

Operating and Programming 01/2002 Edition

sinumerik

Milling
SINUMERIK 802S/802C

SIEMENS

SIEMENS

SINUMERIK 802S SINUMERIK 802C

Operation and Programming Milling

Valid for

<i>Control System</i>	<i>Software Version</i>
SINUMERIK 802S	3
SINUMERIK 802C	3

01.2002 Edition

Introduction	1
Power ON	2
Setup	3
Manually Controlled Operation	4
Automatic Mode	5
Part Programming	6
Services and Diagnosis	7
Programming	8
Cycles	9

SINUMERIK® Documentation

Printing history

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

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

Status code in the "Remarks" column:

A New documentation.

B Unrevised reprint with new Order No.

C Revised edition with new status.

If actual changes have been made on the page since the last edition, this is indicated by a new edition coding in the header on the page.

Edition	Order-No.	Remark
02.99	6FC5598-2AA10-0BP1	A
04.00	6FC5598-3AA10-0BP1	C
01.02	6FC5598-3AA10-0BP2	C

This Manual is included on the documentation on CD-ROM (**DOCONCD**)

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

Trademarks

SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIROTEC®, SINUMERIK® and SIMODRIVE® are registered trademarks of Siemens. Third parties using for their own purposes any other names in this document which refer to trademarks might infringe upon the rights of trademark owners.

This publication was produced with Interleaf V 7

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of utility model or design, are reserved.

© Siemens AG 2002. All rights reserved.

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

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

Subject to change without prior notice.

Safety notices

This Manual contains notices intended to ensure your personal safety and to avoid material damage. The notices are highlighted by a warning triangle and, depending on the degree of hazard, represented as shown below:



Danger

indicates that loss of life, severe personal injury or substantial material damage **will** result if the appropriate precautions are not taken.



Warning

indicates that loss of life, severe personal injury or substantial material damage **may** result if the appropriate precautions are not taken.



Caution

indicates that minor personal injury or material damage may result if the appropriate precautions are not taken.

Caution

without a warning triangle means that a material damage **can** occur if the appropriate precautions are not taken.

Attention

means that an undesired event or status **can** occur if the appropriate note is not observed.

Note

is used to draw your special attention to an important information on the product, the handling of the product or the corresponding part of the documentation.

Qualified personnel

Start-up and operation of a device may only be carried out by **qualified personnel**. Qualified personnel as referred to in the safety notices provided in this Manual are persons who are authorized to start up, ground and tag devices, systems and circuits according to the relevant safety standards.

Usage as per intended purpose

Please observe the following:



Warning

The device may only be used for the cases of application, as intended by the Catalog, and only in conjunction with third-party devices and components recommended or approved by Siemens.

The proper and safe operation of the product requires transport, storage and installation according to the relevant instructions and qualified operation and maintenance at the prescribed intervals.

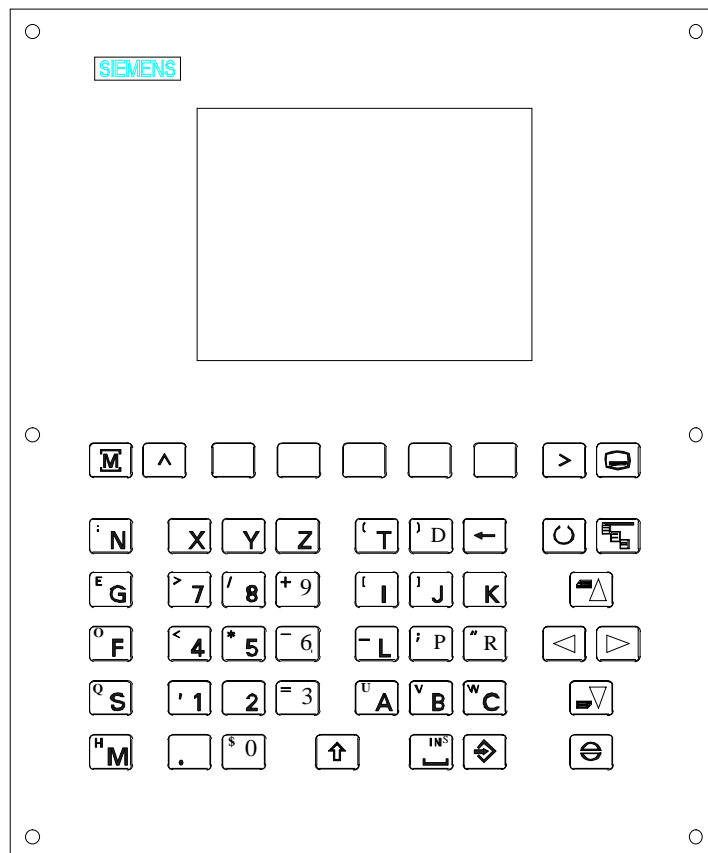
Table of Contents





















1	Introduction	1-11
1.1	Screen Layout	1-11
1.2	Operating areas	1-14
1.3	Overview of the most important softkey functions	1-15
1.4	Pocket calculator	1-16
1.5	Basic principles	1-22
2	Turning On and Reference Point Approach	2-25
3	Setup	3-27
3.1	Entering tools and tool offsets	3-27
3.2	Creating a new tool	3-28
3.3	Tool compensation data	3-29
3.4	Determining the tool offsets	3-30
3.5	Entering/modifying zero offsets	3-32
3.6	Determining the zero offset	3-34
3.7	Programming the setting data – “Parameters” operating area	3-36
3.8	R parameters - “Parameters” operating area	3-38
4	Manually Operated Mode	4-39
4.1	Jog mode - “Machine” operating area	4-39
4.1.1	Assigning handwheels	4-41
4.2	MDA Mode (Manual Data Input) - “Machine” operating area	4-43
5	Automatic Mode	5-45
5.1	Selecting/starting a part program - “Machine” operating area	5-48
5.2	Block search - “Machine” operating area	5-49
5.3	Stopping/cancelling a part program - “Machine” operating area	5-49
5.4	Repositioning after interruption	5-50
5.5	Program execution from external (V.24 interface)	5-51
5.6	Teach In	5-52
6	Part Programming	6-55
6.1	Entering a new program - “Program” operating area	6-57
6.2	Editing part programs - “Program” operating area	6-58
6.3	Programming support	6-61
6.3.1	Vertical menu	6-61
6.3.2	Cycles	6-62
6.3.3	Contour	6-63
6.3.4	Free softkey assignment	6-79
7	Services and Diagnosis	7-81
7.1	Data transfer via the V24 Interface	7-81
7.1.1	Interface parameters	7-84
7.1.2	Special functions	7-85
7.1.3	Interface parameterization	7-85

7.2	Diagnosis and start-up - "Diagnostics" operating area	7-87
8	Programming	8-99
8.1	Fundamentals of NC programming	8-99
8.1.1	Program structure	8-99
8.1.2	Word structure and address	8-100
8.1.3	Block structure	8-100
8.1.4	Character set	8-102
8.1.5	Overview of instructions	8-104
8.2	Position data	8-114
8.2.1	Plane selection: G17 to G19	8-114
8.2.2	Absolute/incremental dimensions: G90, G91	8-115
8.2.3	Metric/inch dimensions: G71, G70	8-116
8.2.4	Programmable zero offset and rotation: G158, G258, G259	8-117
8.2.5	Workpiece clamping – settable zero offset: G54 to G57, G500, G53	8-119
8.3	Axis movements	8-121
8.3.1	Linear interpolation at rapid traverse: G0	8-121
8.3.2	Linear interpolation at feedrate: G1	8-122
8.3.3	Circular interpolation: G2, G3	8-123
8.3.4	Circular interpolation via intermediate point: G5	8-126
8.3.5	Thread cutting with constant lead: G33	8-127
8.3.6	Tapping with compensating chuck: G63	8-128
8.3.7	Thread interpolation: G331, G332	8-129
8.3.8	Fixed-point approach: G75	8-131
8.3.9	Reference point approach: G74	8-131
8.3.10	Feedrate F	8-131
8.3.11	Feed override for circles: G900, G901	8-132
8.3.12	Exact stop / continuous-path operation: G9, G60, G64	8-133
8.3.13	Dwell time: G4	8-136
8.4	Spindle movements	8-137
8.4.1	Spindle speed S, directions of rotation	8-137
8.4.2	Spindle speed limitation: G25, G26	8-137
8.4.3	Spindle positioning: SPOS	8-138
8.5	Rounding, chamfer	8-139
8.6	Tool and tool offset	8-141
8.6.1	General notes	8-141
8.6.2	Tool T	8-142
8.6.3	Tool offset number D	8-142
8.6.4	Selection of tool radius offset: G41, G42	8-146
8.6.5	Behavior at corners: G450, G451	8-148
8.6.6	Tool radius compensation OFF: G40	8-149
8.6.7	Special cases of tool radius compensation	8-151
8.6.8	Example of tool radius compensation	8-153
8.7	Miscellaneous function M	8-154
8.8	Arithmetic parameters R	8-155
8.9	Program branches	8-157
8.9.1	Labels – destination for program branches	8-157
8.9.2	Unconditional program branches	8-157
8.9.3	Conditional branches	8-158
8.9.4	Example of program with branches	8-160
8.10	Subroutine technique	8-161
9	Cycles	9-165
9.1	General information about standard cycles	9-165
9.1.1	Overview of cycles	9-165

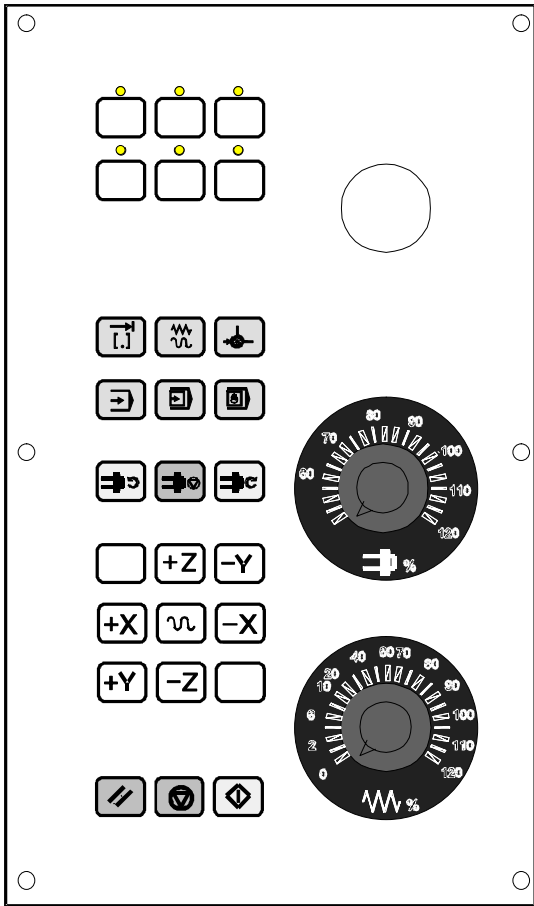
9.1.2	Error messages and error handlingin cycles	9-166
9.2	Drilling cycles	9-168
9.2.1	Drilling, spot facing – LCYC82	9-168
9.2.2	Deep hole drilling – LCYC83	9-170
9.2.3	Tapping without compensating chuck – LCYC84	9-174
9.2.4	Tapping with compensating chuck – LCYC840	9-176
9.2.5	Boring – LCYC85	9-179
9.3	Drilling patterns	9-181
9.3.1	Drilling a row of holes – LCYC60	9-181
9.3.2	Hole circle – LCYC61	9-185
9.4	Milling cycles	9-187
9.4.1	Cutting square pockets, slots and circular pockets – LCYC75	9-187

SINUMERIK 802S Slimline Operator Panel OP



	Softkey		Vertical menu
	Machine area key		Acknowledge alarm
	Recall key		Selection key/toggle key
	ETC key		ENTER / input key
	Area switchover key		Shift key
	Cursor UP with shift: page up		Cursor DOWN with shift: page down
	Cursor LEFT		Cursor RIGHT
	Delete key (backspace)		SPACE (INSERT)
 	Numerical keys shift for alternative assignment	 	Alphanumeric keys shift for alternative assignment

External Machine Control Panel



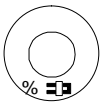
RESET



NC STOP



NC START



Spindle Speed Override
(optional)



User-defined key with LED



User-defined key without LED



INCREMENT



JOG



REFERENCE POINT



AUTOMATIC



SINGLE BLOCK



MANUAL DATA



SPINDLE START LEFT
Counterclockwise direction



SPINDLE START RIGHT
Clockwise direction



SPINDLE STOP



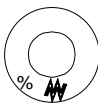
RAPID TRAVERSE OVERLAY



X axis



Z axis



Feed Rate Override

Introduction

1.1 Screen Layout

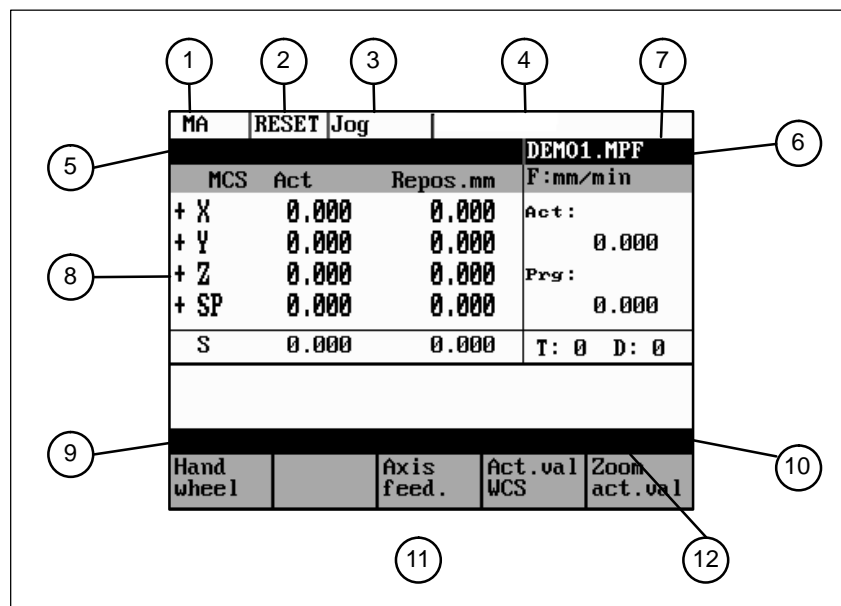


Fig. 1-1 Screen layout

The abbreviations on the screen stand for the following:




Table 1-1 Explanation of display elements

Display Element	Abbreviation	Meaning
① Active operating area	MA	Machine
	PA	Parameter
	PR	Programming
	DI	Services
	DG	Diagnosis
② Program status	STOP	Programm stopped
	RUN	Program running
	RESET	Program aborted

Table 1-1 Explanation of display elements

Display Element	Abbreviation	Meaning
③ Operating mode	Jog	Manual traverse
	MDA	Manual input with automatic function
	Auto	Automatic
④ Status display	SKP	Skip block Program blocks marked by a slash in front of the block number are ignored during program execution.
	DRY	Dry run feed Traversing movements are executed at the feed specified in the Dry Run Feed setting data.
	ROV	Rapid traverse override The feed override also applies to rapid feed mode.
	SBL	Single block with stop after each block When this function is active, the part program blocks are processed separately in the following manner: Each block is decoded separately, the program is stopped at the end of each block. The only exception are thread blocks without dry run feed. In this case, the program is stopped only when the end of the current thread block is reached. SBL can only be selected in the RESET state.
	M1	Programmed stop When this function is active, the program is stopped at each block in which the miscellaneous function M01 is programmed. In this case, the message "5 stop M00/M01 active" appears on the screen.
	PRT	Program test
	1...1000 INC	Incremental mode If the control is in the Jog mode, incremental dimension is displayed instead of the active program control function.

Table 1-1 Explanation of display elements

Display Element	Abbreviation	Meaning
⑤ Operational messages	1	Stop: No NC Ready
	2	
	3	Stop: EMERGENCY STOP active
	4	Stop: Alarm active with stop
	5	Stop: M0/M01 active
	6	Stop: Block ended in SBL mode
	7	Stop: NC STOP active
	8	Wait: Read-in enable missing
	9	Wait: Feed enable missing
	10	Wait: Dwell time active
	11	Wait: Auxiliary function acknowl. missing
	12	Wait: Axis enable missing
	13	Wait: Exact stop not reached
	14	
	15	Wait: For spindle
	16	
	17	Wait: Feed override to 0%
	18	Stop: NC block incorrect
	19	
	20	
	21	Wait: Block search active
	22	Wait: No spindle enable
	23	Wait: Axis feed value 0
⑥ Program name		
⑦ Alarm line		The alarm line is only displayed if an NC or PLC alarm is active. The alarm line contains the alarm number and reset criterion of the most recent alarm.
⑧ Working window		Working window and NC display
⑨ Recall symbol		This symbol is displayed above the softkey bar when the operator is in a lower-level menu. When the Recall key is pressed, you can return to the next-higher menu without saving data.
⑩ Menu extension		ETC is possible If this symbol appears above the softkey bar, further menu functions are provided. These functions can be activated by the ETC key.
⑪ Softkey bar		
⑫ Vertical menu		If this symbol is displayed above the softkey bar, further menu functions are provided. When the VM key is pressed, these functions appear on the screen and can be selected by Cursor UP and Cursor DOWN.

1.2 Operating areas

The basic functions are grouped in the CNC into the following operating areas:

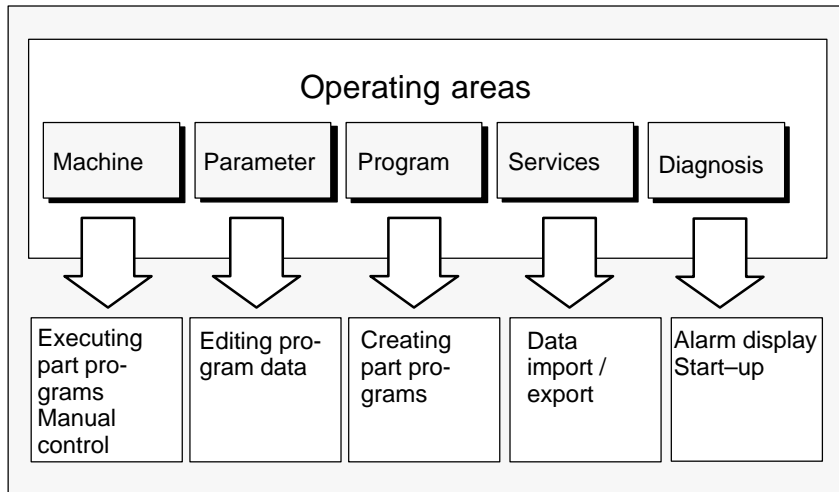


Fig. 1-2 SINUMERIK 802S operating areas

Switching between the operating areas



Press the "Machine area" key for direct access to the "Machine" operating area.



Use the area switching key to return from any operating area to the main menu.

Press the area switching key twice to return to the previous operating area.

After turning on the control system, you will always find yourself in the **Machine** operating area.

Protection levels

Sensible points of the control system are password-protected against entering and modifying data.

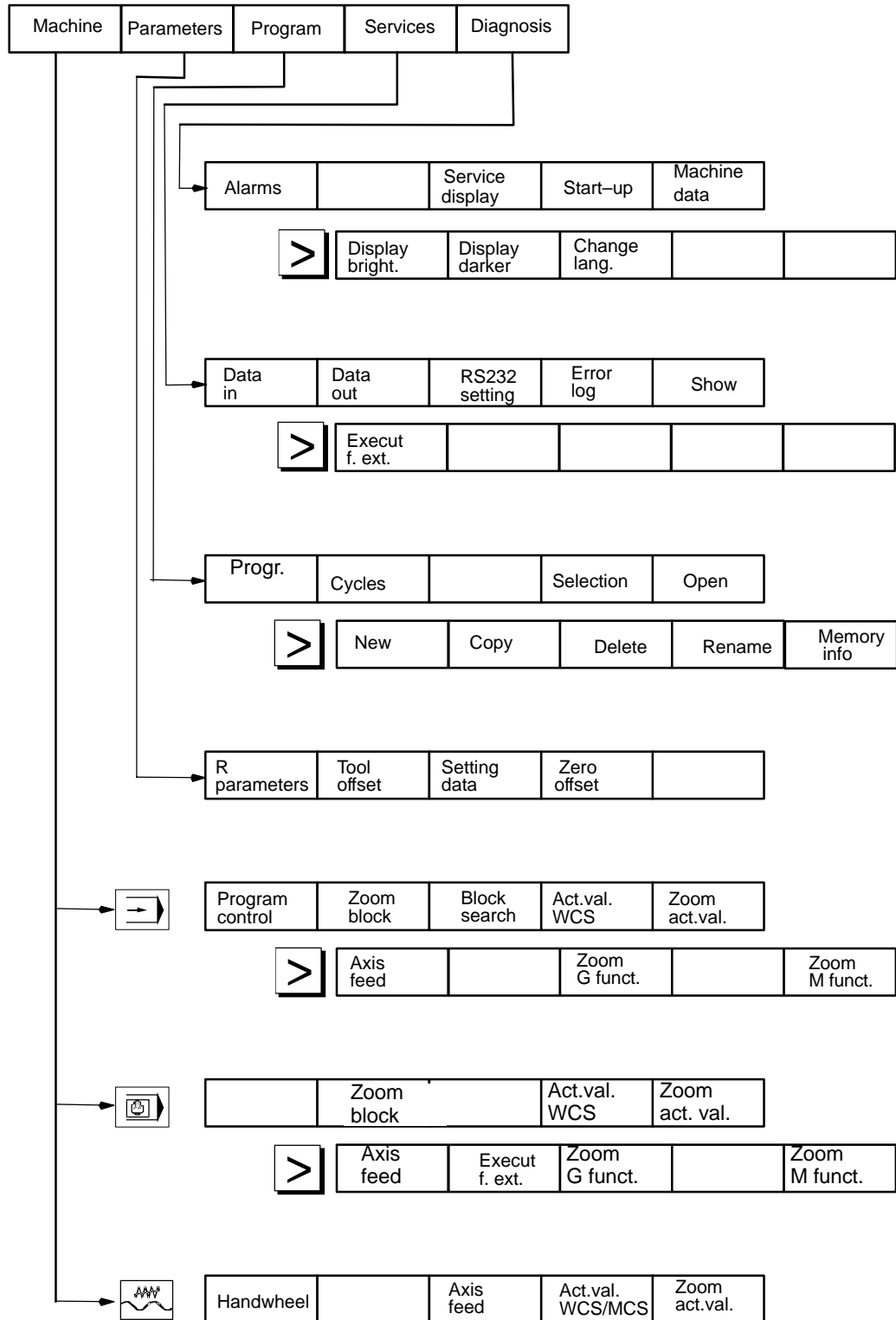
However, the operator can alter the classes of protection in the "Machine data display" menu in the "Diagnosis" operating area.

