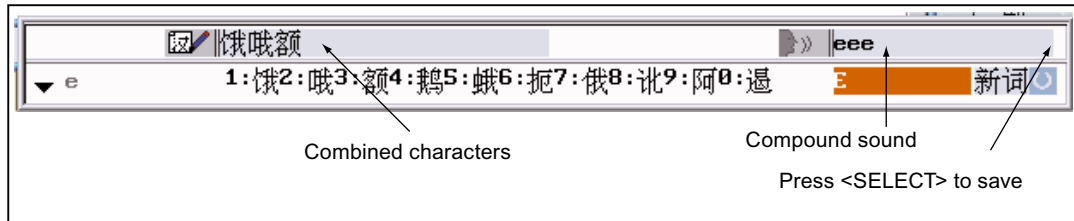


Figure A-4 Structure of editor when learning function is active

- 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.

	1	2	3	4	5	6	7	8	9	0		Backspace
Tab ↔	ㄱ	ㅋ	ㆁ	ㄷ	ㅌ	ㄴ	ㄹ	ㅍ	ㅊ	ㅌ	ㅍ	Enter ↵
Caps Lock	ㄱ	ㅋ	ㆁ	ㄷ	ㅌ	ㄴ	ㄹ	ㅍ	ㅊ	ㅌ	ㅍ	
↑	ㅈ	ㅊ	ㅌ	ㅍ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	↑
Ctrl		Alt										Ctrl

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.



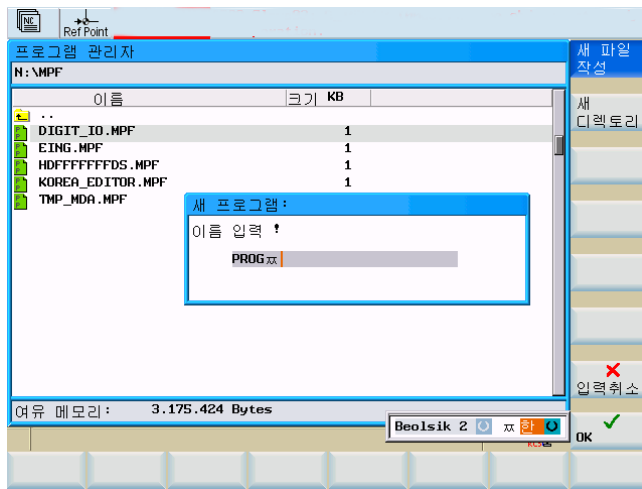


Figure A-6 Korean editor with standard keyboard assignment

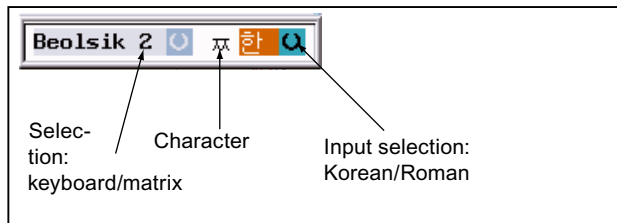


Figure A-7 Structure of Korean editor

- 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.