Default: protection level 3.

In the following menus, entering and modifying data depends on the set protection class:

- tool offsets
- zero offsets
- setting data
- V24 setting

1.3 Overview of the most important softkey functions



1.4 Pocket calculator



This function can be activated for all input fields intended for entry of numerical values by means of the “=” character. To calculate the required value, you can use the four basic arithmetic operations, and the functions sine, cosine, squaring, as well as the square root function.

If the input field is already loaded with a value, this function writes the value in the input line of the pocket calculator.



Fig. 1-3 Pocket calculator

Permissible character

The following characters are permitted for input:

- + Value X plus value Y
- Value X minus value Y
- * Value X multiplied with value Y
- / Value X divided by value Y

S Sine function

The value X in front of the input cursor is replaced by the value $\sin(X)$.

C Cosine function

The value X in front of the input cursor is replaced by the value $\cos(X)$.

Q Square function

The value X in front of the input cursor is replaced by the value X^2 .

R Square root function

The value X in front of the input cursor is replaced by the value \sqrt{X} .

Calculation examples

Task	Input
$100 + (67 \cdot 3)$	100+67*3
$\sin(45^\circ)$	45 <u>S</u> → 0.707107
$\cos(45^\circ)$	45 <u>C</u> → 0.707107
4^2	4 <u>Q</u> → 16
$\sqrt{4}$	4 <u>R</u> → 2

The calculation is carried out when the **Input** key is pressed. The function writes the result to the input field and automatically closes the pocket calculator.

To calculate auxiliary points on a contour, the pocket calculator provides the following functions:

- calculating the tangential transition between a circle sector and a straight line
- moving a point in a plane
- converting polar coordinates into Cartesian coordinates
- adding the second end point of a contour section 'straight line – straight line' given via angular interrelation.

These functions are directly linked with the input fields of the programming support. Any values in this input field are written by the pocket calculator into the input line, and the result is automatically copied into the input fields of the programming support.

Softkeys



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

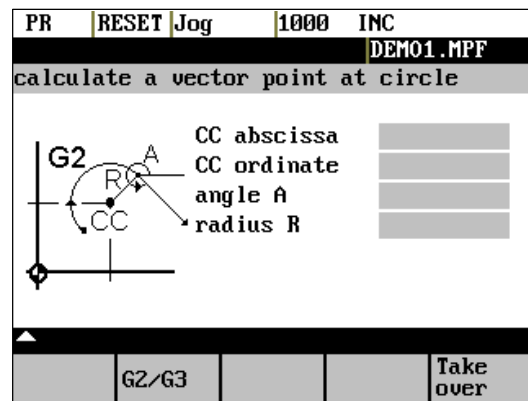


Fig. 1-4 Calculation of a point on a circle

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

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



The values of abscissa and ordinate are calculated. The abscissa is the first axis of the plane, and the ordinate the second axis of the plane.

Example

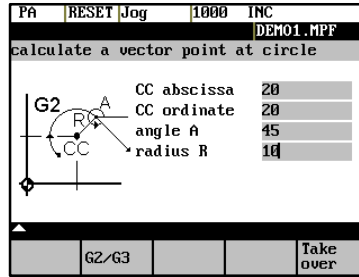
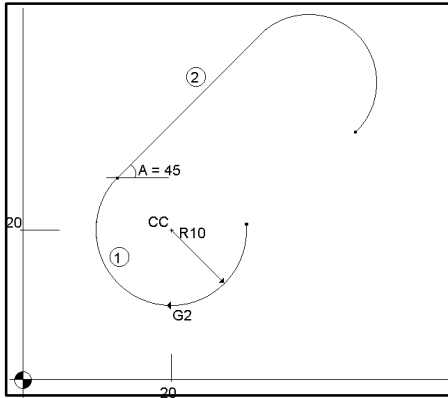
If plane G17 is active, the abscissa is the X axis, and the ordinate the Y axis.

The value of the abscissa is copied into that input field from which the pocket calculator function has been called, and the ordinate value into the next following input field.

Example

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

Given: Radius: 10
 Circle center point: X 20 Y20
 Ongoing angle of the straight line: 45°



Result: X = 12.928
 Y = 27.071



The function calculates the Cartesian coordinates from a straight line specified by length and rise angle.

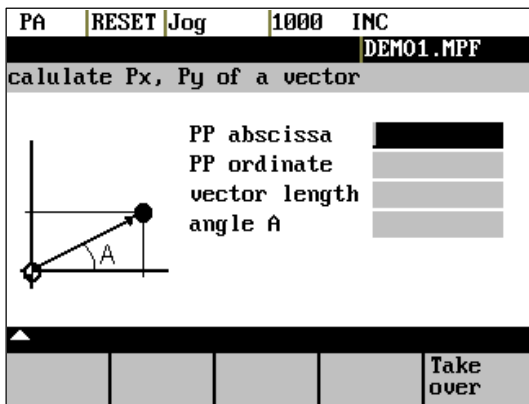


Fig. 1-5 Conversion of the polar coordinates into Cartesian coordinates

Enter the pole point (PP) as both an abscissa and ordinate value, the length and the rise angle of the straight line.

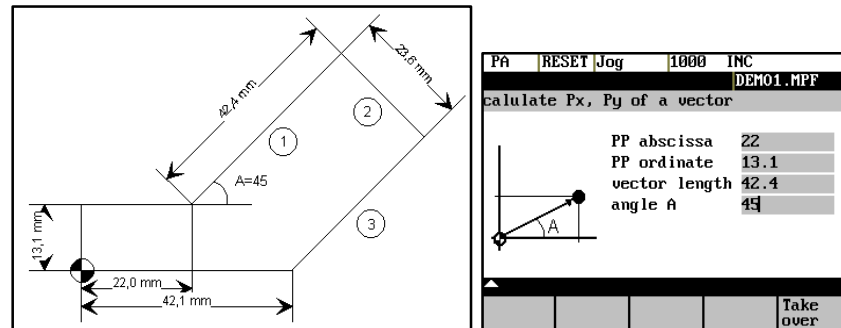


The values of abscissa and ordinate are calculated.

The abscissa value is copied into the input field from which the pocket calculator function has been called, and the ordinate value into the next following input field.

Example

Calculating the end point of the straight line ^①. The straight line is defined by the angle $A=45^\circ$ and its length..



Result: $X = 51.981$
 $Y = 43.081$



This function can be used to move a point in the plane. The point is on a straight line defined by its rise angle.

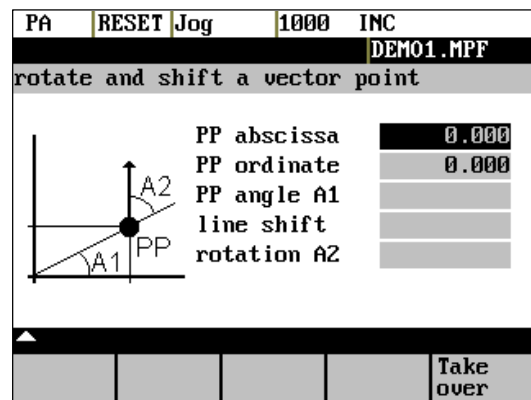


Fig. 1-6 Moving a point in the plane

Enter the rise angle of the straight line and the coordinates of the point.

Enter line shift and rotation of the point with refer to the straight line in the fields "line shift" and "rotation".

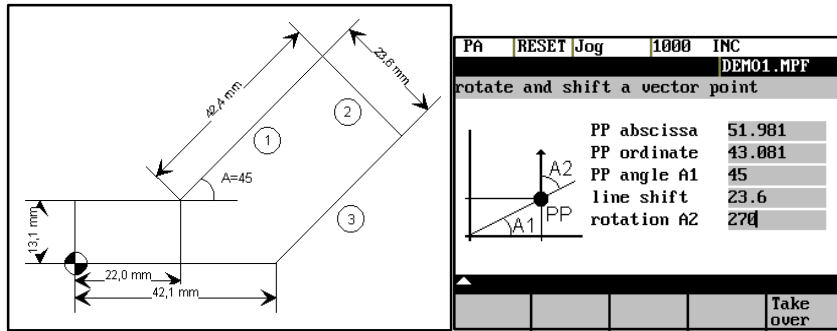
Take over

The values of abscissa and ordinate are calculated.

The pocket calculator copies the abscissa value into the input field from which the pocket calculator function has been called, and the ordinate value into the next following input field.

Example

Calculating the end point of the straight line ^②. The straight line stands vertical on the end point of the straight line ^① (coordinates: X = 51.981, Y = 43.081). The length of the straight line is also given.



Result: X = 68.668
Y = 26.393



This function calculates the missing end point of the contour section straight line – straight line, with the second straight line standing vertically on the first straight line.

The following values of the straight line are known:

Straight line 1: Starting point and rise angle

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

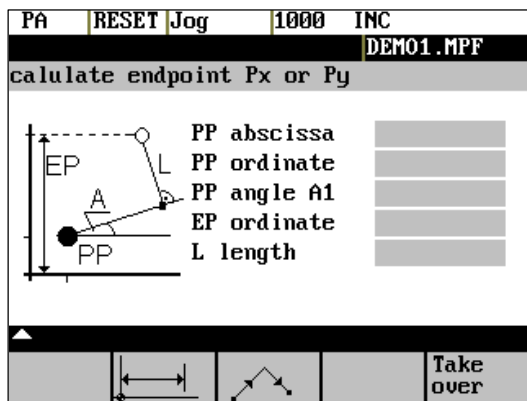
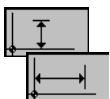
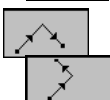


Fig. 1-7



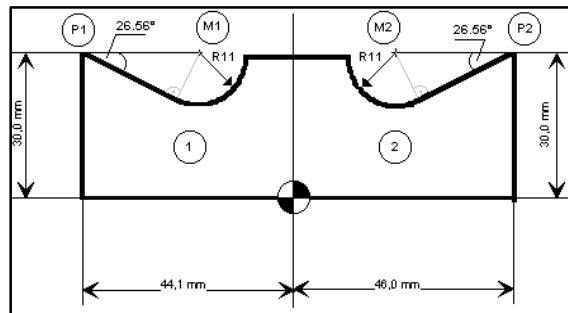
This function chooses the given coordinate of the end position. The value of ordinate and/or abscissa is given.




The second straight line is rotated in clockwise direction or, with refer to the first straight line, rotated by 90 degrees in counter-clockwise direction. The function chooses the appropriate setting.



The missing end position is calculated. The value of the abscissa is copied into that input field from which the pocket calculator function has been called, and the ordinate value into the next following input field.

Example

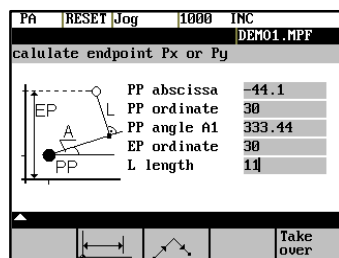
The drawing above must be added by the values of the of the circle center points to be able to calculate the intersection points between the contour sections. Calculating the missing coordinates of the center points is carried out with the pocket calculator function , since the radius in the tangential transition stands vertical on the straight line.

Calculating M1 in section 1:

In this section, the radius stands on the straight line section in counter-clockwise direction.

Use the softkeys  and  to select the given constellation.

Enter the coordinates, the pole point P1, the rise angle of the straight line, the given ordinate value and the circle radius as the length.




Result:

$$X = -19.449$$

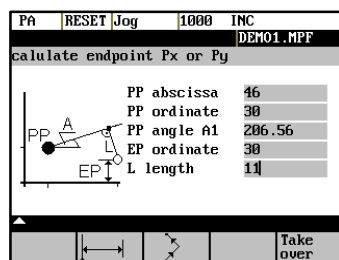
$$Y = 30$$

Calculating M2 in section 2:

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

Use the softkeys  to select the given constellation.

Enter the parameters in the screen form.



Result

$$X = 21.399$$

$$Y = 30$$

1.5 Basic principles

Coordinate systems

Right-handed, rectangular coordinate systems are used for machine tools. Such systems describe the movements on the machine as a relative motion between tool and workpiece.

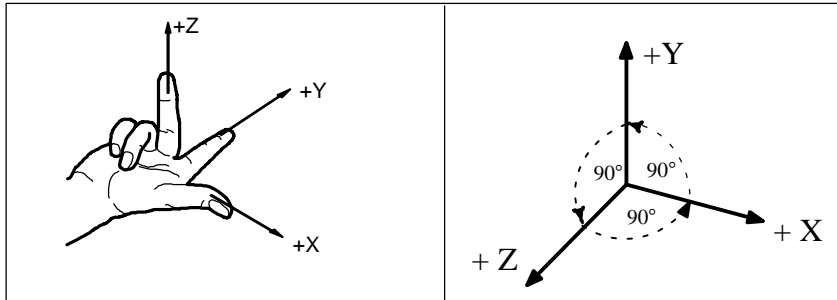


Fig. 1-8 Specification of mutual relationship between axis directions;

Machine coordinate system (MCS)

The orientation of the coordinate system on the machine tool depends on the particular machine type. It can be turned to various positions.

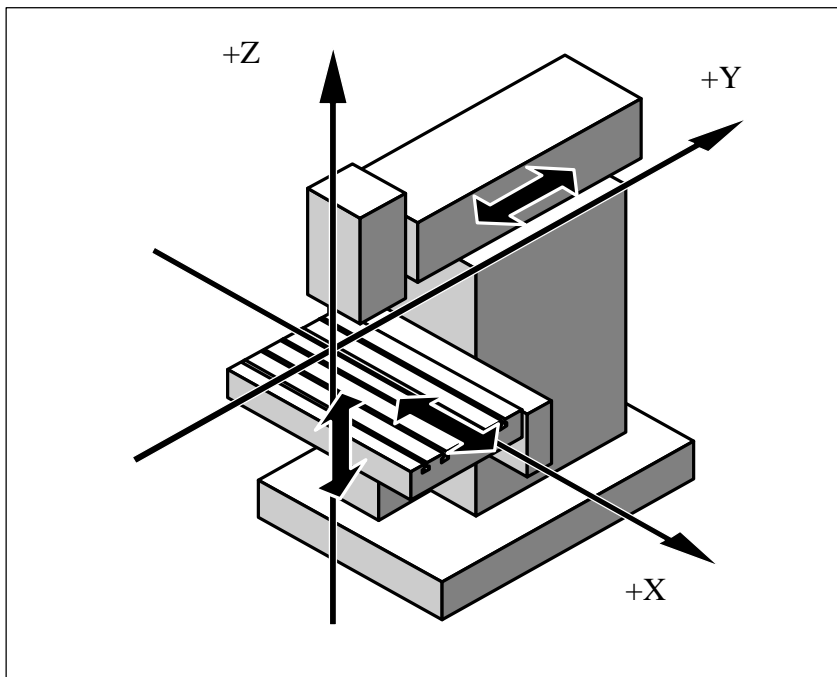


Fig. 1-9 Example of machine coordinates/axes

The origin of the coordinate system is the **machine zero**. All axes are in zero position at this point. This point is merely a reference point determined by the machine manufacturer. It need not to be approachable.

The traversing range of the **machine axes** can be negative.

Workpiece coordinate system (WCS)

The workpiece coordinate system described above (see Fig. 1-8) is also used to describe the geometry of a workpiece in the workpiece program.

The workpiece zero can be freely selected by the programmer. The programmer need not know the real movement conditions on the machine, i.e. whether the workpiece or the tool moves; this can be different in the individual axes.

The definition of the directions is based on the assumption that the workpiece does not move and the tool moves.

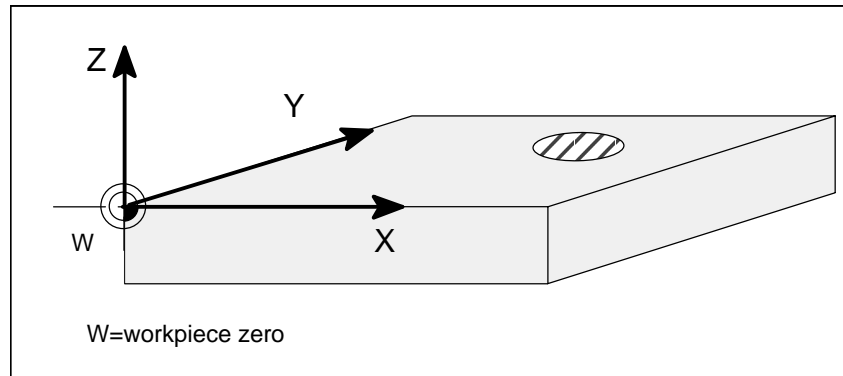


Fig. 1-10 Workpiece coordinate system

Current workpiece coordinate system

If the programmer feels that it is better to continue his geometrical descriptions from another zero than the initially selected zero (workpiece zero), he can define a new zero using the programmable zero offset. Reference is always made to the original zero.

The use of the programmable zero offset provides a completely new current workpiece coordinate system. The current workpiece coordinate system can also be turned to the original workpiece coordinate system (see Section "Programmable Zero Offset and Rotation").

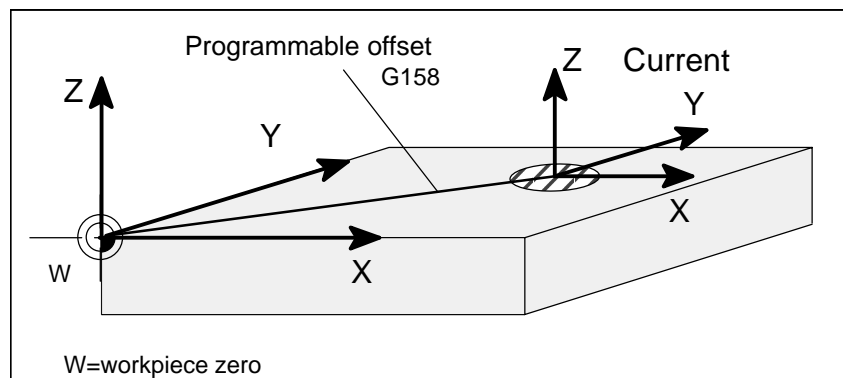


Fig. 1-11 Coordinates on the workpiece, current workpiece coordinate system

Workpiece clamping

To machine the workpiece, it is clamped on the machine. The workpiece must be aligned such that the axes of the workpiece coordinate system run in parallel with the machine axes. Any resultant offset of the machine zero is determined for each axis and entered into the intended data areas for the settable zero offset. This offset is activated during the NC program execution by means, for example, of a programmable G54 (see Section "Workpiece Clamping – Settable Zero Offset ...").

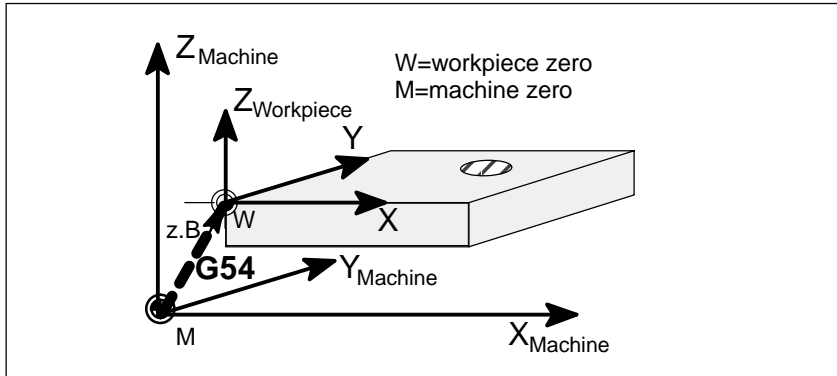


Fig. 1-12 Workpiece on the machine

Turning On and Reference Point Approach

Note

Before you switch on the SINUMERIK and the machines, you should also have read the machine documentation, since turning on and reference point approach are machine-dependent functions.

Operating sequence

First switch on the power supply of the CNC and of the machine. After the control system has booted, you are in the “Machine” operating area, in the **Jog** operating mode.

The *Reference point approach* window is active.

MA	RESET	JOG	REF	DEMO1.MPF	
Reference point mm			F:mm/min		
+ X		0.000	Act:		
+ Y		0.000	0.000		
+ Z		0.000	Prg:		
+ SP		0.000	0.000		
S	0.000	0.000	T: 0	D: 0	

Fig. 2-1 Jog Ref basic screen

Reference-point approach can only be executed in the **Jog** mode.



Activate the “Approach reference point” function by selecting the **Ref** key on the machine control panel.

In the “Reference point approach” window (Fig. 2-1), it is displayed whether or not the axes have to be referenced.

- Axis has to be referenced
- Axis has reached the reference point



Press the direction keys.

The axis does not move if you select the wrong direction.

Approach the reference point in each axis successively.

You can quit the function by selecting another operating mode (**MDA**, **Automatic** or **Jog**).

Setup

Preliminary remarks

- Before you can use the CNC, set up the machine, tools, etc. on the CNC by:
- entering the tools and tool offsets
- entering/modifying the zero offset
- entering the setting data

3.1 Entering tools and tool offsets

Functionality

The tool offsets consist of several data that describe the geometry, wear and tool type.

Each tool has a defined number of parameters depending on the tool type.

Each tool is identified by its own tool number (T number).

See also Section 8.6 "Tool and Tool Offset".

Operating sequences

Parameter

Tool
Corr.

This function opens the *Tool Compensation Data* window, which contains the offset values of the currently active tool. If you select another tool using the <<T or T>> softkeys, the setting remains when you quit the window.

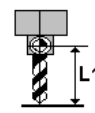

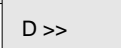


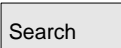

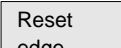
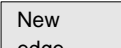
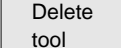
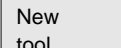
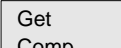
PA	RESET	Jog	DEMO1.MPF		
Tool compensation data			T type: 200		
No. c. edges :2			T No : 2		
D -- number :1					
	mm	Geometry	Wear		
	Leng .1	0.000	0.000		
	Leng .2	0.000	0.000		
	Leng .3	0.000	0.000		
<< D		D >>	<< T	T >>	Search
Reset edge	New edge	Delete tool	New tool	Get Comp.	

Fig. 3-1 Tool list

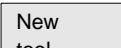
Softkeys

	Select next lower or next higher edge number.
	
	Select next lower or next higher tool.
	
	Pressing this softkey opens the dialog box and the overview of the tool numbers assigned. Enter the tool number you search for in the input window and start search with OK . If the searched tool exists, the search function opens the tool offset data box.
	Press the ETC key to extend the softkey functions.
	All edge compensation values are reset to zero.
	Creates a new edge and loads it with the appropriate parameters. The new edge is created for the currently displayed tool; it is automatically assigned the next higher edge number (D1 - D9). Max. 30 edges (in total) can be stored in the memory.
	Deletes the tool compensation data of all edges of the selected tool.
	Creates new tool compensation data for a new tool. Note: Max. 20 tools can be created.
	Determines the length compensation values.

3.2 Creating a new tool

Operating sequence

Press this softkey to create a new tool.

	Pressing this softkey opens the input window and an overview of the tool numbers assigned.
---	--

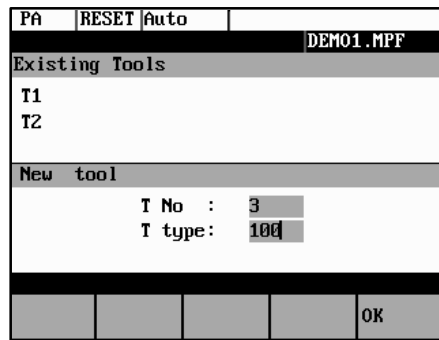
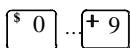
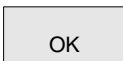


Fig. 3-2 New Tool window



Enter the new T number (in the range from 1 to 32000) and specify the tool type.



Press **OK** to confirm your entry; the *Tool Compensation Data* window is opened.

3.3 Tool compensation data

The tool compensation data are divided into length and radius compensation data. The list is structured according to the tool type.

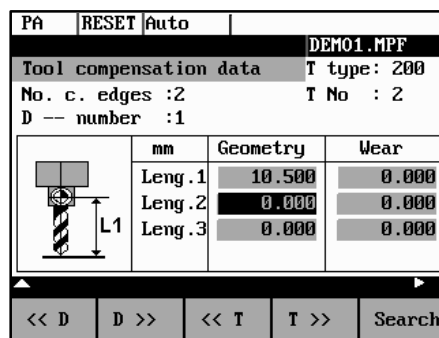
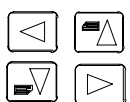


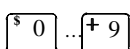
Fig. 3-3 Tool compensation data

Operating sequence

Enter the offsets by



positioning the cursor on the input field to be modified,



entering value(s)



and confirming your entry by pressing **Input** or a cursor selection.

3.4 Determining the tool offsets

Functionality

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

Prerequisite

The appropriate tool has been changed. In JOG mode, approach a point on the machine, from which you know the machine coordinates, with the edge of the tool. This can be a workpiece with a known position. The machine coordinate value can be split into two components: stored zero offset and offset.

Procedure

Enter the offset value in the intended Offset field. Then select the required zero offset (e.g. G54) or G500 if no zero offset is to be calculated. These entries must be made for each selected axis (see Fig. 3–5).

Please note the following: For milling tools, length 1 and the radius must be determined, and for drilling tools only length 1.

Using the actual position of point F (machine coordinate), the offset entry and the selected zero offset Gxx (position of the edge), the control system can calculate the assigned compensation value of length 1 or the tool radius.

Note: You can also use a zero offset already determined (e.g. G54 value) as the known machine coordinate. In this case, approach the workpiece zero with the edge of the tool. If the edge stands directly at the workpiece zero, the offset value is zero.

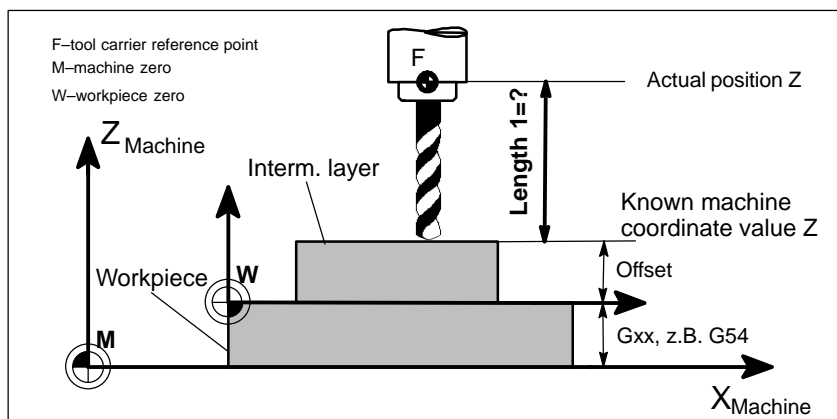
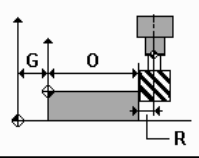


Fig. 3-4 Determination of length compensation using the example of a drill: length 1/Z axis

Operating sequence

Get
Comp.

Select the softkey **Get Comp.** The window *Compensation values* opens.

PA	RESET	Jog	10000	INC
DEMO1.MPF				
Reference		T No	:1	mm
G = G500 / G54-57				
		Axis X	0.000	
		Offset	0.000	
		G	500	0.000
		R		1.400
Next Axis		Calculate	OK	

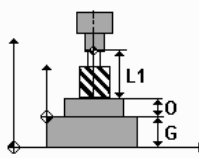
PA	RESET	Jog	10000	INC
DEMO1.MPF				
Reference		T No	:1	mm
G = G500 / G54-57				
		Axis Z	0.000	
		Offset	0.000	
		G	500	0.000
		L1		1.100
Next Axis		Calculate	OK	

Fig. 3-5 Window *Compensation values*

- Enter **offset** if the tool edge cannot approach the zero point **Gxx**. If you work without zero offset, select G500 and enter **offset**.
- When the softkey **Calculate** is pressed, the control system determines the searched geometry length 1 or the radius depending on the preselected axis. This geometry is calculated on the basis of the approached actual position, the selected Gxx function and the entered offset value.
The determined compensation value is stored.

3.5 Entering/modifying zero offsets

Functionality

The actual-value memory and thus also the actual-value display are referred to the machine zero after the reference-point approach. The workpiece machining program, however, refers to the workpiece zero.

This offset must be entered as the zero offset.

Operating sequenc

- Parameter Use the **Parameter** and **Zero Offset** softkeys to select the zero offset.
- Zero offset An overview of settable zero offsets appears on the screen .

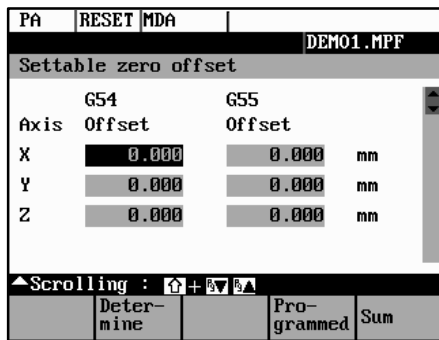
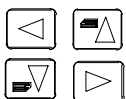
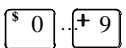


Fig. 3-6 Zero Offset window



Position the cursor bar on the input field to be altered,



enter value(s).



The next zero offset overview is displayed by **Page down**. G56 and G57 are now displayed.



Return to next-higher menu level, without saving the zero offset values.

Softkeys

- Determine Use this function to determine the zero offset with refer to the coordinate origin of the machine coordinate system. When you have selected the tool, which you want to use for measuring, you can set the appropriate conditions in the *Determine* window.

PA	RESET	Auto		
DEMO1.MPF				
Determine zero offset				
Offset		Axis	Position	
G54	0.000 mm	X	0.000 mm	
Tnum:1	Dnum:1	Ttyp: 100		
Length :	+ <input type="checkbox"/>	0.000 mm		
Radius :	+ <input type="checkbox"/>	0.000 mm		
Offset :	0.000 mm			
▲				
Next UFrame	Next Axis		Calculate	OK

Fig. 3-7 Zero offset measuring using the *Determine* function

The toggle fields can be used to calculate the tool compensation values.

It is possible to specify an additional length in the Offset box, which must then be considered in the calculation (for example, when using a spacer).

The current axis position, the active compensation value and the tool compensation data are displayed.

Move the tool to the selected zero and set all compensation values for the selected axis. The **Calculate** softkey function will then calculate the offset and enter the value in the respective field. This process must be repeated for all axes.

Next
UFrame

Selects the next settable zero offset.

Next
Axis

Selects the next axis.

Calcu-
late

The compensation values are calculated with the Offset field and the current axis position (MCS). The result will be assigned to the selected axis as an offset value.

OK

Closes the window.

Pro-
grammed

Opens a window with the programmed zero offset. The value shown in the window cannot be edited.

Sum

Displays the total of all active zero offsets. The values cannot be edited.

3.6 Determining the zero offset

Prerequisite

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

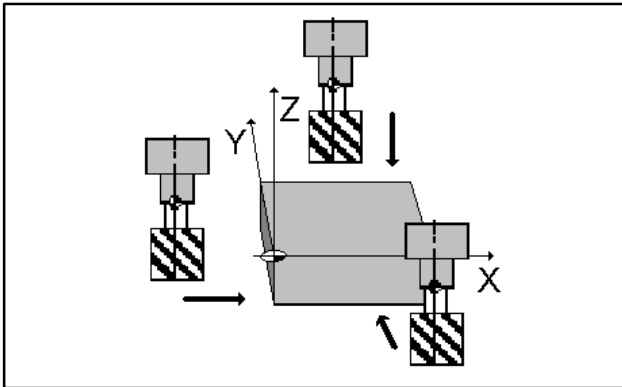


Fig. 3-8 Determining the zero offset for the Z axis

Approach

- A zero offset can only be determined with a known (entered geometry) and active tool. Enter the active tool in the dialog box. Press **OK** to take over the tool; the *Determine* window is then opened.
- The selected axis appears in the Axis area.
The actual position of the tool support reference point (MCS) associated to the axis is displayed in the adjacent field.
- D number 1 is displayed for the tool edge.
If you have entered the valid offsets for the used tool under a D number other than D1, enter that D number here.
- The stored tool type is displayed automatically.
- The effective length compensation value (geometry) is displayed.
- Select the sign (–, +) for calculating the length offset, or select “without” taking the length offset into account.
A negative sign subtracts the length offset value from the actual position.
- If the tool can neither reach, nor “scrape” the desired position, an offset value can be entered in the Offset field.
- Approach the coordinates of the intended workpiece zero offset (if necessary with consideration of the entered offset value) in JOG mode.
- The resulting zero offset is determined from the actual position and all active compensation values by means of the Calculate function.

PA	RESET	Auto	DEMO1.MPF	
Settable zero offset				
	G54	G55		
Axis	Offset	Offset		
X	0.000	0.000	mm	
Tool number				
Select tool number !				
	2			
				OK

Fig. 3-9 Select Tool screen form

PA	RESET	Jog	DEMO1.MPF	
Determine zero offset				
	Offset	Axis	Position	
G54	0.000 mm	X	0.000 mm	
Tnum:1	Dnum:1	Ttyp: 100		
Length :	+ U	0.000 mm		
Radius :	+ U	0.000 mm		
Offset :	0.000 mm			
Next UFrame	Next Axis		Calculate	OK

Fig. 3-10 Determine Zero Offset form

Next
UFrame

Softkey can be used to select the zero offsets **G54** to **G57**. The selected zero offset is displayed on the selected softkey.

Calcu-
late

Pressing the **Calculate** softkey calculates the zero offset.

OK

Press the **OK** softkey to quit the window.

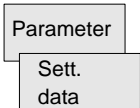
3.7 Programming the setting data – “Parameters” operating area

Functionality

Use the setting data to define the settings for the operating states. These can also be modified if necessary.

Operating sequence

Use the **Parameter** and **Setting Data** softkeys to select *Setting Data*.
The **Setting Data** softkey branches to another menu level in which various control options can be set.

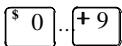


PA	RESET	Jog	10000	INC
Jog data		Spindle data		
Jog feedrate:		Minimum: 1 rpm		
100.000 mm/min		Maximum: 1000 rpm		
Spindle speed :		Program: 25 rpm		
5 rpm				
Dry run feedrate		Start angle		
250.500 mm/min		360.000 °		
Jog data	Spindle data	Dry feed	Start angle	

Fig. 3-11 Setting Data basic screen



Use the **paging keys** to position the cursor on the desired line within the display areas.



Enter the new value in the input fields.



Use **Input** or the cursor keys to confirm.

Softkeys



This function can be used to change the following settings:

Jog feed

Feed value in Jog mode

If the feed value is zero, the control system uses the value stored in the machine data.

Spindle

Spindle speed

Direction of rotation of the spindle

Spindle
data

Minimum / Maximum

Limits for the spindle speed set in the Max. (G26)/Min. (G25) fields must be within the limit values specified in the machine data.

Programmed (LIMS)

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

Dry
feed

Dry-run feedrate for dry-run operation (DRY)

The feedrate you enter here is used in the program execution instead of the programmed feed during the Automatic mode when the Dry-Run Feedrate is active (see Program Control, Fig. 5-3).

Start
angle

Start angle for thread cutting (SF)

A start angle representing the starting position for the spindle is displayed for thread cutting operations. It is possible to cut a multiple thread by altering the angle and repeating the thread cutting operation.

3.8 R parameters - “Parameters” operating area

Functionality

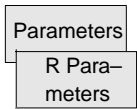
All R parameters (arithmetic parameters) that exist in the control system are displayed on the R Parameters mainscreen as a list (see also Section 8.8 “Arithmetic Parameters /R Parameters”).

These can be modified if necessary.

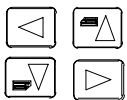
PA	RESET	Jog	10000	INC
R Parameters				
R0	0.000000	R1	0.000000	
R2	0.000000	R3	0.000000	
R4	0.000000	R5	0.000000	
R6	0.000000	R7	0.000000	
R8	0.000000	R9	0.000000	
R10	0.000000	R11	0.000000	
R12	0.000000	R13	0.000000	
R Parameter	Tool Corr.	Setting data	Zero offset	

Fig. 3-12 R Parameters window

Operating sequence



Use the **Parameter** and **R Parameter** softkeys



to position the cursor on the input field that you want to edit.



Enter value(s).



Press **Input** or use the cursor keys to confirm.

Manually Operated Mode

Preliminary remarks

The manually operated mode is possible in the Jog and MDA mode.

In the **Jog** mode, you can traverse the axes, and in the **MDA** mode, you can enter and execute individual part program blocks.

4.1 Jog mode - "Machine" operating area

Functionality

In the **Jog** mode, you can

- traverse the axes and
- set the traversing speed by means of the override switch, etc.

Operating sequence



Use the **Jog** key on the machine control panel to select the Jog mode.



To traverse the axes, press the appropriate axis direction keys.

As long as the direction key is pressed and hold down, the axes traverse continuously at the speed stored in the setting data. If this setting is zero, the value stored in the machine data is used.



If necessary use the override switch to set the traversing speed.



If you press the **Rapid Traverse Override** key at the same time, the selected axis is traversed at rapid traverse speed for as long as both keys are pressed down.



In the **Incremental Feed** operating mode, you can use the same operating sequence to traverse the axis by settable increments. The set increment is displayed in the display area. **Jog** must be pressed again to cancel.

4.1 Jog mode - "Machine" operating area

The Jog basic screen displays position, feed and spindle values as well as the current tool.

MA	RESET	Jog		
			DEMO1.MPF	
MCS	Act	Repos.mm	F:mm/min	
+ X	0.000	0.000	Act: 0.000	
+ Y	0.000	0.000	Prg: 0.000	
+ Z	0.000	0.000		
+ SP	0.000	0.000		
S	0.000	0.000	T: 0 D: 0	
Hand wheel		Axis feed.	Act.val WCS	Zoom act.val

Fig. 4-1 Jog basic screen

Parameters

Table 4-1 Description of parameters in the Jog basic screen

Parameter	Explanation
MKS X Y Z	Display of addresses of existing axes in machine coordinate system (MCS).
+X - Z	If you traverse an axis in the positive (+) or negative (-) direction, a plus or minus sign appears in the respective field. No axis is displayed, if the axis is in position.
Act. mm	The current position of the axes in the MCS or WCS is displayed in these fields.
Repos offset	If the axes are traversed in the Jog mode in the Program Interrupted condition, the distance traversed by each axis in relation to the break point is displayed in this column.
Spindle S rpm	Display of actual value and setpoint of spindle speed
Feed F mm/ min	Display of path feed actual value and setpoint
Tool	Display of currently active tool with the current cutting edge number

Softkeys

Hand-wheel

Call the Handwheel window.

Axis feed

Call the Axis Feed or Interp. Feed window.

Interp./feed

Use this softkey to change between the Axis Feed window and the Interp. Feed window.

The softkey label changes to **Interp. feed** when the Axis/Feed window is opened.

Act. val.
WCS
Act.val.
MCS

The **actual values are displayed** as a function of the selected coordinate system. There are two different coordinate systems, i.e. the machine coordinate system (MCS) and the workpiece coordinate system (WCS).

The softkey changes between MCS and WCS. When doing this, the softkey label changes as follows:

- The values of the machine coordinate system are selected, the softkey label changes to **Act. val. WCS.**
- When the workpiece coordinate system is selected, the label changes to **Act. val. MCS.**

Zoom
act.val.

Enlarged view of actual values

4.1.1 Assigning handwheels

An axis is assigned to the respective handwheel and becomes active as soon as you press **OK.**

Operating sequence



In **Jog** mode, call the *Handwheel* window.

Hand-wheel

After the window has opened, all axis identifiers are displayed in the Axis column and also appear in the softkey bar. Depending on the number of connected handwheels, it is possible to change from handwheel 1 to handwheel 2 using the cursor.



Place the cursor on the line with the handwheel to which you wish to assign an axis. Then select the softkey that contains the name of the axis.



The symbol  appears in the window.

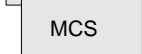
MA	RESET	Jog	100	INC
DEMO1.MPF				
Handwheel				
Number	Axis		MCS	
	X	Y	Z	SP
1		<input checked="" type="checkbox"/>		
▲				
WCS	X	Y	Z	OK

Fig. 4-2 Handwheel window

4.1 Jog mode - "Machine" operating area



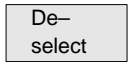
The **WCS/MCS** softkey is used to select the axes from the machine or workpiece coordinate system for assignment to the handwheel. The current setting is displayed in the handwheel window.



Use the **OK** softkey to take over the selected setting; the window is then closed.



Menu extension



The assignment you have made is reset for the selected handwheel.

4.2 MDA Mode (Manual Data Input) - "Machine" operating area

Functionality

You can create and execute a part program block in the MDA mode.

Contours that require several blocks (e.g. roundings, chamfers) cannot be executed/programmed.



Caution

This mode is protected by the same safety interlocks as fully automatic mode.

Furthermore, the MDA mode is subject to the same prerequisites as the fully automatic mode.

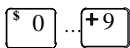
Operating sequence



Use the **MDA** key on the machine control panel to select the **MDA** mode.

MA	RESET	MDA	DEMO1.MPF	
MCS	Act	Dist	mm	F:mm/min
+ X	0.000	0.000		act:
+ Y	0.000	0.000		0.000
+ Z	0.000	0.000		Prg:
+ SP	0.000	0.000		0.000
S	0.000	0.000		T: 0 D: 0
	Zoom block		Act.val WCS	Zoom act.val
Axis feed.		Zoom G funct		Zoom M funct

Fig. 4-3 MDA basic screen



Enter a block using the control keyboard.



The entered block is executed by pressing **NC START**. The block cannot be executed while machining is taking place.

After processing, the contents of the input field remains stored so that the block can be traversed with new **NC Start**. The block is deleted by entering any new character.

Parameters

Table 4-2 Description of the parameters in the **MDA** working window.

Parameter	Explanation
MCS X Y Z	Display of existing axes in MCS or WCS
+X -Z	If you traverse an axis in the positive (+) or negative (-) direction, a plus or minus sign appears in the respective field. No sign is displayed if the axis is in position.
Act. value mm	The current position of the axes in the MCS or WCS is displayed in these fields.
Spindle S rpm	Display of actual value and setpoint of spindle speed
Feed F	Display of path feed actual value and setpoint in mm/min or mm/rev.
Tool	Display of currently active tool with the current tool edge number (T..., D...).
Edit window	In the Stop or Reset program state, an edit window is provided for input of the part program block.

Softkeys

- Zoom
block

The window shows the currently edited block full length.
- Act.val.
WCS

Act.val.
MCS

The actual values for the **MDA** mode are displayed as a function of the selected coordinate system. There are two different coordinate systems, i.e. the machine coordinate system (MCS) and the workpiece coordinate system (WCS).
- Zoom
act.val.

Enlarged view of the actual values
- >

Menu extension
- Axis
feed

Interp.
feed

Display of *Axis Feed* or *Interp. Feed* window

This softkey can be used to change between the two windows. The softkey label changes to **Interp. Feed** when the Axis Feed window is opened.
- Zoom
G funct.

The G Function window contains all active G functions. Each G function is assigned to a group and has a fixed position in the window.
More G functions can be displayed by pressing the **PAGE UP** or **PAGE DOWN** keys. You can exit the window by pressing **Recall**.
- Zoom
M funct.

Opens the M function window for displaying all active M functions of the block.

Automatic Mode

Functionality

In the Automatic mode, part programs can be executed fully automatically, i.e. this is the operating mode for standard processing of part programs.

Preconditions

The preconditions for executing part programs are:

- Reference point approached.
- You have already stored the required part program in the control system.
- You have checked or entered the necessary offset values, e.g. zero offsets or tool offsets.
- The required safety interlocks are activated.

Operating sequence



Use the **Automatic** key to select the **Automatic** mode.

The *Automatic* basic screen appears that displays the position, feed, spindle, and tool values, as well as the current block.

MA	RESET	Auto	DEMO1.MPF	
MCS	Act	Dist	mm	F:mm/min
+ X	20.000	0.000		Act:
+ Y	34.000	0.000		44.000
+ Z	0.000	0.000		Prg:
+ SP	0.000	0.000		44.000
S	0.000	0.000	T: 0	D: 0
>G2 G90 X100 Y60 I39.351128 J-7.175564				
Progr. control	Zoom block	Search	Act. val WCS	Zoom act. val
Axis feed.	Execu	Zoom G funct		Zoom M funct

f. ext.

Fig. 5-1 *Automatic* basic screen


Parameters


Table 5-1 Description of the parameters in the working window

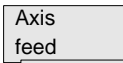
Parameter	Explanation
MCS X Y Z	Display of existing axes in MCS or WCS.
+ X - Z	If you traverse an axis in the positive (+) or negative (-) direction, a plus or minus sign appears in the respective field. No sign is displayed if the axis is in position.
Act. val. mm	The current position of the axes in the MCS or WCS is displayed in these fields.
Distance to go	The remaining distance to be traversed by these axes in the MCS or WCS is displayed in these fields.
Spindle S rpm	Display of actual value and setpoint of spindle speed
Feed F mm/min or mm/rev	Display of path feed actual value and setpoint
Tool	Display of currently active tool with the current cutting edge number (T..., D...).
Current block	The block display contains the current block. The block is output in one line only and truncated if necessary.