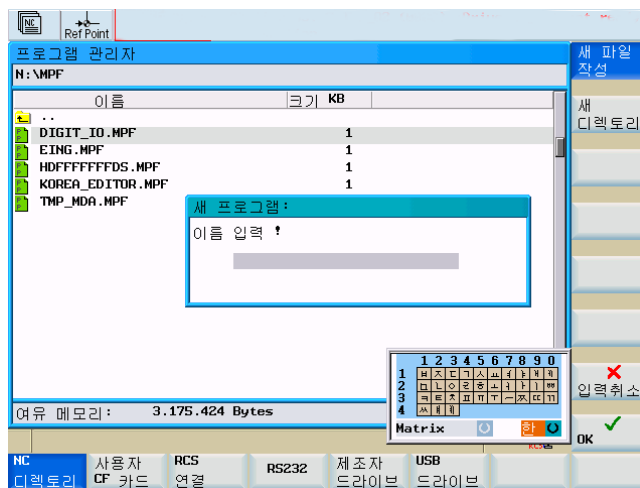


Figure A-8 Korean editor with selection matrix

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>

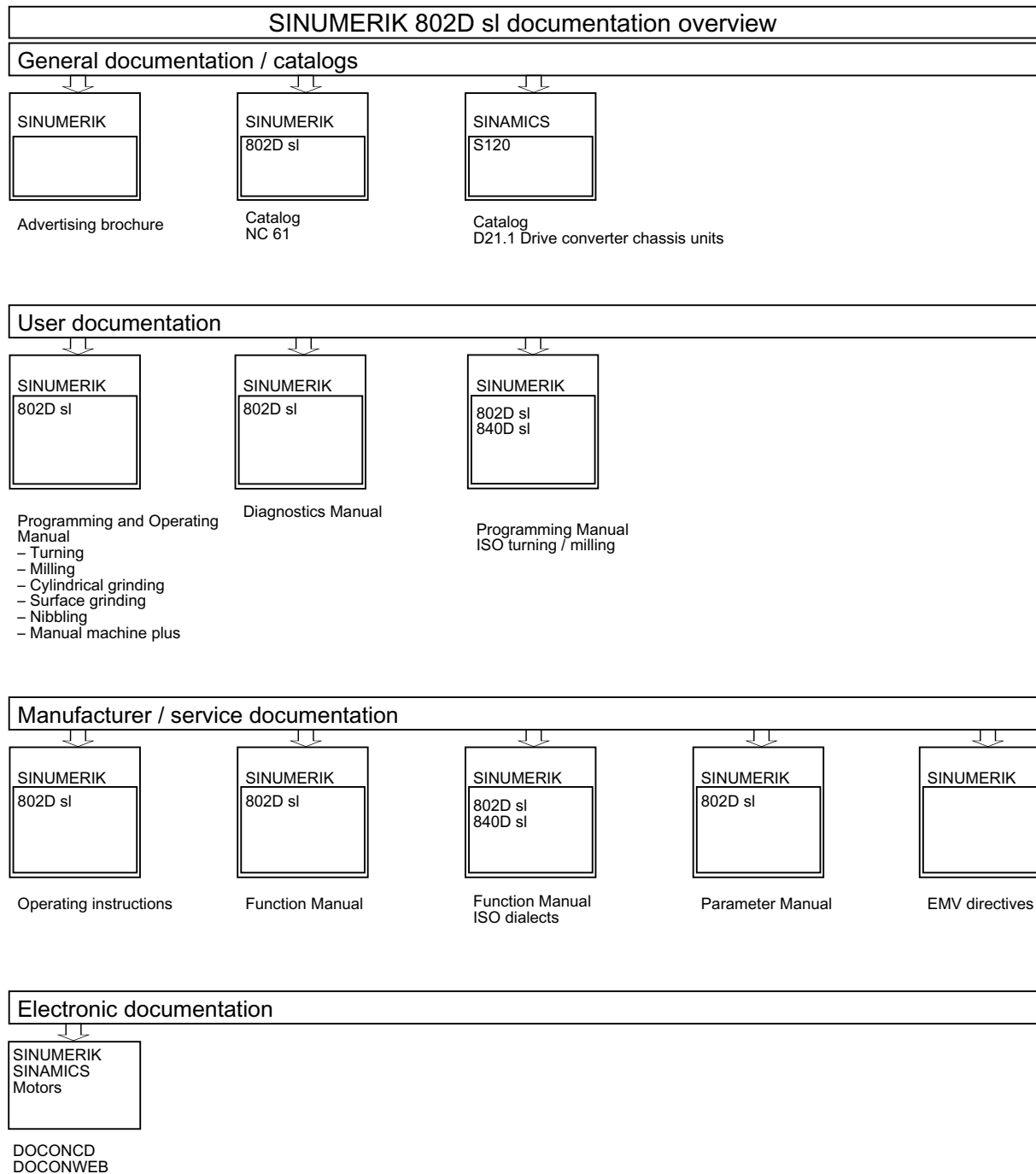
Fax: +49 9131 - 98 2176

Please use the fax form on the back of this page.

<b>To</b> SIEMENS AG I DT MC MS1 P.O. Box 3180  D-91050 Erlangen / Germany  Fax: +49 9131 - 98 2176 (Documentation)	<b>From</b>
	Name:
	Address of your company/department
	Street:
	Zip code:            City:
	Phone:                /
Fax:                    /	

Suggestions and/or corrections

## A.3 Overview of documentation





# Index

"

"Display areas", 89  
"Program" operating area, 121

## A

Absolute drilling depth, 358, 407, 436, 441  
Absolute/incremental, 135  
Access right, 29  
Address, 192  
Arithmetic parameters, 62  
Axis  
    Coupled-motion, 271  
Axis assignment, 349  
Axis-specific machine data, 153

## B

Behavior when quantity parameter is zero, 394  
Block format, 193  
Block search, 86  
Boring cycle, 354

## C

Call, 355  
Call conditions, 349  
Cartesian/polar, 135  
Centering, 357  
Chamfer, 127  
Change language, 148  
Channel-specific machine data, 154  
Circle of holes, 399  
Circular spigot milling - CYCLE77, 428  
Circumferential slot - SLOT2, 446  
Configuring the input screens, 353  
Connecting network drives, 488  
Contour allowance, 128, 133  
Contour elements, 120, 130  
Contour simulation, 107  
Contour transition element, 127  
Coordinate systems, 19  
    Machine coordinate system (MCS), 20  
    Relative coordinate system, 21

    Workpiece coordinate system (WCS), 21  
Corner deceleration at all corners, 267  
Corner deceleration at inside corners, 267  
Coupled motion, 268  
    Dynamics limit, 271  
Coupled-axis combinations, 268  
Coupling factor, 268  
Coupling status, 271  
Cycle alarms, 472  
Cycle call, 350  
Cycle support in the program editor, 352  
CYCLE71, 404  
CYCLE72, 410  
CYCLE77, 428  
CYCLE81, 357  
CYCLE82, 360  
CYCLE83, 363  
CYCLE84, 368  
CYCLE840, 372  
CYCLE85, 379  
CYCLE86, 382  
CYCLE87, 386  
CYCLE88, 389  
CYCLE89, 391  
CYCLE90, 465