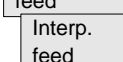
Softkeys

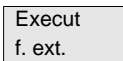
Progr. control	The window to select Program Control (e.g. skip block, program test) appears on the screen.
Zoom block	This window displays the previous, current and next block in full. In addition, the names of the current program or subroutine are displayed.
Search	Use the Block Search function to jump to the desired point in the program.
Search	The Search softkey provides the functions "Find line" and "Find text".
Interr. point	The cursor is positioned to the main program block of the breakpoint ("interrupt point"). The search target is automatically set in the subroutine levels.
Contin. search	Continue search
Start B search	The Start B Search softkey starts the search process in which the same calculations are carried out as in normal program mode, but without axis movements. The block search can be canceled by NC Reset.
Act.val. WCS	The values of the machine or workpiece coordinate system are selected. The softkey label changes to Act. val. WCS or Act. val. MCS .
Act.val. MCS	

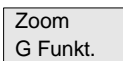
 Zoom act.val. Enlarged view of actual values

 Menu extension

 Axis feed When pressing these softkeys, the *Axis Feed* or *Interp. Feed* window appears.

 Interp. feed This softkey can be used to change between the windows. The softkey label changes to **Interp. feed** when the *Axis Feed* window is opened.

 Execut f. ext. An external program is transferred to the control system via the V.24 interface and executed immediately by pressing **NC Start**.

 Zoom G Funkt. Opens the *G Function* window to display all active G functions.

The *G Function* window contains all active G functions. Each G function is assigned to a group and has a fixed position in the window. More G functions can be displayed by pressing the **PAGE UP** or **PAGE DOWN** keys.


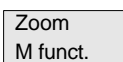
MA	RUN	Auto	
			DEMO1.MPF
active G functions			
1:G1	2:	3:	
4:	5:	6:	
7:	8:G94	9:	
10:	11:	12:	
13:	14:	15:	
16:			
▲Scrolling : 			

Fig. 5-2 Active G Functions window

 Zoom M funct. Opens the *M Function* window to display all active M functions.

5.1 Selecting/starting a part program - "Machine" operating area

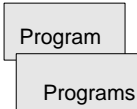
Functionality

The control system and the machine must be set up before the program is started. Please note the safety instructions provided by the machine manufacturer.

Operating sequence



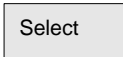
Use the **Automatic** key to select the **Automatic** mode.



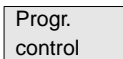
An overview of all programs stored in the control system is displayed.



Position the cursor bar on the desired program.



Use the **Select** softkey to select the desired program for processing.



If necessary now you can make settings for program control.

The following program control functions can be activated and deactivated:

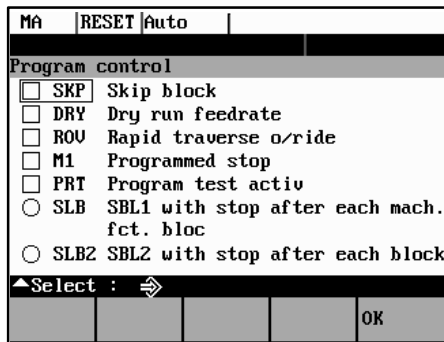


Fig. 5-3 Program Control window



The part program is executed when **NC START** is pressed.

5.2 Block search - "Machine" operating area

Operating sequence

Precondition: The desired program has already been selected (cf. Section 5.1), and the control system is in the reset state.

Search

The block search function can be used to advance the program up to the desired point in the part program. The search target is set by positioning the cursor directly on the desired block in the part program.

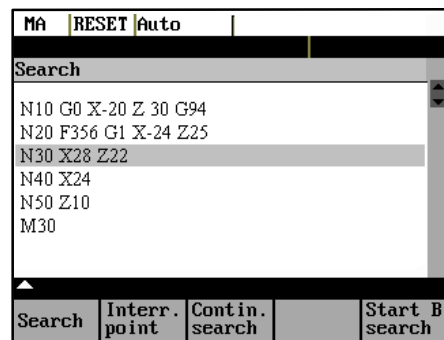


Fig. 5-4 Search window

Start B search

The funktion starts program advance and closes the *Search* window.

Search result

The desired block is displayed in the *Current Block* window.

5.3 Stopping/cancelling a part program - "Machine" operating area

Functionality

Part programs can be stopped and aborted.

Operating sequence



The execution of a part program can be interrupted by selecting **NC STOP**. The interrupted program can be continued by **NC START**.



The current program can be aborted by pressing **RESET**. When you press **NC START** again, the aborted program is restarted and executed from the beginning.

5.4 Repositioning after interruption

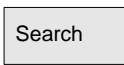
Functionality

After a program interruption (NC STOP), you can move the tool away from the contour in the manual mode (Jog). The control system stores the coordinates of the breakpoint ("interrupt point"). The path differences traversed by the axes are displayed.

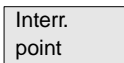
Operating sequence



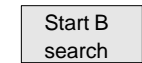
Select the **Automatic** mode.



Open the *Block Search* window to load the breakpoint.



The breakpoint is loaded. The routine is adjusted to the start position of the interrupted block.



A block search to the breakpoint is started.



Continue execution of the program by **NC START**.

5.5 Program execution from external (V.24 interface)

Functionality

An external program is transferred into the control system via the V.24 interface and executed immediately by pressing NC Start.

While processing the contents of the buffer memory, the program is automatically reloaded. For example, as an external device, a PC can be used, on which the PCIN tool for data transfer is installed.

Operating sequence

Prerequisite: The control system is reset.

The V.24 interface is parameterized correctly (see Chapter 7) and not occupied by any other application (DataIn, DataOut, STEP7).

Execute
f. ext.

Press this softkey.

Use PCIN on the external device (PC) to set the program for data output active.

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

For the program execution, it is advantageous to wait until the buffer memory is filled.



The program execution starts with **NC START**. The program is reloaded continuously.

Either at the end of the program or when pressing **RESET**, the program is automatically removed from the control system.

Note

- As an alternative, **External Program Execution** can also be activated in the **Services** area.
 - Any transfer errors are displayed in the **Services** area when you press the **Error log** softkey.
-

5.6 Teach In

Functionality

Use the submode Teach In to accept the axis position values directly into a parts program block to be generated or modified.

The axis positions are approached either in Automatic mode by traversing the JOG keys or by using the handwheel. However, first press the appropriate softkey (see below) in the Programming operating area to enable the submode Teach In.

Operating sequence

Prerequisite:

- Teach-in option is set;
- The control system is either in the state Stop or Reset.

Programs	A list of all programs existing in the control system is displayed.
open	Pressing Open calls the editor for the selected program and opens the editor window.
>	Menu extension
Edit	Select
>	Menu extension
Teach In on	Select

PR	RESET	TEACHIN	DEMO1.MPF	
PP Editor: DEMO1.MPF 3				
ANF: G1 G94 X78 F3000 T1=1 D1=1"				
G5 Z-8.000 X21.000 KZ=0.000 IX=8.000"				
GNA :X78 Z75"				
M51 Z0 M3 S1000"				
M76 G1 X10.000 Z-2.000"				
M80 GOTOB ANA"				
G1 Z0.000 X0.000"				
T2 D3"				
Technol Data	TeachIn Records	Finish Record	Progr. run	TeachIn Off

Fig. 5-5 Teach In basic screen

Softkeys

Technol. Data	Use this softkey to generate a block with technological data.
------------------	---

Use this screen form to enter

- feed value
- spindle speed and direction of rotation (CW; CCW; stop)
- tool and edge number
- machining level
- Feed mode (active; mm/min corresponds to G64; mm/rev. of spindle corresponds to G96)
- Positioning behavior (active; exact stop G60; continuous-path control mode G64)

PR	RESET	TEACHIN	DEMO1.MPF	
PP Editor:		DEMO1.MPF	9	
G5 Z-8.000 X21.000 KZ=0.000 IX=8.000"				
Generate a technolog. record				
F:	850	mm/min	3	T-Nr.
S:	700	U/min	2	D-Nr.
⊕	left	U	Plane	Z_X U
F-Mode:	mm/min		U	
Approach behavior	continuous path U			
				OK

Fig. 5-6 Input screen form for technological data

When you press **OK**, a block with the technological data entered is generated and inserted in front of the block to which the cursor is positioned. Pressing **RECALL** cancels your entry and lets you return to the *Teach In* basic screen.

Teach In
Records

Use this softkey to generate NC blocks using the traversing keys or the handwheel.

Simple NC blocks are generated by traversing with parallel axes using either the traversing keys of the axes or the handwheel. It is also possible to correct the values of an existing blocks.

PR	RESET	TEACHIN	DEMO1.MPF	
PP Editor:		DEMO1.MPF	4	
ANF: G1 G94 X78 F3000 T1=1 D1=1"				
G5 Z-8.000 X21.000 KZ=0.000 IX=8.000"				
:X78 Z75"				
Teach In Axis				
X	0.000			
Y	0.000	Linear G1		
Z	0.000			
Fast trav.	Linear	Circul.	Accept Insert	Accept Change

Fig. 5-7 Teach In of NC blocks

Fast
Trav.

Use this softkey to generate a rapid traverse block (G0).

Linear

Use this softkey to generate a linear feed block (G1).

Circul.

Use this softkey to generate a circular block (G5 with intermediate point and end point).

Accept
Insert

Use this softkey to generate a block with the values taught. The new block is inserted in front of the block to which the cursor is positioned.

Accept
Change

Values are corrected in the block (accepted from the screen form) to which the cursor is positioned. Use **RECALL** to return to the *Teach In* basic screen. Any amendments you wish to make can be later inserted manually.

Finish
Record

Use this softkey to generate an M2 block to be inserted after the current block (cursor position)

Progr.
run

Use this softkey to traverse the programmed block.

The machine screen set in Automatic mode appears again. Use **NC Start** to continue the selected but interrupted program from the block selected last (if the control system has not been in Reset state). Teach In remains enabled. Block search with NCK is not possible.

Teach In
Off

Use this softkey to turn off the submodule Teach In.

Note

After turning off Teach In, the **interrupted** program can no longer be edited.

Example

Teaching a G5 block

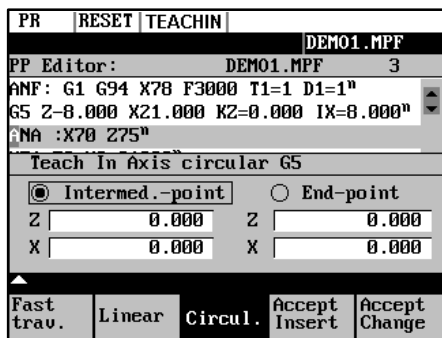


Fig. 5-8 Teach In of a circular block

- The program block with G5 is selected by the cursor.
- Press the softkey **Circul.**
The circle start point is the end point of the previous block.
- Approach to the intermediate point of the contour and press **Accept Change**.
- Approach to the end point of the contour and press **Accept Change**.

Part Programming

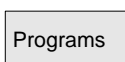
Functionality

This Section describes how to create a new part program.

The standard cycles can also be displayed provided you have the required access authorization.

Operating sequence

You are in the main menu.

 The *Programming* main screen appears.

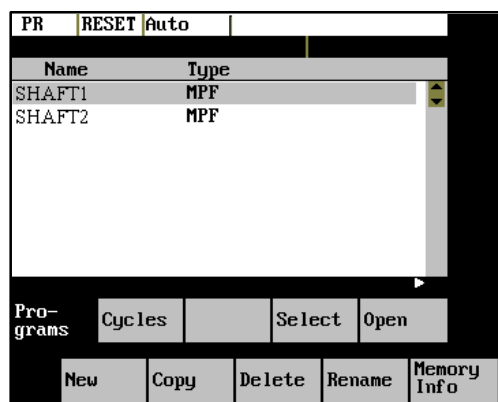

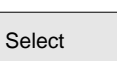


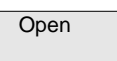
Fig. 6-1 *Programming* main screen


When the **Program** operating area is selected for the first time, the directory for part programs and subroutines is automatically selected (see above).

Softkeys

 The Standard Cycles directory is displayed by pressing the **Cycles** softkey. This softkey is only displayed if the operator has the appropriate access authorization.

 This function selects the program highlighted by the cursor for execution. The program is started on next **NC START**.

 Opens the files selected by the cursor for editing.

 Menu extension

New

Use the **New** softkey to create a new program. A window appears in which you are prompted to enter program name and type.

After you have confirmed your inputs by **OK**, the program editor is called, and you can enter part program blocks. Select **RECALL** to cancel this function.

Copy

Use the **Copy** softkey to copy the selected program into another program.

Delete

The program highlighted by the cursor is deleted after the system has requested confirmation of the delete operation.

Press **OK** to confirm the Delete request and **RECALL** to cancel it.

Rename

When you select the **Rename** softkey, a window appears in which you can rename the program that you have already highlighted by the cursor.

After you have entered the new name, confirm your rename request by **OK** or cancel by **RECALL**.

The **Programs** softkey can be used to change to the program directory.

Memory
Info

When you press this softkey, the totally available NC memory (in kbytes) is displayed.

6.1 Entering a new program - "Program" operating area

Functionality

This Section describes how to create a new file for a part program. A window appears in which you are prompted to enter program name and type.

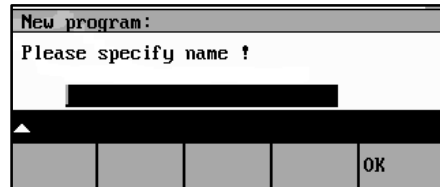


Fig. 6-2 New Program input screenform

Operating sequence

- Program

 You have selected the **Program** operating area. The Program Overview window showing the programs already stored in the CNC is displayed on the screen.
- New

 Press the **New** softkey. A dialog window appears in which you enter the new main program or subroutine program name. The extension .MPF for main programs is automatically entered. The extension .SPF for subroutines must be entered with the program name.
- A
Z

 Enter the new name.
- OK

 Complete your input by selecting the **OK** softkey. The new part program file is generated and is now ready for editing.
- ^

 The creation of the program can be interrupted by **RECALL**; the window is then closed.

6.2 Editing part programs - "Program" operating area

Functionality

Part programs or sections of a part program can only be edited if not being executed.

All modifications to the part program are stored immediately.

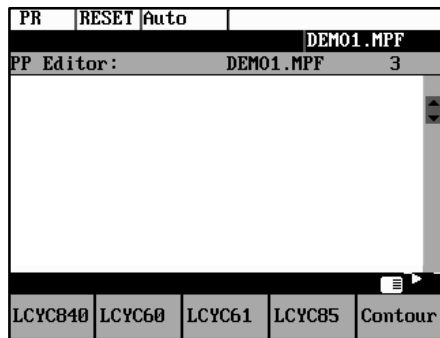



Fig. 6-3 Editor window

Operating sequence

- Programs

You are in the main menu and have selected the **Program** operating area. The program overview appears automatically.
- 

Use the **paging keys** to select the program you wish to edit.
- Open

Pressing the **Open** softkey calls the editor for the selected program and pulls down the editor window.

The file can now be edited. All changes are stored immediately.
- Select


Pressing the **Select** softkey selects the edited program for execution. This program is started with next **NC-Start**.

Softkeys

- User-assignable softkeys**

You can assign predefined functions to the softkeys 1 – 4 (see Section 6.3.4 "User-Assignable Softkeys").

The softkeys are assigned process-specific functions by the control manufacturer.
- Contour

The contour functions are described in Section 6.3 "Programming Support".
- 

Menu extension

Edit	
Mark	This function selects a section of text up to the current cursor position.
Delete	This function deletes the selected text.
Copy	This function copies the selected text to the clipboard.
Past	This function inserts a text from the clipboard at the current cursor position.
Recomp. cycles	For re-compilation, the cursor must stand on the cycle call line of the program. The required parameters must be arranged directly in front of the cycle call and may not be separated by instruction or comment lines. The function decodes the cycle name and prepares the screenform with the appropriate parameters. If there are any parameters outside the range of validity, the function automatically enters standard values. After the screenform has been quitted, the original parameter block is replaced by the corrected one. Note Only automatically generated blocks can be recompiled.

Note

To carry out these functions outside the Edit menu, it is also possible to use the key combinations <SHIFT> and

softkey 1	Mark
softkey 2	Delete block
softkey 3	Copy block
softkey 4	Insert block.



Menu extension

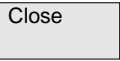
Assign SK	This function can be used to change the assignment of the softkey functions 1 – 4. For more detail description refer to Section 6.3.4.
--------------	---

Search	The softkeys Search and Contin. search can be used to search for a string chain in the program file displayed on the screen.
--------	--

Text	Type the text you wish to find in the input line and start the Search operation by selecting the OK softkey. If the character string you have specified cannot be found in the program file, an error message appears that must be acknowledged with OK . You can exit the dialog box without starting the search by selecting RECALL .
------	--

Line no.	Type the line number in the input line. The search is started by pressing OK . You can quit the dialog box without starting the search by selecting RECALL .
----------	--

Contin. Search	The functions searches through the file to find another character string that matches the target string.
-------------------	--



This function stores the changes in the file system and automatically closes the file.

Editing cyrillic letters

This function is only available if the Russian language option is selected.

Procedure

The control system offers a window for cyrillic letters to choose from. This is enabled/disabled using the Toggle key.

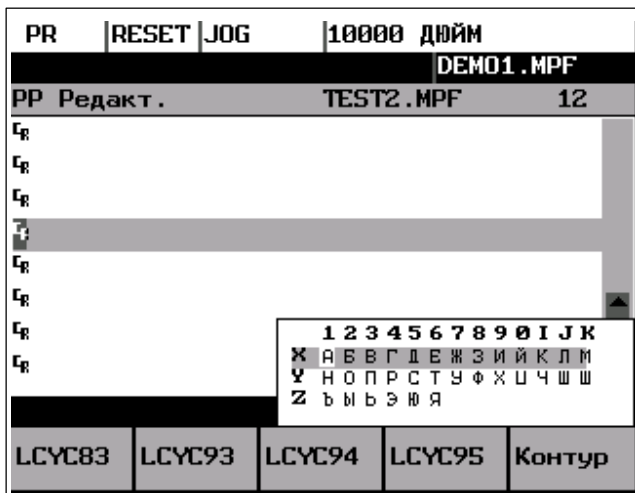


Fig. 6-4

To select a character,

- use the letters X, Y or Z to choose the line
- and then enter the digit or the letter assigned to the corresponding column.

When you enter the digit, the character will be copied into the edited file.

6.3 Programming support

Functionality

The programming support facility contains various help levels simplifying the programming of part programs without constraining your choice of inputs.

6.3.1 Vertical menu

Functionality

The vertical menu is displayed in the program editor.

The vertical menu allows you to quickly insert certain NC instructions into the part program.

Operating sequence

You are in the program editor.



Press the **VM** key and select the desired instruction from the list.

PR	RESET	Auto	
			DEMO1.MPF
PP Editor:			DEMO1.MPF 3
Paste:			Zyklus...
ANF: G1	1. LCYCL	call cycle	
ANA :X70	2. SIN	sin(x)	
N51 Z0 M	3. COS	cos(x)	
N60 X100	4. TAN	tan(x)	
N75 F850	5. SQRT	sqrt(x)	
N76 X0 Z	6. GOTOF <Label>	jump forward	
N80 GOT0	7. GOTOB <Label>	jumb backw.	
▲Select :			⇒

Fig. 6-5 Vertical menu

Lines that end in “...” contain a collection of NC instructions. You can list these instructions by pressing the **Input** key or entering the number of the line.

PR	RESET	Auto	
			LCYCB40
PP Editor:			DEMO1.MP LCYCB5
Paste:			LCYCB40 LCYCB0
ANF: G1	1. LCYCL	c LCYCB1	
ANA :X70	2. SIN	s	
N51 Z0 M	3. COS	c	
N60 X100	4. TAN	t	
N75 F850	5. SQRT	s	
N76 X0 Z	6. GOTOF <Label>	j	
N80 GOT0	7. GOTOB <Label>	jumb backw.	
▲Select :			⇒

Fig. 6-6 Vertical menu



Use the paging keys to browse through the list.



Confirm your entry by pressing **Input**.

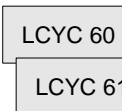
Alternatively, the number of the lines from 1 to 7 can be entered to select instructions and take them over into the part program.

6.3.2 Cycles

Functionality

You can either specify your own machining cycles on assigning parameters or, alternatively, use input forms in which you set all the necessary R parameters.

Operating sequences



The screenforms are selected either with the available softkey functions or by means of the vertical menu.

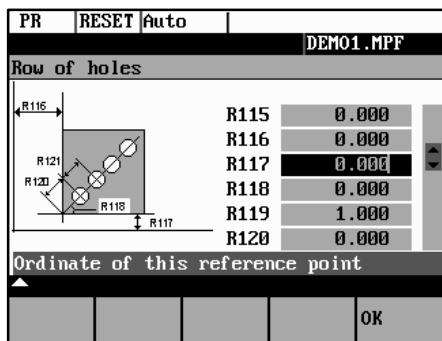
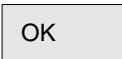


Fig. 6-7

The cycle support provides a screenform in which you can fill in all the necessary R parameters. A graphic and a context-sensitive help will assist you to fill in the form.



Select the **OK** softkey to transfer the generated cycle call to the part program.

6.3.3 Contour

Functionality

To create part programs quickly and reliably, the control system offers various contour screenforms. To use them, enter the appropriate parameters in the interactive screenforms.

The following contour elements and contour sections can be programmed using the contour screenforms:

- Straight section with specification of end point or angle
- Circle sector with specification of center point / end point / radius
- Contour sector 'straight line – straight line' with specification of angle and end point
- Contour section 'straight line – circle' with tangential transition; calculated from angle, radius and end point
- Contour section 'straight line – circle' with any transition; calculated from angle, radius and end point
- Contour section 'circle – straight line' with tangential transition; calculated from angle, radius and end point
- Contour section 'circle – straight line' with any transition; calculated from angle, radius and end point
- Contour section 'circle – circle' with tangential transition; calculated from angle, radius and end point
- Contour section 'circle – circle' with any transition; calculated from center point, radius and end point
- Contour section 'circle – circle' with any transition; calculated from center points and end point
- Contour section 'circle – straight line' with tangential transitions
- Contour section 'circle – circle – circle' with tangential transitions
- Contour section 'straight line – circle – straight line' with tangential transitions

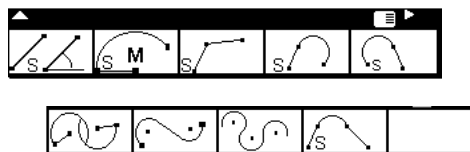


Fig. 6-8

Softkeys

The softkey functions branch into the contour elements.



Programming aid for programming straight line sections.

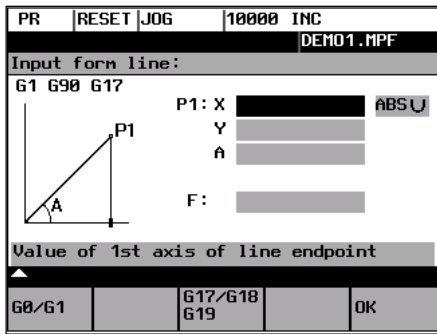


Fig. 6-9

Enter the end point of the straight line.

G0/G1

The block is traversed either at rapid traverse or with the programmed feedrate.

The end point can be entered either in the absolute dimension, as an incremental dimension (referred to the starting point) or in polar coordinates. The current setting is displayed in the interactive dialog screenform.

The end point can also be specified by a coordinate and the angle between the 1st axis and the straight line.

If the end point is determined using polar coordinates, the length of the vector between pole and end point is required, as well as the angle of the vector with reference to the pole. When using the possibility, first a pole must be set.

G17/18/19

Pressing this softkey selects the plane G17 (X–Y), G18 (Z–X) or G19 (Y–Z)

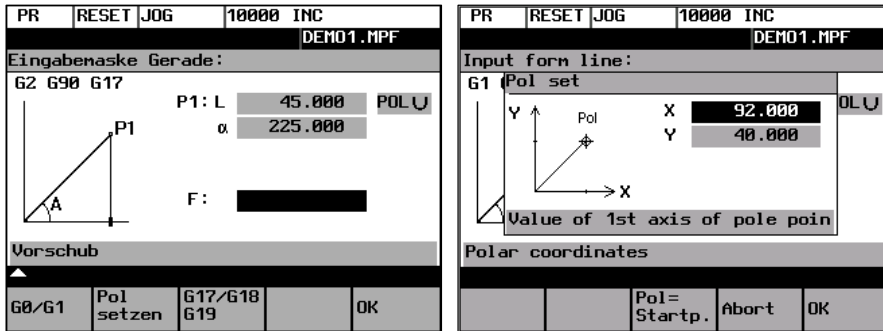


Fig. 6-10

Type the values in the input fields and close the screenform with **OK**.

OK

Pressing the **OK** softkey takes over the block into the part program and displays the *Additional Functions* form in which you can extend the block by adding more instructions.

Additional functions

PR	RESET	Jog	
Input form additional functions:			
G			
M			
S			
T			
D			
RND			
FASE			
▲			
			OK

Fig. 6-11 Additional Functions screenform

Enter additional commands in the fields. The commands can be separated by means of blanks, commas or semi-colons.

This screenform is available for all contour elements.



The **OK** softkey transfers the commands to the part program.

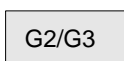
Select **RECALL** if you wish to exit the interactive form without saving the values.



The dialog screenform is used to create a circular block by means of the end and center point coordinates.

PR	RESET	JOG	10000	INC
DEMO1.MPF				
Form sector: Centre/end point				
G2 G90 G17	E: X		ABS	U
	Y			
	M: X		ABS	U
	Y			
	R			
F:				
Value of 1st axis of circle endpoint				
▲				
G2/G3		G17/G18		OK
		G19		

Fig. 6-12



This softkey changes the direction of rotation from G2 to G3. G3 appears on the display. When you press the softkey again, you will return to G2.



Use this softkey to select the plane G17 (X–Y), G18 (Z–X) or G19 (Y–Z).



Pressing the **OK** softkey takes over the block into the part program and displays the *Additional Functions* form in which you can extend the block by adding more instructions.



This function is intended to calculate the intersection point between two straight lines.

Specify the coordinates of the end point of the second straight line and the angles of the straight line. For the coordinate value, the toggle key can be used to choose between absolute, incremental or polar coordinates.

If the starting point cannot be selected based on the previous blocks, the operator must set the starting point.

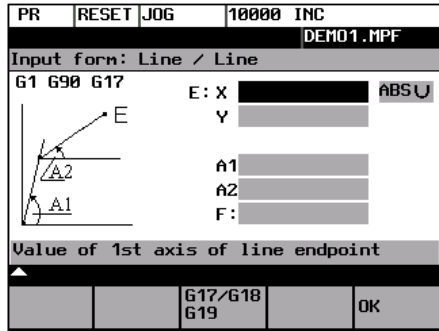


Fig. 6-13 Calculating the intersection point between two straight lines

Table 6-1 Input in the interactive screenform

End point of straight line 2	E	Specify the end point of the straight line depending on the selected plane (G17/18/19).
Angle of straight line 1	A1	The angle must be specified in the CCW direction in the range between 0 and 360 degrees.
Angle of straight line 2	A2	The angle must be specified in the CCW direction in the range between 0 and 360 degrees.
Feedrate	F	Feedrate
Plane		X-Y, Z-X, Y-Z



This function is used to calculate the tangential transition between a straight line and a circle sector. The straight line must be described by starting point and angle. The circle must be described by the radius and by the end point.

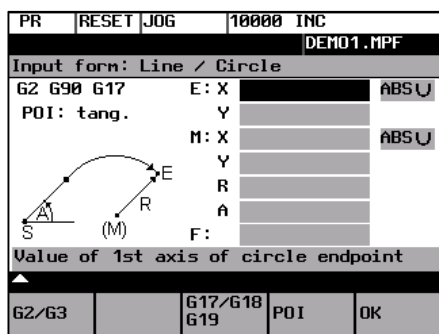


Fig. 6-14 Straight line – circle with tangential transition

Table 6-2 Input in the interactive screenform

Circle end point	E	The end point of the circle must be specified depending on the selected plane.
Straight line angle	A	The angle is specified in the CCW direction in the range between 0 and 360 degrees.
Circle radius	R	Input field for the circle radius
Feed	F	Input field for the interpolation feed.
Circle center point	M	If there is no tangential transition between the straight line and the circle, the circle center must be known. The circle center point is specified depending on the calculation method (absolute or incremental dimension / polar coordinates) selected in the previous block.

G2/G3

This softkey is used to switch the direction of rotation from G2 to G3. G3 is displayed on the screen. Pressing this softkey once more will switch the display back to G2.

Both the center and the end points can be acquired either in the absolute dimensions, in the incremental dimensions or as polar coordinates. The current setting is displayed in the interactive screenform.

G17/18/19

Selection of the plane G17 (X–Y), G18 (Z–X) or G19 (Y–Z)

POI

You can choose between tangential or any transition.

If the starting point cannot be determined from the previous blocks, the starting point must be set by the operator.

The screenform will generate a straight line and a circle block from the entered data.

If there are several intersection points, the desired intersection point must be selected by the operator in an interactive screenform.



This function is used to calculate the tangential transition between a circle sector and a straight line. The circle sector must be described by the parameters starting point and radius, and the straight line must be described by the parameters end point and angle.

PR	RESET	JOG	10000	INC
DEMO1.MPF				
Input form: Circle / Line				
G2	G90	G17	E: X	ABS U
POI: tang.			Y	
			M: X	ABS U
			Y	
			R	
			A	
			F:	
Value of 1st axis of line endpoint				
▲				
G2/G3		G17/G18 G19	POI	OK

Fig. 6-15 Tangential transition

Table 6-3 Input in the interactive screenform

Straight line end point	E	Enter the end point of the straight line depending on the selected plane (G17/18/19).
Center point	M	The center point of the circle must be entered either in absolute, incremental or polar coordinates.
Circle radius	R	Input field for the circle radius
Angle of straight line 1	A	The angle is specified in the CCW direction in the range between 0 and 360 degrees.
Feedrate	F	Input field for the interpolation feedrate

G2/G3 This softkey is used to switch the direction of rotation from G2 to G3. G3 is displayed on the screen. Pressing this softkey once more will switch back to G2; the display will change to G2.

The center point can be acquired either in absolute dimensions, incremental dimensions or polar coordinates. The current setting is displayed in the interactive screenform.

G17/18/19 Selection of the plane G17 (X–Y), G18 (Z–X) or G19 (Y–Z)

POI Use this softkey to choose between tangential or any transition.

If the starting point cannot be generated from the previous blocks, the starting point must be set by the operator.

The screenform will generate both a straight line and a circle block based on the entered data.

If there are several intersection points, the desired intersection point must be selected by the operator from a dialog box.



This function is used to calculate the tangential transition between two circle sectors. Circle sector 1 must be described by the parameters starting point and center point, and circle sector 2 must be described by the parameters end point and radius.

To avoid an overdetermination, input fields not needed are hidden.

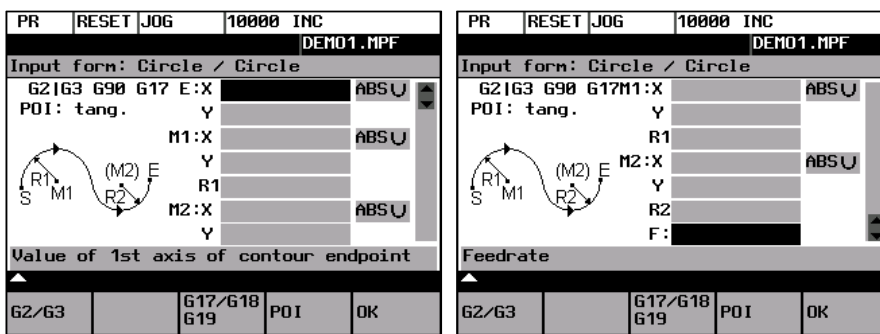


Fig. 6-16 Tangential transition

Table 6-4 Input in the interactive screenform

End point of circle 2	E	1st and 2nd geometry axes of the plane
Center point of circle 1	M1	1st and 2nd geometry axes of the plane
Radius of circle 1	R1	Input field for the circle radius
Center point of circle 2	M2	1st and 2nd geometry axes of the plane
Radius of circle 2	R2	Input field for the circle radius
Feedrate	F	Input field for the interpolation feedrate

G2/G3

This softkey is used to switch the direction of rotation from G2 to G3. G3 is displayed on the screen. Pressing this softkey once more will switch back to G2; the display will change to G2.

The center point can be acquired either in absolute dimensions, incremental dimensions or polar coordinates. The current setting is displayed in the interactive screenform.

G17/18/19

Selection of the plane G17 (X–Y), G18 (Z–X) or G19 (Y–Z)

POI

Use this softkey to choose between tangential or any transition.

If the starting point cannot be generated from the previous blocks, the starting point must be set by the operator.

The screenform will generate two circle blocks based on the entered data.

Selecting the intersection point

If there are several intersection points, the desired intersection point must be selected by the operator from a dialog box.

POI 1

The contour is drawn using intersection point 1.

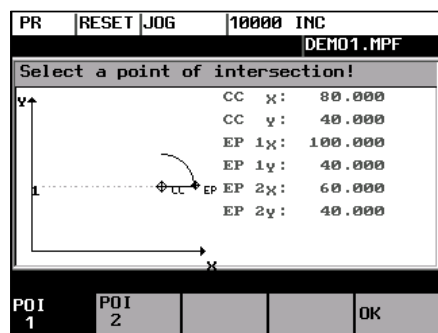


Fig. 6-17 selection of intersection point 1

POI 2

The contour is drawn using intersection point 2.

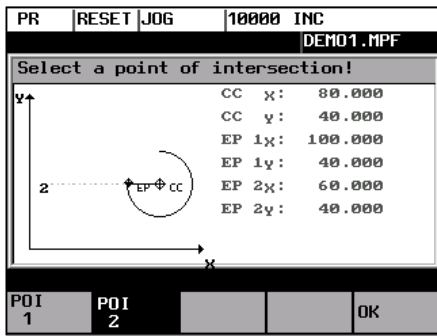
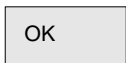


Fig. 6-18 Selection of intersection point 2



Pressing this softkey will accept the intersection point of the displayed contour into the part program.



This function is used to insert a straight line tangentially between two circle sectors. The sectors are determined by their center points and their radii. Depending on the selected direction of rotation, different tangential intersection points result.

Use the screenform, which will appear, to enter the parameters center point and radius for sector 1, as well as the parameters end point, center point and radius for sector 2. In addition, the direction of rotation must be selected for the circles. The current setting is displayed in a help screen.

The end and center points can be acquired either as absolute, incremental or polar coordinates.

The OK function will calculate three blocks from the given values and will insert them into the part program.

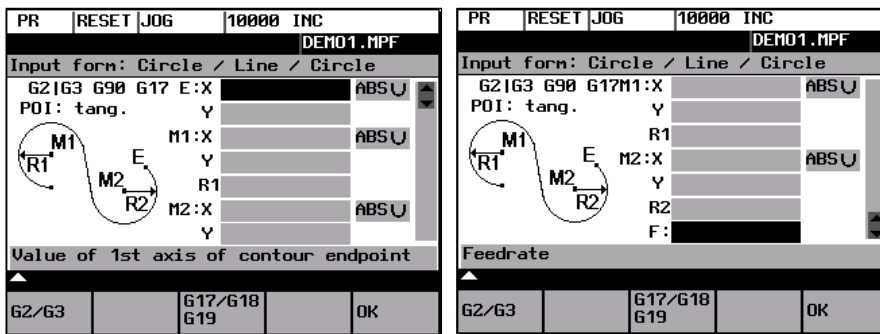


Fig. 6-19 Screenform for calculating the contour section 'circle – straight line – circle'

Table 6-5 Input in the interactive screenform

End point	E	1st and 2nd geometry axes of the plane If no coordinates are entered, the function will provide the intersection point between the inserted circle sector and sector 2.
Center point of circle 1	M1	1st and 2nd geometry axes
Radius of circle 1	R1	Input field for radius 1
Center point of circle 2	M2	1st and 2nd geometry axes of the plane

Table 6-5 Input in the interactive screenform, Fortsetzung

Radius of circle 2	R2	Input field for radius 2
Feedrate	F	Input field for the interpolation feedrate

If the starting point cannot be determined based on the previous blocks, the appropriate coordinates must be entered in the "Starting point" screenform.

The screenform will generate both a straight line and two circle blocks based on the entered data.

G2/G3

Use this softkey to define the direction of rotation of the two circle sectors. You can choose between

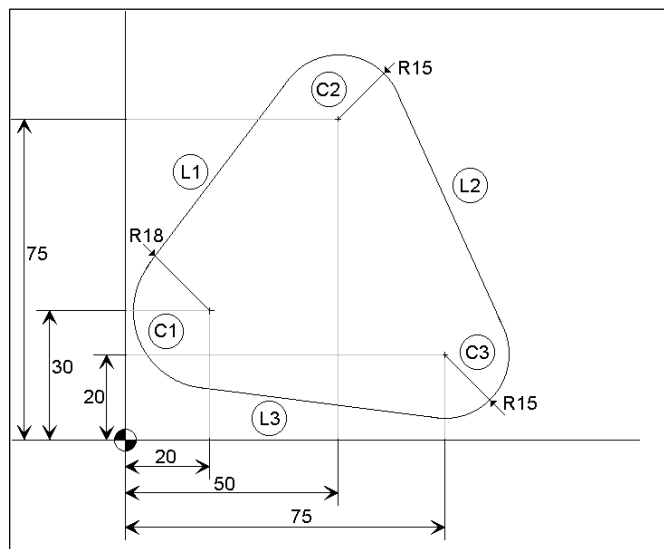
Sector 1	Sector 2
G2	G3,
G3	G2,
G2	G2 and
G3	G3

The end point and the center points can be acquired either in absolute, incremental or polar coordinates. The current setting is displayed in the interactive screenform.

G17/18/19

Selection of the plane G17 (X-Y), G18 (Z-X) or G19 (Y-Z)

Example



6.3 Programming support

Given: R1 18 mm
 R2 15 mm
 R3 15 mm
 M1 X 20 Y 30
 M2 X 50 Y 75
 M3 X 75 Y 20

Starting point: The point X = 2 and Y = 30 mm is supposed as the starting point.

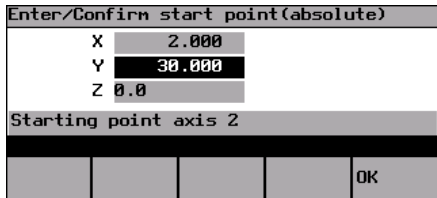



Fig. 6-20 Setting the starting point

After the starting point has been confirmed, the  screenform can be used to calculate the contour section (C1) – (L1) – (C2).

Use softkey 1 to set the direction of rotation of the two circle sectors and to fill out the parameter list.

The end point can be left open or the points X 50 Y 90 (75 + R 15) must be entered.

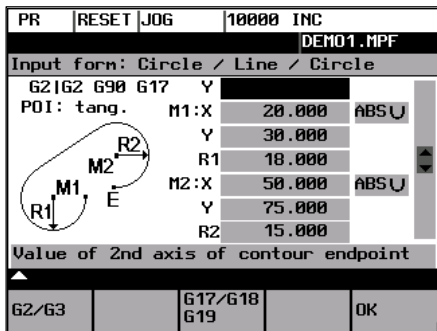


Fig. 6-21 Calling the screenform

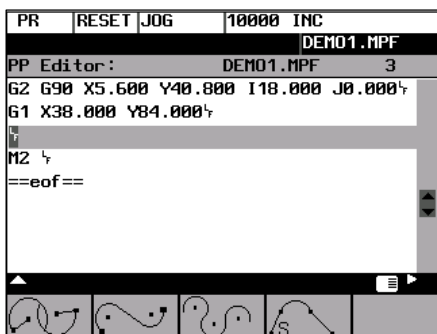


Fig. 6-22 Result of step 1

After you have filled out the screenform, press OK to quit the screenform. The intersection points are calculated and the two blocks are generated.

Since the end point has been left open, the intersection point between the straight line ^(L1) and the circle sector ^(C2) is also the starting point for the subsequent contour definition.

Now, call the screenform for calculating the contour section ^(C2) – ^(C3) again.

PR	RESET	JOG	10000	INC
DEMO1.MPF				
Input form: Circle / Line / Circle				
G2	G90	G17	Y	
POI: tang.				
M1:X	50.000	ABS	U	
Y	75.000			
R1	15.000			
M2:X	75.000	ABS	U	
Y	20.000			
R2	15.000			
Value of 2nd axis of contour endpoint				
G2/G3		G17/G18	G19	OK

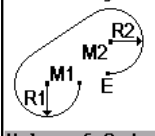


Fig. 6-23 Calling the screenform

PR	RESET	JOG	10000	INC
DEMO1.MPF				
PP Editor: DEMO1.MPF 5				
G2 G90 X5.600 Y40.800 I18.000 J0.000				
G1 X38.000 Y84.000				
G2 G90 X63.655 Y81.207 I12.000 J-9.000				
G1 X88.655 Y26.207				
M2				
==eof==				




Fig. 6-24 Result of step 2

The end point of step 2 is the intersection point of the straight line ^(L2) with the circle sector ^(C3). Then, calculate the contour section starting point 2 – circle sector ^(C1).

PR	RESET	JOG	10000	INC
DEMO1.MPF				
Input form: Circle / Line / Circle				
G2	G90	G17	Y	
POI: tang.				
M1:X	75.000	ABS	U	
Y	20.000			
R1	15.000			
M2:X	20.000	ABS	U	
Y	30.000			
R2	18.000			
Value of 2nd axis of contour endpoint				
G2/G3		G17/G18	G19	OK

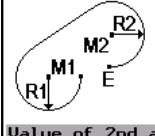


Fig. 6-25 Calling the screenform

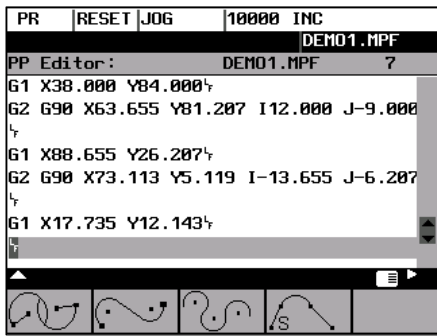


Fig. 6-26 Result of step 3

Then, link the new end point with the starting point. To do so, use the  function.

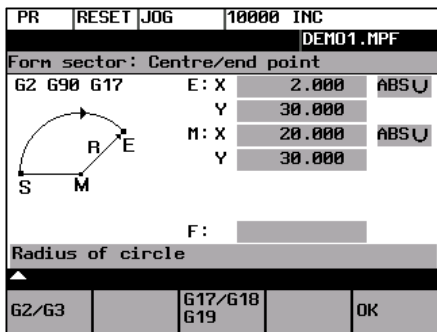


Fig. 6-27 Step 4

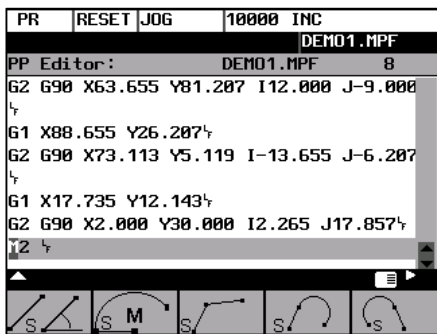


Fig. 6-28 Result of step 4



This function is used to insert a circle sector tangentially between two adjacent circle sectors. The circle sectors are described by their center points and their circle radii. The inserted sector is described by its radius.

Use the screenform to enter the parameters center point and radius for circle sector 1, and the parameters end point, center point and radius for circle sector 2. In addition, the radius for the inserted circle sector 3 must be entered and the direction of rotation be defined.

The end point and the center points can be acquired either as absolute, incremental or polar coordinates.

The selected setting is displayed in a help screen.

The **OK** function will calculate three blocks from the given values and will insert them into the part program.

Fig. 6-29 Screenform for calculating the contour section 'circle – circle – circle'

Table 6-6 Input in the dialog screenform

End point	E	1st and 2nd geometry axes of the plane If no coordinates are entered, the function provides the intersection point between the inserted circle sector and sector 2.
Center point of circle 1	M1	1st and 2nd geometry axes of the plane
Radius of circle 1	R1	Input field for radius 1
Center point of circle 2	M2	1st and 2nd geometry axes of the plane
Radius of circle 2	R2	Input field for radius 2
Radius of circle 3	R3	Input field for radius 3
Feed	F	Input field for the interpolation feed

If the starting point cannot be determined from the previous blocks, the respective coordinates must be entered in the "Starting point" screenform.