## D

Data transfer, 491  
Deep-hole drilling, 363  
Deep-hole drilling with chip breaking, 364  
Deep-hole drilling with swarf removal, 364  
Determining the tool offsets  
    Manual, 42  
Disconnecting network drives, 488  
Display of machine data, 156  
Drilling, 354, 357  
Drilling 1, 379  
Drilling 2, 382  
Drilling 3, 386  
Drilling 4, 389  
Drilling 5, 391  
Drilling cycles, 347  
Drilling pattern cycles, 348, 394  
Drilling pattern cycles without drilling cycle call, 394  
Drilling, counterboring, 360  
Drive machine data, 155

- E**
  - Enabling the communication ports, 483
  - Entering tools and tool offsets, 36
  - Error displays, 14
  - Execution from external, 94
  - EXTCALL, 316, 317
  - External thread, 466
  
- F**
  - Face milling, 404
  - FENDNORM, 267
  - Files
    - Copy, 497
    - Paste, 497
  - Fonts, 195
  - Free contour programming, 118
  - Fundamental Principles of NC Programming, 191
  
- G**
  - G62, 199, 267
  - G621, 199, 267
  - General machine data, 152
  - Geometrical parameters, 354
  - Geometry processor, 118
  
- H**
  - Hand wheel, 69
  - Help mode, 126
  - Help system, 30
  - HOLES1, 395
  - HOLES2, 399
  - Hot keys, 16
  
- I**
  - Interface parameters, 186
  - Internal thread, 467
  
- J**
  - JOG, 65
  - JOG mode, 65
  
- L**
  - LED displays on the CNC operator panel (PCU), 14
  
- Long holes located on a circle - LONGHOLE, 433
- LONGHOLE, 433
  
- M**
  - Machine data, 151
    - Axis-specific machine data, 153
    - Channel-specific machine data, 154
    - Display of machine data, 156
    - Drive machine data, 155
    - General machine data, 152
  - Machine operating area, 65
  - Machine zero, 54
  - Machining parameters, 354
  - Manual input, 70
  - Manufacturer archive, 183
  - MASLDEF, 272
  - MASLDEL, 272
  - MASLOF, 272
  - MASLOFS, 272
  - MASLON, 272
  - MDA mode, 70
  - Messages, 474
  - Milling a circular pocket - POCKET4, 460
  - Milling a rectangular pocket - POCKET3, 451
  - Milling cycles, 348
  - Modem, 176
  - Monitoring counter, 324
  
- N**
  - Network connection, 482
  - Network operation, 481
  - Network parameters, 482
  - Non-printable special characters, 195
  
- O**
  - Online help, 30
  - Operating areas, 28
  - Operating plane, 349
  - Operating the cycle support, 352
  - Operator control and display elements, 13
  - Overview of cycle alarms, 472
  - Overview of cycle files, 352
  
- P**
  - Parameters operating area, 36
  - Part program, 121
    - selection and start, 84



Stopping / canceling, 91  
Path milling, 410  
Plane definition, 349  
Plausibility checks, 394  
POCKET3, 451  
POCKET4, 460  
Polar coordinates, 134  
Pole, 120, 134  
Pole change, 136  
Printable special characters, 195  
Program list, 181  
Program Manager, 97  
Protection levels, 29

## R

Radius, 127  
RCS log in, 485  
RCS tool, 476  
Reapproach after cancellation, 92  
Reapproaching after interruption, 93  
Recompile, 122  
Reference plane, 357  
Reference point approach, 33  
Relative drilling depth, 358, 407, 436, 441  
Retraction plane, 357  
Rigid tapping, 368  
Row of holes, 395  
RS232 interface, 491

## S

Safety distance, 357  
Saving data, 149  
Screen layout, 23  
Service life, 324  
SETPIECE, 329  
Setting data, 58  
Sharing directories, 487  
Simulation of cycles, 351

Simultaneous recording, 83  
SLOT1, 439  
SLOT2, 446  
Slots on a circle - SLOT1, 439  
SPOS, 369, 370  
Standard simulation, 106  
Starting point, 123, 129  
Status displays, 14

## T

Tangent to preceding element, 130  
Tapping with compensating chuck, 372  
Tapping with compensating chuck with encoder, 374  
Tapping with compensating chuck without encoder, 373  
Thread milling, 465  
Tool list, 36  
Tool monitoring, 323  
Tool radius compensation  
    Corner deceleration, 267  
Tool zero, 54  
TRAILOF, 268  
TRAILON, 268  
Transmission messages, 492  
Transmission protocol, 492

## U

User log-in, 485  
User management, 484

## W

Word structure, 192  
Work offset, 54  
Workpiece count, 324