G2/G3

This softkey defines the direction of rotation of the three circles. It is possible to select between:

Sector 1	Inserted Sector	Sector 2
G2	G 3	G2,
G2	G2	G2,
G2	G2	G3,
G2	G3	G3,
G3	G2	G2,
G3	G3	G2,
G3	G2	G3,
G3	G3	G3

G17/18/19

Selection of the plane G17 (X–Y), G18 (Z–X) or G19 (Y–Z)

Example

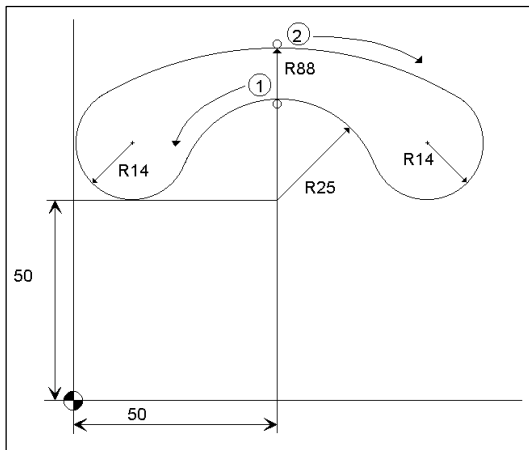



Fig. 6-30 Example

Given: R1 88 mm
 R2 25 mm
 R3 14 mm
 M1 X 50 Y 0
 M2 X 50 Y 50

The coordinates X 50, Y 75 (50 + R2) will be selected as the starting point.

After you have confirmed the starting point, use the  screenform to calculate the contour section ^① (circle sector R2 – circle sector R1). The coordinates X50, Y 88 (R1) constitute the end point for this contour section ^①.

Enter/Confirm start point(absolute)	
X	50.000
Y	75.000
Z	0.0
Starting point axis Z	
OK	

Fig. 6-31 Setting the starting point

PR	RESET	JOG	10000	INC
DEMO1.MPF				
Input form: Circle / Circle / Circle				
G3 I62 I62 G90 G17				
POI: tan				
E:X		50.000		ABS U
Y		88.000		
M1:X		50.000		ABS U
Y		50.000		
M2:X		50.000		ABS U
Y		50.000		
Value of 1st axis of contour endpoint				
G2/G3		G17/G18		OK
		G19		

PR	RESET	JOG	10000	INC
DEMO1.MPF				
Input form: Circle / Circle / Circle				
G3 I62 I62 G90 G17				
POI: tan				
Y		50.000		
M2:X		50.000		ABS U
Y		0.000		
R3		14.000		
Radius R3 circle 3				
G2/G3		G17/G18		OK
		G19		

Fig. 6-32 Calling the screenform 'circle – circle – circle'



PR	RESET	Jog	10000	INC
DEMO1.MPF				
PP Editor: TEST2.MPF 28				
"				
G3 G90 X26.80499 Y59.32692 I0.0 J-25.0				
"				
G2 X6.97012 Y76.76216 I-12.98921 J5.22				
300"				
G2 X50.0 Y88.0 I43.02988 J-76.76216"				
"				
"				
				

Fig. 6-33 Result of step 1

In the second step, screenform  is used to calculate the contour section ^② (circle sector R1 – circle sector R2). For calculation, select direction of rotation G2 – G2 – G3. Since the end point of step 1 is at the same time the starting point for step 2, no new starting point needs to be set. For step 2, the coordinates X 50 Y 75 (50 + R2) constitute the end point. The contour is thus closed.

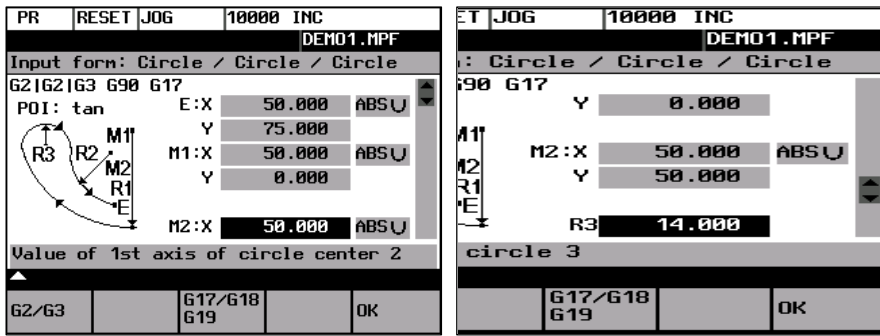


Fig. 6-34 Calling the screenform 'circle – circle – circle'

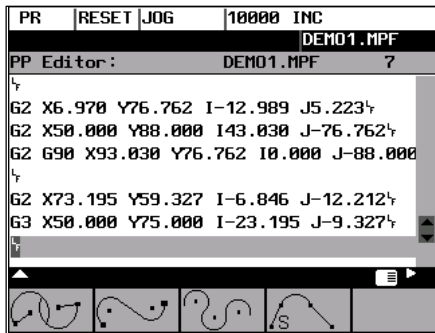


Fig. 6-35 Result of step 2



This function is used to insert a circle sector (with tangential transitions) between two straight lines. The circle sector is described by the center point and the radius. The coordinates of the end point of the second straight line and, optionally, angle A2. The first straight line is described by the starting point and the angle A1.

If the starting point cannot be determined from the previous blocks, the starting point must be set by the operator.

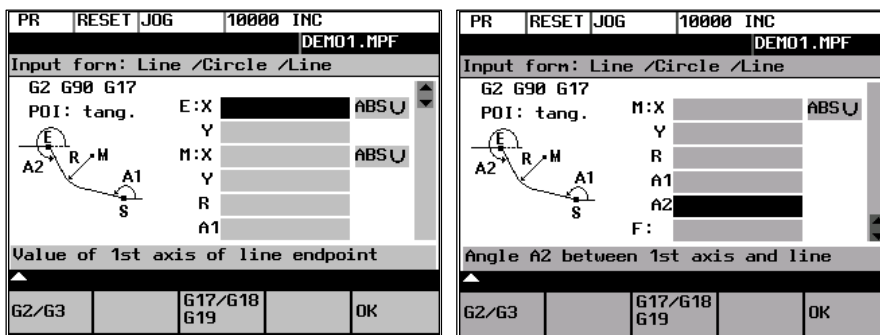


Fig. 6-36 Straight line – circle – straight line

Table 6-7 Input in the interactive screenform

End point of straight line 2	E	Enter the end point of the straight line.
Circle center point	M	1st and 2nd axes of the plane
Angle of straight line 1	A1	The angle must be specified in the CCW direction.
Angle of straight line 2	A2	The angle must be specified in the CCW direction.
Feedrate	F	Input field for the feedrate

End and center points can be specified either in absolute, incremental or polar coordinates. The screenform will generate a circle and two straight line blocks from the entered data.

G2/G3

Use this softkey to switch the direction of rotation from G2 to G3. G3 is displayed on the screen. Pressing this softkey once more will switch back to G2; the display will change to G2.

G17/18/19

Selection of the plane G17 (X–Y), G18 (Z–X) or G19 (Y–Z)

6.3.4 Free softkey assignment

Assign
SK

You can assign the softkeys various cycles or contours. To this aim, the softkeys 1 to 4 in the softkey bar in the Program operating area are provided.

Once you have activated the **Assign softkeys** function, a list of all available cycles or contours appears on the screen.

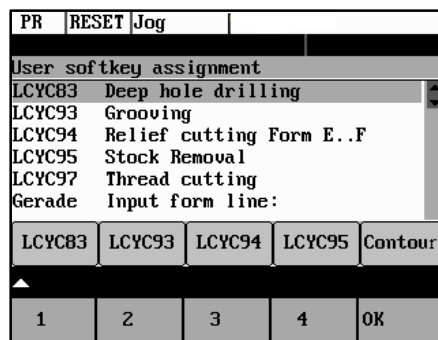


Fig. 6-37

Position the **cursor** on the element you wish to assign.

Press the desired softkey from **1** to **4** to assign them the desired element. The assignment you have made appears in the softkey bar under the selection list.

OK

Confirm the assignment you have made by selecting the **OK** softkey.

Services and Diagnosis

7.1 Data transfer via the V24 Interface

Functionality

You can use the V.24 (RS 232) interface of the CNC to output data (e.g. part programs) to an external data storage medium or to read in them from there. The V.24 interface and the data storage device must be matched to one another. The control system provides an interactive screenform in which you can set the special data for your storage medium.

After you have selected the **Services** operating area, a list of all available part programs and subroutines appears on the screen.



Fig. 7-1 Services main screen

File types

Provided the access authorization is set, files can be read in or read out via the V24 interface.

If the access authorization is set (cf. Technical Manual), the following data can be transmitted:

- **Data**
 - Option data
 - Machine data
 - Setting data
 - Tool offsets
 - Zero offsets
 - R parameters

7.1 Data transfer via the V24 Interface

- **Part programs**
 - Part programs
 - Subroutines
- **Start-up data**
 - NCK data
 - PLC data
 - Alarm texts
- **Compensation data**
 - Leadscrew pitch/encoder errors
- **Cycles**
 - Standard cycles

Operating sequence



Use the **Service** softkey to select the **Services** operating area.

Service

Softkeys

Data In
Start

This key starts reading in data.

DataOut
Start

This key starts reading out data to the PG/PC or another device.

RS232
setting

With the access authorization set, this function can be used to modify the interface parameters and to save them.

SU	RUN	Auto	
Setting values:		RS232 user	
Parameter		spec. funct.	
Device	BTS CTS	Start with XON	NU
Baud rate	9600	Conf. Overw.	NU
Stop bits	1	End block w. CR	YU
Parity	None	Stop with EOF	YU
Data bits	8	Eval DSR	NU
XON (Hex)	11	Leader/Trailer	NU
XOFF (Hex)	13	Tape format	YU
End of Tran	a	Time monitor.	NU
RS232 user	RS232 printer		OK

Fig. 7-2 Interface settings

Position the cursor on the desired data.

Use the selection key to modify the settings in the left column. The special functions can be activated and deactivated by the **Select** key.

Activating the transmission log

These softkeys are intended to adapt the V.24 interface to the transmission log. 2 logs are set by default.

RS232
text

Use this softkey to produce a log for the transfer of data, part programs and cycles.

RS232
binary

Use this softkey to produce a log for the transfer of start-up data.
The baud rate can be adapted according to the receiver.

OK

Press this softkey to save your settings.

Select **RECALL** to exit the window without saving your settings.

Error
log

A log is output for the transferred data.

- For files to be output, it contains
 - the file name and
 - an error acknowledgement
- For imported files, it contains
 - the file name and the path specification
 - an error acknowledgement

Transmission messages:

OK	Transmission completed successfully
ERR EOF	End-of-file character received, but the archive file is not complete.
Time Out	Timeout monitoring is signaling an interruption in the transmission.
User Abort	Transmission aborted by Stop softkey
Error Com	Error at COM 1
NC / PLC Error	NC error message
Error Data	Data errors 1. Files read in with/without leader or 2. Files transferred in tape format without file name
Error File Name	The file name does not comply with NC name conventions.
no access right	No access right for this function

Show

Display of the data that are amongst the data types marked with "...". Use this function to transfer individual files.



Menu extension

Execut
f. ext.

An external program is transferred to the control system via the V.24 interface and executed immediately by pressing **NC Start** (see Section 5.5).

Note

As an alternative, **program execution from external** can also be activated in the **Automatic** area.

7.1.1 Interface parameters

Table 7-1 Interface parameters

Parameter	Description
Device type	<ul style="list-style-type: none"> • XON/XOFF One possible method of controlling the transmission operation is to use the XON (DC1, DEVICE CONTROL 1) and XOFF (DEVICE CONTROL 2) control characters. As soon as the buffer of the I/O device is full, it returns XOFF, and as soon as it can receive data again, it sends XON. • RTS/CTS The RTS signal (Request to Send) controls the send operation of the data transmission device. Active signal: Send data Passive signal: Do not exit send mode until all transferred data have been sent. The CTS signal is the acknowledgment signal for RTS and indicates that the data transmission device is ready to send.
XON	This is the character that is used to start transmission. It is effective only for device type XON/XOFF.
XOFF	This is the character with which data transmission is stopped.
End of transmission	This is the character that signals end of transmission of a text file. The special function "Stop with end of transmission" character may not be active if binary data are to be transferred.
Baud rate	Interface speed settings 300 baud 600 baud 1200 baud 2400 baud 4800 baud 9600 baud
Data bits	Number of data bits for asynchronous transmission. Input: 7 data bits 8 data bits (default)
Stop bits	Number of stop bits for asynchronous transmission. Input: 1 stop bit (default) 2 stop bits
Parity	Parity bits are used to detect errors. These are added to the coded character in order to obtain either an even or odd number of positions set to "1" . Input: No parity (default) Even parity Odd parity

7.1.2 Special functions

Table 7-2 Special functions

Function	Active	Inactive
Start with XON	Transmission starts if the transmitter receives an XON character in the data flow.	Transmission starts independently of any XON character.
Overwrite with confirmation	When a file is imported, a check is made for an existing file of the same name in the NC.	The files are overwritten without confirmation request.
End of block with CR LF	CR characters (hexadecimal 0D) are inserted with tape format outputs.	No additional characters are inserted.
Stop at end of transmission	The end-of-transmission character is active.	The character is not evaluated.
Evaluate DSR signal	Transmission is interrupted if the DSR signal is missing.	DSR signal has no effect.
Leader and trailer	Leader is skipped when data are received. A leader with 120 * 0 h is generated when data are output.	Leader and trailer are read in with other data. No leader is generated when data are output.
Tape format	Import of part programs	Import of archives in the SINUMERIK archive format
Timeout monitoring	Transmission is interrupted after 5 seconds in case of transmission problems.	No abortion of transmission

7.1.3 Interface parameterization

Please find examples for setting the V24 interface below.

Start-up data

Settings for transferring archives with the start-up data

SV RESET JOG 10000 INC			
Setting values: RS232 binary			
Parameter		spec. funct.	
Device	RTS CTS	Start with XON	NU
Baud rate	19200	Conf. Overw.	NU
Stop bits	1	End block w.CR	NU
Parity	None	Stop with EOF	NU
Data bits	8	Eval DSR	NU
XON (Hex)	11	Leader/Trailer	NU
XOFF (Hex)	13	Tape format	NU
End of Trans	1a	Time monitor.	NU
RS232 text	RS232 binary		OK

Fig. 7-3

7.1 Data transfer via the V24 Interface

Punched-tape input / output

If a punched-tape reader/puncher is connected, check the "Leader/Trailer" box.

If the punched-tape reader is controlled via CTS, then check the "Stop at end of transmission" box.

Device type: RTS/CTS
 XON: 0
 XOFF: 0
 End of transm.: 0
 Baud rate: 9600 baud
 Data bits: 8
 Stop bits: 2
 Parity: No parity
 Start with XON
 Overwrite with confirmation
 X Ends of block with CR LF
 Stop at transmission end
 X Evaluate DSR signal
 Leader and trailer
 X Tape format
 X Timeout monitoring

Parameters for a serial printer

A printer with a serial interface is connected via an appropriate cable (cable check at CTS).

Device type: RTS/CTS
 XON: 11(H)
 XOFF: 13(H)
 End of transm.: 1A(H)
 Baud rate: 9600 baud
 Data bits: 8
 Stop bits: 1
 Parity: No parity
 Start with XON
 Overwrite with confirmation
 X End of block with CR LF
 Stop at transmission end
 X Evaluate DSR signal
 X Leader and trailer
 X Tape format
 X Timeout monitoring

7.2 Diagnosis and start-up - "Diagnostics" operating area

Functionality

In the "Diagnostics" operating area, you can call service and diagnostic functions, set start-up switches, etc.

Operating sequence

Diagnostics

Selecting the **Diagnostics** softkey will open the *Diagnostics* main screen.

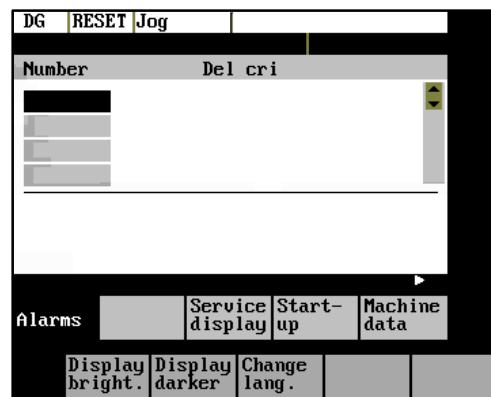






Fig. 7-4 *Diagnostics* main screen

Softkeys for diagnostic functions

Alarms

This window displays all pending alarms line by line, starting with the alarm with the highest priority. Alarm number, cancel criterion and error text are displayed. The error text refers to the alarm number on which the cursor is positioned.

Explanations with regard to the screenform above:

- Number
 - The "Number" item displays the alarm number. The alarms are displayed in chronological sequence.
- Cancel criterion
 - The symbol of the key required to reset the alarm is displayed for every alarm.
 -  Switch the device off and on again.
 -  Press the **RESET** key.
 -  Press the "Acknowledge alarm" key.
 -  Alarm is reset by **NC START**.
- Text
 - The alarm text is displayed.

7.2 Diagnosis and start-up - "Diagnostics" operating area

Service display The *Service Axes* window appears on the screen.

Service axes The window displays information about the axis drive.

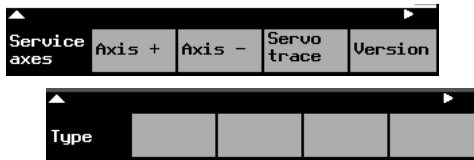


Fig. 7-5 The "Service Axes" window

In addition, the **Axis+** and **Axis-** softkeys are displayed. They can be used to call the values for the next or previous axis.

Servo trace To optimize the drives, an oscillograph function is provided for graphical representation of the velocity setpoint. The velocity setpoint corresponds to the $\pm 10V$ interface.

The start of recording can be linked with various criteria which permit recording in parallel to internal conditions of the control system. The setting needed for this option must be carried out in the "Select Signal" function.

The following functions can be used to analyze the result:

- Change scaling of abscissa and ordinate,
- Measure value by means of a horizontal or vertical marker,
- Measure the abscissa and ordinate values as a difference between two marker positions.

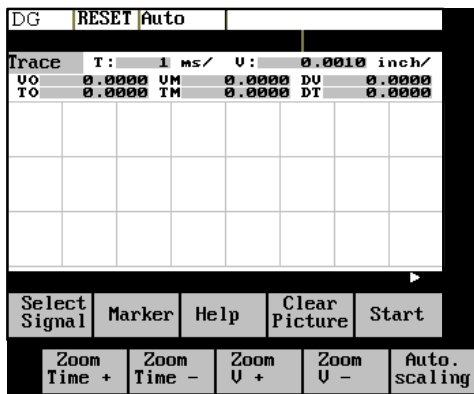


Fig. 7-6 The "Servo Trace" main screen

The heading of the diagram contains the current graduation of abscissa and ordinate, the current measured positions and the difference values of the markers.

The displayed diagram can be moved within the visible screen area by means of the cursor keys.

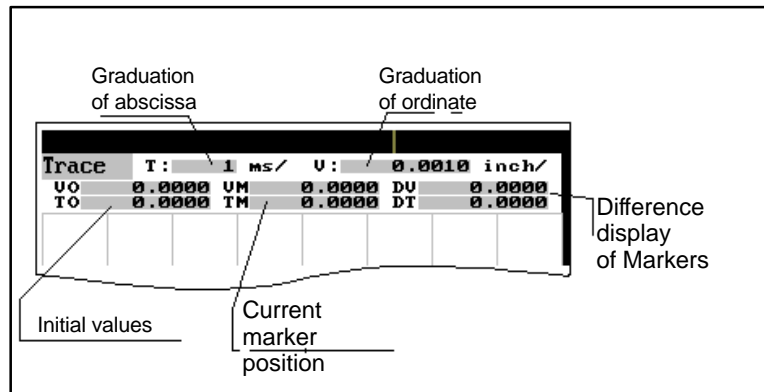


Fig. 7-7 Meaning of the fields

Select signal

Use this menu to select the axis to be measured, the measuring time, threshold value, pre-trigger/post-trigger time and trigger conditions. The signal settings are fixed.

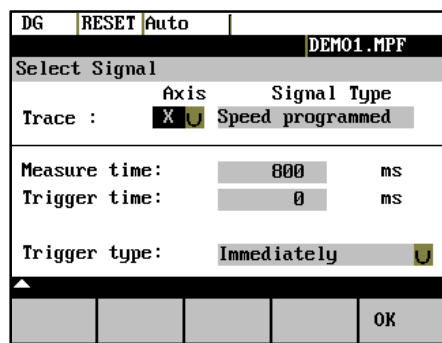


Fig. 7-8 Signal selection

- **Selecting the axis:** The axis is selected in the Axis toggle field.
- **Signal type:**
 - Velocity setpoint
 - Actual position value of measuring system 1
 - Following error
- **Determining the measuring time:** The measuring time is entered in ms directly in the "Measuring Time" input field.
- **Determining trigger time to or after**

With input values < 0, recording starts by the set time prior to the trigger event, and with values > 0 accordingly after the trigger event, whereby the following conditions must be observed:

$$\text{Trigger time} + \text{measuring time} \geq 0.$$
- **Selecting the trigger condition:** Position the cursor on the Trigger Condition field and select the condition using the toggle key.
 - No trigger, i.e. the measuring starts immediately after pressing the Start softkey.
 - Negative edge
 - Exact stop fine reached
 - Exact stop coarse reached

7.2 Diagnosis and start-up - "Diagnostics" operating area

- **Determining the trigger threshold:** The threshold is entered directly in the Threshold input field. It acts only for the trigger conditions "Positive edge" and "Negative edge".

Marker

This function branches to another softkey level, in which the horizontal or vertical marker can be switched on or off. The markers are displayed in the status bar.

The markers are moved in steps of one increment by means of the cursor keys. Larger step widths can be set in the input fields. The value specifies the number of raster units per **<SHIFT> + cursor movement** by which the marker is to be moved.

If a marker reaches the margin of the diagram, the next raster in horizontal or vertical direction is automatically pulled down.

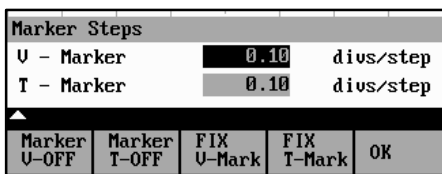


Fig. 7-9 Setting the markers

The markers can also be used to determine the differences in the horizontal or vertical direction. To this aim, position the marker on the start point and press either the Fix H – Mark. or the Fix T – Mark. softkey. The difference between the start point and the current marker position is now displayed in the status bar. The softkey labeling changes to "Free H – Mark." or "Free T – Mark."

Help

Pressing this functions calls explanations with regard to the displayed values on the screen.

Start

Pressing the **Start** softkey starts recording. The softkey labeling changes to **Stop**. The note "Recording active" is displayed.

When the measuring time is elapsed, the softkey labeling changes to **Start**.

Stop

Pressing the **Stop** softkey aborts the current measuring. The softkey labeling changes to **Start**.

Zoom Time +

The scaling changes in the following steps:
1, 2, 5, 10, 20, 50, 100, 200, 500, 1,000 ms/div.

Zoom Time -

Zoom V +

The horizontal scaling changes in the following steps:
0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1,000, 5,000 unit/ div

Zoom V -

Auto. scaling

This function calculates the vertical scaling from the peak values.

Version

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

Type

displays the control type

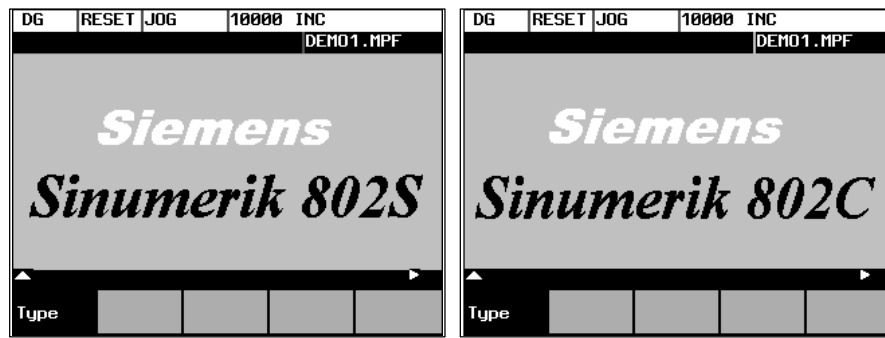


Fig. 7-10 Control type

Softkeys for start-up functions

Note

See also Technical Manual

Start-up

The start-up function branches to the following softkey functions:

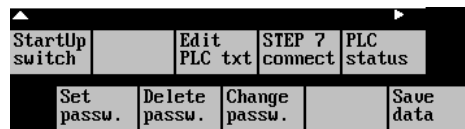


Fig. 7-11

Start-up switch

Start-up switch

You can assign the system power-up parameters various parameters.



Caution

Changes in the start-up branch have a considerable influence on the machine.

NC

Selecting the power-up mode of the NC.

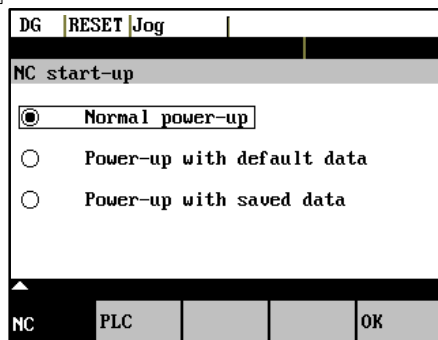


Fig. 7-12 NC start-up

PLC

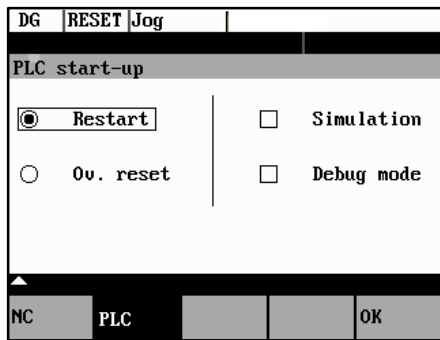


Fig. 7-13 PLC start-up

The PLC can be started in the following modes:

- Restart
- General reset

In addition, it is possible to link the selected mode with

- subsequent simulation or
- subsequent debugging mode.

OK

Use the **OK** key to start the NC start-up.

Return to the Start-up main screen without further action by **RECALL**.

Edit PLC txt

This function can be used to insert or modify PLC alarm messages. Select the desired alarm number using the softkey function "Next Number". The text currently valid is displayed in the window and in the input line.

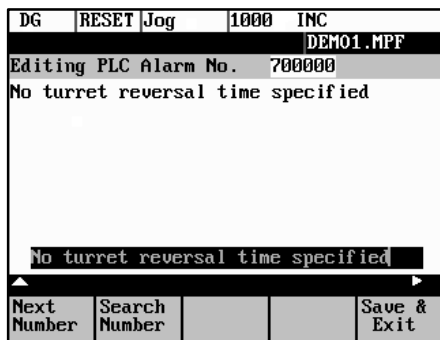


Fig. 7-14 Screenform for editing a PLC alarm text

Enter the new text in the input line. Complete your input by pressing **INPUT**.

For the notation of the texts refer to the Start-up Guide.

Next Number

This function selects the next following text number for editing. When the last text number is reached, the process restarts with the first number.

Search Number

This function selects the entered number for editing.

Save & Exit Pressing this function saves the modified texts. The editor is then quitted.

Recall The editor is quitted without saving the changes.

Editing Chinese characters

This function is only available if a Chinese character set is loaded.

The editor shows a section of Chinese characters. Use the cursor to navigate in the list. If the character you are looking for is not contained in the section, another section can be selected using the letters A – Z. Pressing softkey 4 takes over the desired character to the input line. In this mode, Latin letters cannot be entered.

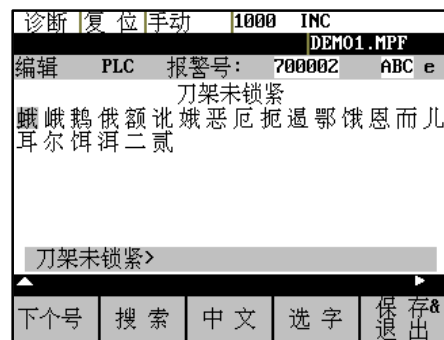


Fig. 7-15 screenform for editing a PLC alarm text in Chinese

The following softkey functions are realized:

Next Number This function selects the next following text number for editing. When the last text number is reached, the process restarts with the first number.

Search Number This function selects the entered number for editing.

Change Mode This function toggles between the selection of the section and the input of Latin letters.

Choose Char Pressing this softkey accepts the selected character into the input line.

Save & Exit Pressing this softkey saves the modified texts. The editor is then quitted.

Recall The editor is quitted without saving the changes.

STEP 7 connect The S7-Conn menu can be used to link the PLC with the external programming package S7-200. If the RS232 interface is already occupied by the data transfer, you can link the control system with the programming package only when the transmission is completed.

7.2 Diagnosis and start-up - "Diagnostics" operating area

When the link is activated, the RS232 interface is initialized. The following interface parameters are defined by the used program:

Device RTS – CTS
 Baud rate 38400
 Stop bits 1
 Parity even
 Data bits 8

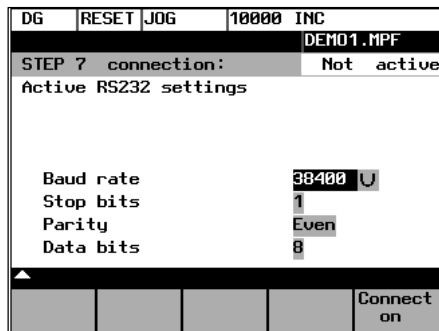


Fig. 7-16 S7-200 connection

Conn.
on
Conn.
off

This function activates the connection between the PC and the control system. The softkey labeling changes to Connection off (**Conn. off**).

The active or inactive condition, respectively, is maintained even if Power On is carried out (except for booting with default data).

Press **RECALL** to quit the menu.

PLC
status

You can display information about the current states of PLC memory cells listed below; if desired they can be altered.

It is possible to display 6 operands simultaneously.

Inputs	I	Input byte (IBx), input word (Iwx), input double word (IDx)
Outputs	Q	Output byte (Qbx), output word (Qwx), output double word (QDx)
Bit memories	M	Memory byte (Mx), memory word (Mw), memory double word (MDx)
Timers	T	Timer (Tx)
Counters	C	Counter (Zx)
Data	V	Data byte (Vbx), data word (Vwx), data double word (VDx)
Format	B H D	Binary Hexadecimal Decimal Binary representation cannot be used for double words. Counters and timers are displayed in decimal format.

DG	RESET	Jog	
PLC status display			active
Operand	Format	Value	
VB81000000	B	0000 0000	
change			delete

Fig. 7-17 PLC status display

There are further softkeys provided under this menu item.

- Edit
Cyclic updating of the values is interrupted. You can then edit the operand values.
- Cancel
Cyclic updating continues without the entered values being transferred to the PLC.
- Accept
The entered values are transferred to the PLC; cyclic updating continues.
- Delete
All operands are deleted.
- Operand +
The address of the operand can be incremented in steps of 1.
- Operand -
The address of the operand can be decremented in steps of 1.

Set
password

Set password

There are four different password levels implemented by the control system, thereby allowing four different levels of access authorization:

- Siemens password
- System password
- Manufacturer password
- User password

You can edit the data depending on your level of access authorization (refer also to the Technical Manual)



Enter the password.

If you do not know the password, you will not be granted access.

The password is set when you press the **OK** softkey.

You can return to the Start-up main screen without saving your input by selecting **RECALL**.

Delete password

The access authorization is reset.

Change password

Change password

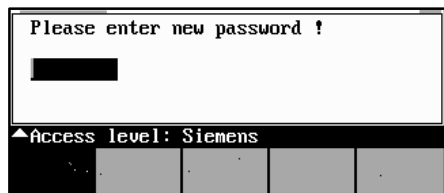


Fig. 7-18

Depending on the access authorization, various options for changing the password are provided in the softkey bar.

Use the softkeys to select the password level. Enter the new password and complete your input with **OK**.

The system asks you to confirm the new password again.

Press **OK** to complete the password change.

You can return to the Start-up main screen without saving your input by **RECALL**.

Save data

Save data

This function saves the contents of the volatile memory to a non-volatile memory area. Prerequisite: No program is currently being run.

It is not allowed to perform any operating actions while saving data.

Softkeys for service functions

Machine data

Machine data (see also Technical Manual)

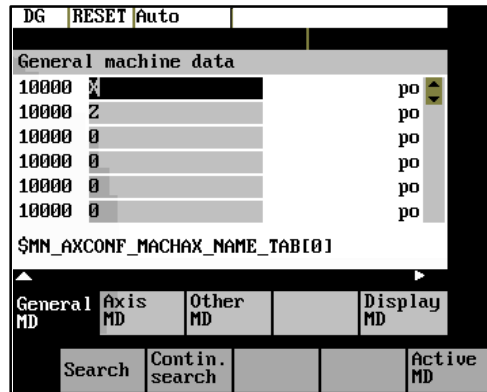


Fig. 7-19

Changes to the machine data have a considerable influence on the machine. Incorrect parameter settings can result in irreparable damage to mechanical components.

Units	userdef	User-defined
	m/s**2	Meters per second
	U/s**3	Revolutions per second
	s	Second
	Kgm**2	Moment of inertia
	mH	Inductivity
	Nm	Torque
	us	Microseconds
	uA	Microamperes
	uVs	Microvolt seconds
Effectiveness	so	Effective immediately
	cf	With confirmation
	re	Reset
	po	Power ON

General MD

General machine data

Open the *General Machine Data* window. Use the paging keys to page up and down.

Axis MD

Axis-specific machine data

Open the *Axis-Specific Machine Data* window. The softkey bar is extended by the **Axis +** and **Axis -** softkeys.

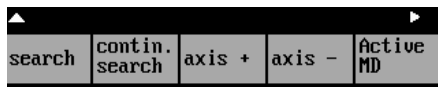


Fig. 7-20

The data of the axis are displayed.

Other MD

Other machine data

Open the *Other Machine Data* window. Use the paging keys to page up and down.

Display MD

Display machine data

Open the *Display Machine Data* window. Use the paging keys to page up and down.

Search

Search

Enter the number or name of the machine data you want to find and press **Input**.

The cursor jumps to the target data.

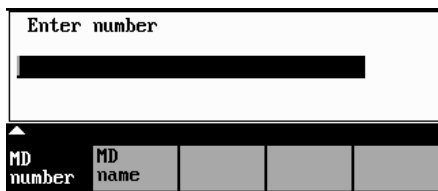


Fig. 7-21

Continue search

The search for the next number or name continues.

Axis +

The **Axis +** and **Axis -** softkeys are used to switch over to the machine data area of the next or previous axis.

Axis -

Active MD

This softkey is used to activate the machine data marked with "cf" .

Active MD

This softkey can be used to activate the machine data marked with "cf".

Display bright

Brightness

This softkey can be used to adjust the brightness of the screen.

Display darker

The power-up setting can be input via a display machine data. The adjustment via these softkeys does not effect the setting in the display machine data.

Change lang.

Switching the language

Use the **Change lang.** softkey to switch between foreground and background languages.

Programming

8.1 Fundamentals of NC programming

8.1.1 Program structure

Structure and contents

The NC program consists of a sequence of blocks (see Table 8-1).

Each block constitutes a machining step.

Instructions are written in a block in the form of **words**.

The last block in the sequence contains a special word for the **end of program: M2**.

Table 8-1 NC program structure

Block	Word	Word	Word	...	; Comment
Block	N10	G0	X20	...	; 1st block
Block	N20	G2	Z37	...	; 2nd block
Block	N30	G91	; ...
Block	N40	
Block	N50	M2			; End of program

Program names

Every program has its own program name.

Note

When generating the program, its name can be freely chosen provided the following conditions are complied with:

- The first two characters must be letters;
 - otherwise letters, digits or underscore may be used.
 - Do not use more than 8 characters.
 - Do not use separators (see Section "Character Set")
-

Example: **FRAME5231**

8.1.2 Word structure and address

Functionality/structure

The word is an element of a block and is mainly a control instruction.

The word (see Fig. 8-1) consists of

- an **address character**

The address character is generally a letter,

- and a **numerical value**.

The numerical value consists of a sequence of digits. A preceding sign or a decimal point can be added to this sequence for certain addresses.

A positive sign (+) can be omitted.

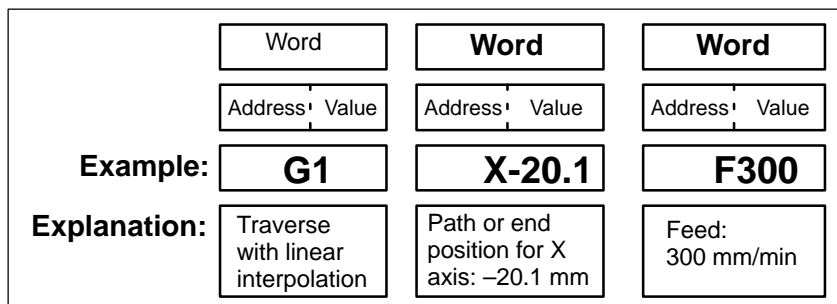


Fig. 8-1 Word structure

Several address characters

A word may also contain several address letters. In such cases, however, an "=" sign must be inserted to assign the numerical value to the address letters.

Example: CR=5.23

8.1.3 Block structure

Functionality

A block should contain all data required to execute a machining step.

The block generally consists of several **words** and always ends with the **end-of-block character "LF"** (line feed). This character is automatically generated when the carriage return or **Input** key is pressed during typing.

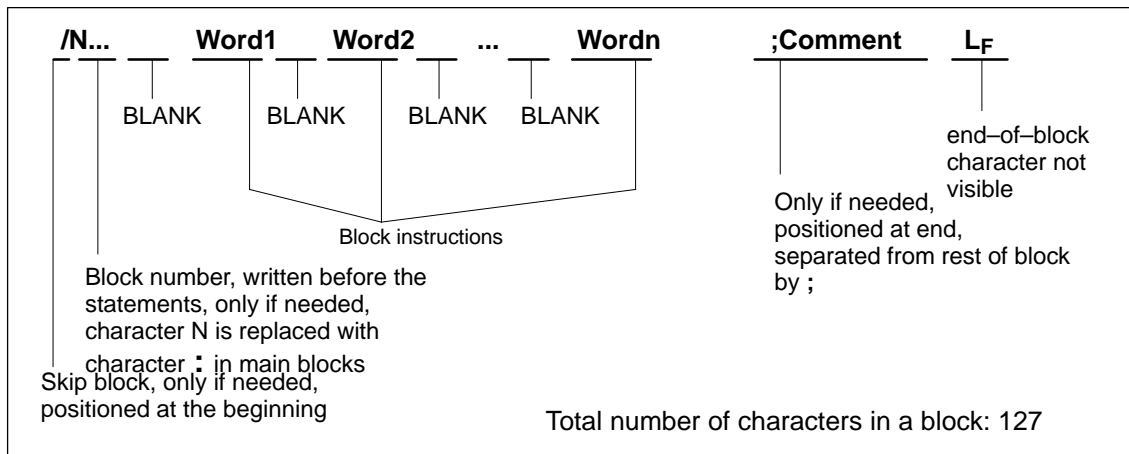


Fig. 8-2 Diagram of block structure

Word sequence

When a block contains more than one statement, the words in the block should be arranged in the following sequence:

N... G... X... Y... Z... F... S... T... D... M...

Note with regard to the block numbers

Select the block numbers first in steps of 5 or 10. This will allow you to insert blocks later while retaining the ascending order of the block numbers.

Block skipping (see Fig. 5-3)

Program blocks that must not be executed during every program run can be marked with a slash "/" in front of the block number word.

Block skipping is activated by means of an operator input or by the interface control (signal). A program section can be skipped by skipping several successive blocks with "/".

If block skipping is active during program execution, none of the blocks marked with "/" is executed. Any statements contained in such blocks are ignored. The program continues at the next block not marked.

Comment, remark

Comments (remarks) can be used to explain the statements in the blocks of a program.

Comments are displayed together with the other contents of the block in the current block display.

Programming example

```
N10 ; G&S Order No. 12A71
N20 ; Pump part 17, Drawing No.: 123 677
N30 ; Program created by Mr. Adam Dept. TV 4
:50 G17 G54 G94 F470 S20 D0 M3 ;Main block
N60 G0 G90 X100 Y200
N70 G1 Y185.6
N80 X112
/N90 X118 Y180 ;Block can be skipped
N100 X118 Y120
N110 X135 Y70
N120 X145 Y50
N130 G0 G90 X200
N140 M2 ;End of program
```

8.1.4 Character set

The following characters can be used for programming and are interpreted according to the following definitions:

Letters

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

No distinction is made between upper-case and lower-case letters. Lower-case letters are therefore equivalent to upper-case letters.

Digits

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

Printable special characters

(Left round bracket
)	Right round bracket
[Left square bracket
]	Right square bracket
<	Less than
>	Greater than
:	Main block, label termination
=	Assignment, equals
/	Division, block skip
*	Multiplication
+	Addition, positive sign
-	Subtraction, negative sign
"	Quotation marks
_	Underscore (together with letters)
.	Decimal point
,	Comma, separator
;	Start of comment
%	Reserved, do not use
&	Reserved, do not use
'	Reserved, do not use
\$	Reserved, do not use
?	Reserved, do not use
!	Reserved, do not use

Non-printable special characters

LF	End-of-block character
Blank	Separator between words, blank
Tabulator	Reserved, do not use.

8.1.5 Overview of instructions

Address	Meaning	Value Assignment	Information	Programming
D	Tool offset number	0 ... 9, integer values only, without sign	Contains offset data for a particular tool T...; D0→offset values= 0, max. 9 D numbers per tool	D...
F	Feed (in combination with G4, the dwell time is also programmed under F)	0.001 ... 99 999.999	Tool/workpiece path traverse rate in mm/min or mm/rev., depending on G94 or G95	F...
G	G function (preparatory function)	Only specific integer values	The G functions are divided into G groups. Only one G function of a group can be programmed in any one block. A G function can be modal (until canceled by another function of the same group) or non-modal – it is only active for the block in which it is programmed. G group:	G...
G0	Linear interpolation with rapid traverse		1: Motion commands (interpolation type) modally effective	G0 X... Y... Z...
G1 *	Linear interpolation with feed			G1 X... Y... Z... F...
G2	Circular interpolation with feed			G2 X... Y... I... J... F... ;Center and end point 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
G3	Circular interpolation in clockwise direction			G3 ;otherwise as with G2
G5	Circular interpolation via Intermediate point			G5 X... Y... Z... IX=... JY=... KZ=... F...
G33	Thread cutting with constant pitch			S... M... ;Spindle speed, direction G33 Z... K... ;Tapping with compensating chuck, e.g. in Z axis

G331	Thread interpolation		N10 SPOS=... ;Spindle in position control N20 G331 Z... K... S... ;Tapping without compensating chuck, e.g. in Z axis ;RH or LH thread is defined by the sign of the lead (e.g. K +): + : as with M3 - : as with M4
G332	Thread interpolation – retraction		G332 Z... K... ;Tapping without compensating chuck, e.g. in Z axis Retraction movement ; Sign of lead as with G331
G4	Dwell time	2: Special movements, non-modal	G4 F... ;separate block or G4 S.... ;separate block
G63	Tapping with compensating chuck		G63 Z... F... S... M...
G74	Reference-point approach		G74 X... Y... Z... ;separate block
G75	Fixed-point approach		G75 X... Y... Z... ;separate block
G158	Programmable offset		3: Write to memory non-modal
G258	Programmable rotation	G258 RPL=... ; Rotation in current plane G17 to G19, separate block	
G259	Additive programmable rotation	G259 RPL=... ; add. rotation in current plane G17 to G19, separate block	
G25	Lower spindle speed limit	G25 S... ;separate block	
G26	Upper spindle speed limit	G26 S... ;separate block	
G17 *	X/Y plane	6: Plane selection modal	
G18	Z/X plane		
G19	Y/Z plane		
G40 *	Tool radius compensation OFF	7: Tool radius compensation modal	
G41	Tool radius compensation left of contour		
G42	Tool radius compensation right of contour		

G500 *	Settable zero offset OFF	8: Settable zero offset modal	
G54	1st settable zero offset		
G55	2nd zero offset		
G56	3rd zero offset		
G57	4th zero offset		
G53	Non-modal suppression of settable zero offset	9: Suppression of settable zero offset non-modal	
G60 *	Exact positioning	10: Approach behavior modal	
G64	Continuous path operation		
G9	Non-modal exact stop	11: Non-modal exact positioning non-modal	
G601 *	Exact positioning window fine for G60, G9	12: Exact positioning window modal	See Section 8.3.12 "Exact Stop/Continuous Path Operation..."
G602	Exact positioning window coarse G60, G9		
G70	Dimensions in inches	13: Dimensions in inches/metric values modal	
G71 *	Dimensions in metric values		
G90 *	Absolute dimensioning	14: Absolute/incremental dimensions modal	
G91	Incremental dimensioning		
G94 *	Feed F in mm/min	15: Feed/spindle modal	
G95	Feed F in mm/rev. of spindle		
G901 *	Feed override for circle ON	16: Feed override modal	See Section 8.3.10 "Feed F"
G900	Feed override OFF		
G450 *	Transition circle	18: Behavior at corners with tool radius compensation modal	
G451	Intersection		
The functions marked with an * are active from the beginning of the program (with the version of the control system intended for milling unless otherwise programmed)			

Address	Meaning	Value Assignment	Information	Programming
I	Interpolation parameter	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	For X axis, meaning depends on whether G2,G3→circle center or G33, G331, G332→thread pitch has been programmed	see G2, G3, G33, G331 and G332
J	Interpolation parameter	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	For Y axis, otherwise as for I	see G2, G3, G33, G331 and G332
K	Interpolation parameter	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	For Z axis, otherwise as for I	see G2, G3, G33, G331 and G332
L	Subroutine, name and call	7 decimal places, integers only, without sign	Instead of a user-defined name L1 ... L9999999 can also be selected; this also calls the subroutine in its own block Note: L0001 is not the same as L1	L.... ;separate block
M	Miscellaneous function	0 ... 99 integers only, without sign	E.g. to trigger actions, such as "Coolant ON", max. 5 M functions in one block,	M...
M0	Programmed stop		Machining is stopped at the end of a block containing M0, operation is continued with "START"	
M1	Optional stop		As for M0, but operation only stops if a special signal has been given	
M2	End of program		Programmed in the last block to be executed	
M30	–		Reserved, do not use	
M17	–		Reserved, do not use	
M3	Spindle clockwise rotation			
M4	Spindle counterclockwise rotation			
M5	Spindle stop			
M6	Tool change		Only if activated with M6 in machine data, otherwise tool change performed directly with T command	
M40	Automatic speed change			

Address	Meaning	Value Assignment	Information	Programming
M41 to M45	Speeds 1 to 5			
M70	–		Reserved, do not use	
M...	Remaining M functions		This functionality is not predefined in the control system and can therefore be freely assigned by the machine manufacturer.	
N	Block number – subblock	0 ... 9999 9999 integer only, without sign	Can be used with a number to identify blocks, programmed at the beginning of a block	E.g.: N20
:	Block number – main block	0 ... 9999 9999 integer only, without sign	Special identification of blocks – used instead of N... , this block should contain all the instructions for a complete set of subsequent machining operations	E.g.: :20
P	Number of subroutine passes	1 ... 9999 integer only, without sign	Programmed in the same block as a subroutine to be called several times, e.g. : N10 L871 P3 ; called three times	E.g.: L781 P... ;separate block
R0 to R249	Arithmetic parameters	$\pm 0.0000001 \dots 9999 9999$ (8 decimal places) or mit with exponent: $\pm (10^{-300} \dots 10^{+300})$	R0 to R99 – user assignable R100 to R249 – transfer parameters for the machining cycles	
Arithmetic functions			In addition to the 4 basic arithmetic operations + – * / the following arithmetic functions are also available:	
SIN()	Sine	in degrees		E.g.: R1=SIN(17.35)
COS()	Cosine	in degrees		E.g.: R2=COS(R3)
TAN()	Tangent	in degrees		E.g.: R4=TAN(R5)
SQRT()	Square root			E.g.: R6=SQRT(R7)
ABS()	Absolute value			E.g.: R8=ABS(R9)
TRUNC()	Integer fraction			E.g.: R10=TRUNC(R11)
RET	End of subroutine		Used instead of M2 to maintain continuous path mode	RET ;separate block
S	Spindle speed or dwelling with G4	0.001 ... 99 999.999	Spindle speed in rev/min, with G4 dwell time in spindle revolutions	S...

Address	Meaning	Value Assignment	Information	Programming
T	Tool number	1 ... 32 000 integer only, without sign	Tool change can be performed directly with T command or not until M6 is programmed. This can be set in the machine data.	T...
X	Axis	±0.001 ... 99 999.999	Position data	X...
Y	Axis	±0.001 ... 99 999.999	Position data	Y...
Z	Axis	±0.001 ... 99 999.999	Position data	Z...
AR	Aperture angle for circular interpolation	0.00001 ... 359.99999	To be specified in degrees, a method of defining the circle with G2/G3	see G2; G3
CHF	Chamfer	0.001 ... 99 999.999	Insert a chamfer of the specific length between two contour blocks	N10 X... Y... CHF=... N11 X... Y...
CR	Radius for circular interpolation	0.010 ... 99 999.999 Negative sign – for circle selection: greater than semi-circle	A method of defining a circle with G2/G3	see G2; G3
GOTOB	GOTO instruction backwards	–	Jumps to the block defined by the label, the target of the jump is located in the direction of the beginning of the program.	E.g.: N20 GOTOB MARKE1
GOTOF	GOTO instruction forwards	–	Jumps to the block defined by the label, the target of the jump is located in the direction of the end of the program.	E.g.: N20 GOTOF MARKE2
IF	Jump condition	–	If the jump condition is fulfilled, the jump goes to the next instruction, Comparators: = = equal to, <> not equal to > greater than, < less than >= greater than or equal to <= less than or equal to	E.g.: N20 IF R1>5 GOTOB MARKE1
IX	Intermediate point for circular interpolation	±0.001 ... 99 999.999	For the X axis, programmed for circular interpolation with G5	see G5
JY	Intermediate point for circular interpolation	±0.001 ... 99 999.999	For the Y axis, programmed for circular interpolation with G5	see G5
KZ	Intermediate point for circular interpolation	±0.001 ... 99 999.999	For the Z axis, programmed for circular interpolation with G5	see G5

Address	Meaning	Value Assignment	Information	Programming
LCYC...	Machining cycle call	Specified values only	Machining cycles have to be called in a separate block, the transfer parameters to be used must be assigned values Transfer parameters:	
LCYC82	Drilling, spot-facing		R101: Retraction plane (absolute) R102: Safety clearance R103: Reference plane (absolute) R104: Final drilling depth (absolute) R105: Dwell time in seconds	N10 R101=... R102=... N20 LCYC82 ;separate block
LCYC83	Deep-hole drilling		R101: Retraction plane (absolute) R102: Safety clearance R103: Reference plane (absolute)) R104: Final drilling depth (absolute) R105: Dwell time at drilling depth R107: Feed for drilling R108: Feed for first drilling depth R109: Dwell time at starting point and with swarf removal R110: First drilling depth (absolute) R111: Degreession amount R127: Machining type: Chip breaking=0 Swarf removal=1	N10 R101=... R102=... N20 LCYC83 ;separate block
LCYC840	Tapping with compensating chuck		R101: Retraction plane (absolute) R102: Safety clearance R103: Reference plane (absolute) R104: Final drilling depth (absolute) R106: Thread lead value R107: Direction of rotation of spindle for tapping Value: 3 for M3 4 for M4	N10 R101=... R102=... N20 LCYC840 ;separate block
LCYC84	Tapping without compensating chuck		R101: Retraction plane (absolute) R102: Safety clearance R103: Reference plane (absolute) R104: Final drilling depth (absolute) R105: Dwell time at thread base in seconds R106: Thread lead value R112: Speed for tapping R113: Speed for retraction	N10 R101=... R102=... N20 LCYC84 ;separate block

Address	Meaning	Value Assignment	Information	Programming
LCYC85	Boring_1		R101: Retraction plane (absolute) R102: Safety clearance R103: Reference plane (absolute) R104: Final drilling depth (absolute) R105: Dwell time at drilling depth in seconds R107: Feed for drilling R108: Feed on retract from drill hole	N10 R101=... R102=... N20 LCYC85 ;separate block
LCYC60	Row of holes		R115: Number of drilling/tapping cycle Value: 82, 83, 84, 840, 85 (corresponding to LCYC...) R116: Abscissa reference point R117: Ordinate reference point R118: Distance of first hole from reference point R119: Number of holes R120: Angle to position of row of holes R121: Distance between the holes	N10 R115=... R116=... N20 LCYC60 ;separate block
LCYC61	Hole circle		R115: Number of drilling/tapping cycle Value: 82, 83, 84, 840, 85 (corresponding to LCYC...) R116: Center point abscissa of hole circle (absolute) R117: Center point ordinate of hole circle (absolute) R118: Radius of circle hole R119: Number of holes R120: Starting angle (-180<R120<180) R121: Indexing angle	N10 R115=... R116=... N20 LCYC61 ;separate block

Address	Meaning	Value Assignment	Information	Programming
LCYC75	Cutting pockets, slots and circular pockets		R101: Retraction plane (absolute) R102: Safety clearance R103: Reference plane (absolute) R104: Pocket depth (absolute) R116: Pocket center point abscissa R117: Pocket center point ordinate R118: Pocket length R119: Pocket width R120: Corner radius R121: Max. infeed depth R122: Feed for down infeed R123: Feed for machining in the plane R124: Finishing allowance in the plane R125: Finishing allowance in the depth R126: Cutting direction Value: 2 for G2 3 for G3 R127: Machining type Value: 1 for roughing 2 for finishing	N10 R101=... R102=... N20 LCYC75 ;separate block
RND	Rounding	0.010 ... 99 999.999	Inserts a rounding with the radius value specified tangentially between two contour blocks	N10 X... Y... RND=... N11 X... Y...
RPL	Rotary angle with G258, G259	± 0.00001 ... 359.9999	Specified in degrees, angle for a programmable rotation in the current plane G17 to G19	see G258; G259
SF	Thread commencement point with G33	0.001 ... 359.999	Specified in degrees, the thread commencement point with G33 is offset by the specified amount (not relevant for tapping)	
SPOS	Spindle position	0.0000 ... 359.9999	Specified in degrees, the spindle stops at the specified position (spindle must be designed to do this)	SPOS=...
STOPRE	Preprocessing stop	–	Special function, the next block is not decoded until the block prior to STOPRE is completed	STOPRE ;separate block
\$P_TOOL	Active tool cutting edge	read-only	integer, DO to D9	IF \$P_TOOL==7 GOTOF ...

Address	Meaning	Value Assignment	Information	Programming
\$P_TOOL NO	Active tool number	read-only	integer, TO – T32000	IF \$P_TOOLNO==46 GOTOF ...
\$P_TOOL P	Tool number last programmed	read-only	integer, TO – T32000	IF \$P_TOOLNP==11 GOTOF ...

8.2 Position data

8.2.1 Plane selection: G17 to G19

Functionality

For example, in order to assign **tool radius and tool length compensation**, a plane with two axes is selected from three axes. It is possible to activate a tool radius compensation in this plane. The associated length compensation is carried out depending on the tool type (cutter, drill, turning tool, ...)

The length offset for drill and cutter is assigned the axis standing vertical on the selected plane (see Section 8.6 "Tool and Tool Offsets").

Another factor influencing the plane selection is described in conjunction with the respective functions (e.g. Section 8.5 "Rounding, Chamfer").

Furthermore, the individual planes are intended to define **the direction of rotation of the circle for circular interpolation** in clockwise or counterclockwise direction. Abscissa and ordinate are defined in the plane in which traversing along the contour of the circle is performed, and clockwise or counterclockwise traversing are thus also defined. Circles can also be traversed in another plane than the currently active G17 to G19 plane (see Section 8.3 "Axis Movements").

The following planes and axis assignments are possible:

Table 8-2 Planes and axis assignments

G Function	Plane (Abscissa/Ordinate)	Vertical Axis in Plane (Length Offset Axis for Drilling/Milling)
G17	X / Y	Z
G18	Z / X	Y
G19	Y / Z	X

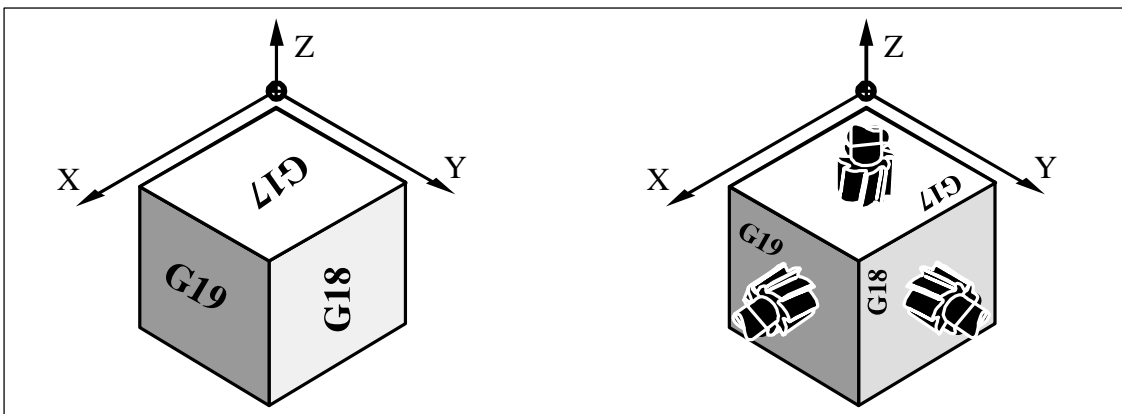


Fig. 8-3

Programming example

```

N10 G17 T... D... M...           ; X/Y plane selected
N20 ... X... Y... Z...           ; Tool length compensation in Z axis

```

8.2.2 Absolute/incremental dimensions: G90, G91**Functionality**

If the instruction G90 or G91 is active, the specified position information X, Y, Z is interpreted as a coordinate point (G90) or as an axis path to be traversed (G91). G90/G91 applies to all axes.

These instructions do **not determine the actual path** on which the end points are reached. This is done by a G group (G0,G1,G2,G3,... see Section 8.3 "Axis Movements").

Programming

```

G90           ;Absolute dimensioning
G91           ;Incremental dimensioning

```

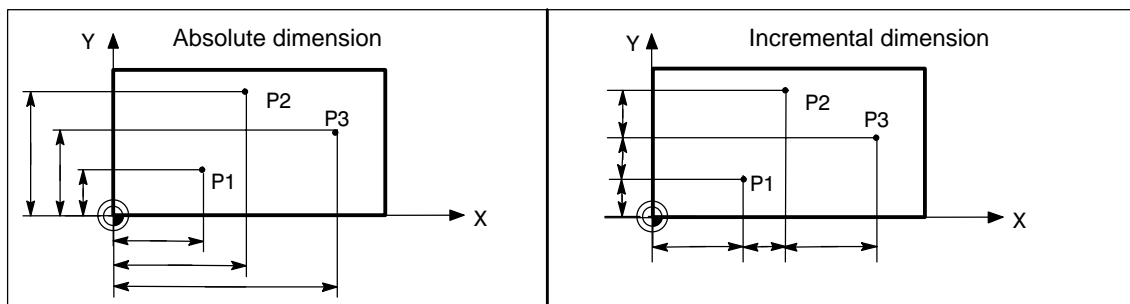


Fig. 8-4 Different dimensioning in the part drawing

Absolute dimension G90

If absolute dimensioning is selected, the dimension data refer to the **zero point of the currently active coordinate system** (workpiece coordinate system, current workpiece coordinate system or machine coordinate system). Which of the systems is active depends on which offsets are currently effective, i.e. programmable, settable or none at all.

G90 is active **for all axes** on program start and remains so until it is deactivated by G91 (incremental dimensioning selection) in a subsequent block (modal command).

Incremental dimension G91

If incremental dimensioning is selected, the numerical value in the position information corresponds to the **path to be traversed by an axis**. The **traversing direction** is determined by the sign.

G91 applies to all axes and can be deactivated by G90 (absolute dimensioning) in a later block.

Programming example G90 and G91

```

N10 G90 X20 Y90           ;Absolute dimensioning
N20 X75 Y-32             ;Absolute dimensioning remains active
...
N180 G91 X40 Y20        ;Switching to incremental dimensioning
N190 X-12 Y17           ;Incremental dimensioning remains active

```

8.2.3 Metric/inch dimensions: G71, G70**Functionality**

If a workpiece has dimensions that deviate from the basic system settings in the control system (inch or mm), then these can be entered directly in the program. The control system then converts them to the basic system.

Programming

```

G70           ;Inch dimensions
G71           ;Metric dimensions

```

Programming example

```

N10 G70 X10 Z30          ;Inch dimensions
N20 X40 Z50             ;G70 remains active
...
N80 G71 X19 Z17.3       ;Metric dimensions from here
...

```

Information

Depending on the current **default settings**, the control system interprets all geometric values as metric **or** inch dimensions. “Geometric values” also include tool offsets and settable zero offsets including the display as well as feed F in mm/min or inch/min.

The basic setting can be changed in the machine data.

All examples in this Guide assume that the **default setting** is **metric**.

G70 and G71 interpretes all geometrical data that refer directly to the **workpiece** in inches or metrically:

- Position information X, Y, Z with G0,G1,G2,G3,G33
- Interpolation parameters I, J, K (incl. lead)
- Circle radius CR
- **Programmable** zero offset (G158)

Any other geometric data not relating directly to the workpiece, such as feedrates, tool offsets, **settable** zero offsets, are not affected by G70/G71.

8.2.4 Programmable zero offset and rotation: G158, G258, G259

Functionality

The programmable zero offset is used for frequently recurring shapes/arrangements in different positions on a workpiece or simply when choosing a new reference point for the dimension data. The programmable offset produces the **current workpiece coordinate system**. The newly programmed dimension data then refer to this system.

The offset can be applied to all axes. The rotation is carried out in the current plane G17 or G18 or G19.

Programming

G158 X... Y... Z... ;Programmable offset,
old offset, rotation OFF
G258 RPL=... ;Programmable rotation,
old offset, rotation OFF
G259 RPL=... ;Additive programmable rotation
The instruction containing G158, G258, G259 require a separate block.

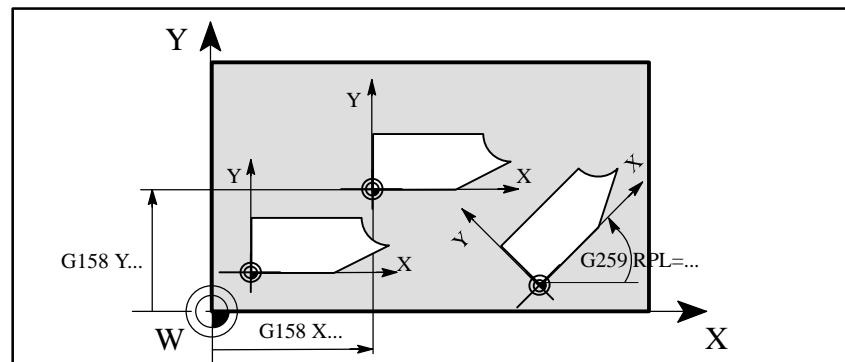


Fig. 8-5 Example of programmable offset, rotation on the workpiece

Offset G158

The instruction G158 can be used to program a zero offset for all axes. A new G158 instruction will **replace** all previous instructions of **programmable offset and rotation**; i.e. the old instructions are deleted.

Rotation G258

The instruction G258 can be used to program a rotation in the currently active plane (G17 to G19). A G258 instructions will **replace** all previous instructions of **programmable offset and rotation**, i.e. the old instructions are deleted.

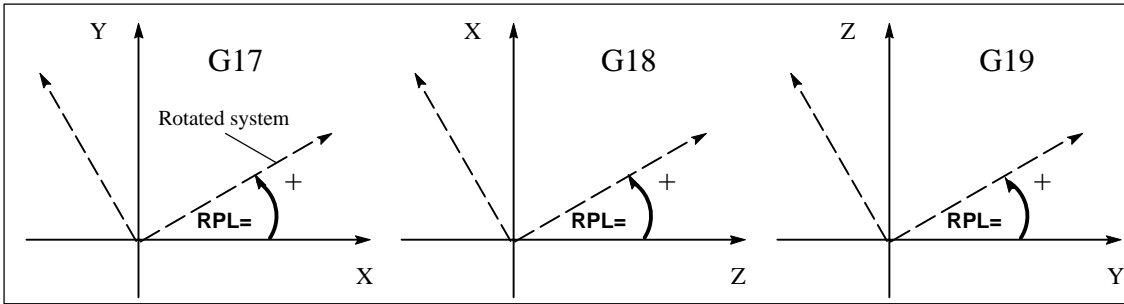


Fig. 8-6 Positive direction of the rotation angles in the different planes

Additive rotation G259

The instruction G259 can be used to **program** a rotation in the currently active plane (G17 to G19). If any one of the instructions G158, G258 or G259 is already active, the new rotation programmed under G259 is added to the previously programmed offset or rotation.

Delete offset, rotation

If the instructions **G158 without axes** or **G258 without RPL=** are written in a block, any **programmable offset or rotations active are deleted.**

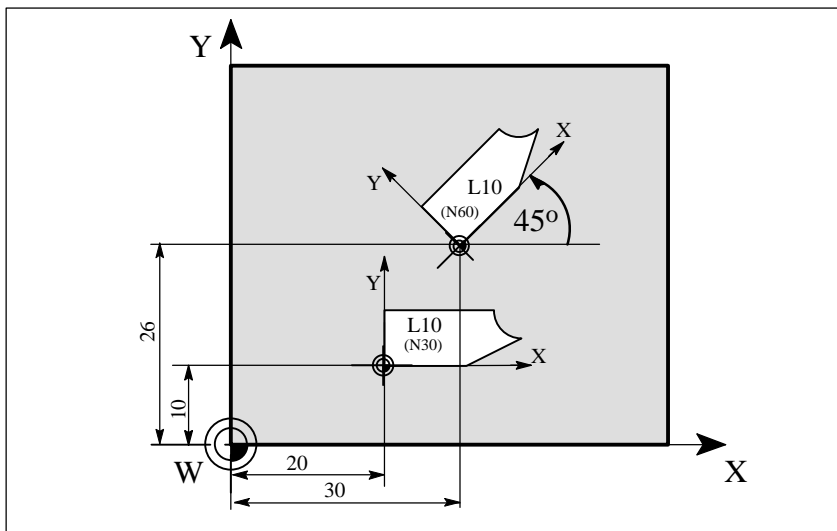


Fig. 8-7 Programming example for programmable offset and rotation

Programming example

```

N10 G17 ... ;X/Y plane
N20 G158 X20 Y10 ;Programmable offset
N30 L10 ;Subroutine call, contains the geometry
           to be offset
N40 G158 X30 Y26 ;New offset
N50 G259 RPL=45 ;Additive rotation by 45 degrees
N60 L10 ;Subroutine call
N70 G158 ;Offset and rotation deleted
...
Subroutine call see Section "Subroutine System"

```

8.2.5 Workpiece clamping – settable zero offset: G54 to G57, G500, G53

Functionality

The settable zero offset specifies the position of the workpiece zero point on the machine (offset between workpiece zero and machine zero). This offset is calculated when the workpiece is clamped on the machine and must be entered by the operator in the data field provided. The value is activated by the program through selection from four possible groups: G54 to G57.

See Section 3.5 "Enter/Modify Zero Offset" for operating sequence.

Programming

```

G54 ;1st settable zero offset
G55 ;2nd settable zero offset
G56 ;3rd settable zero offset
G57 ;4th settable zero offset
G500 ;Settable zero offset OFF – modal
G53 ;Settable zero offset OFF – non-modal,
      also suppresses the normal offset

```

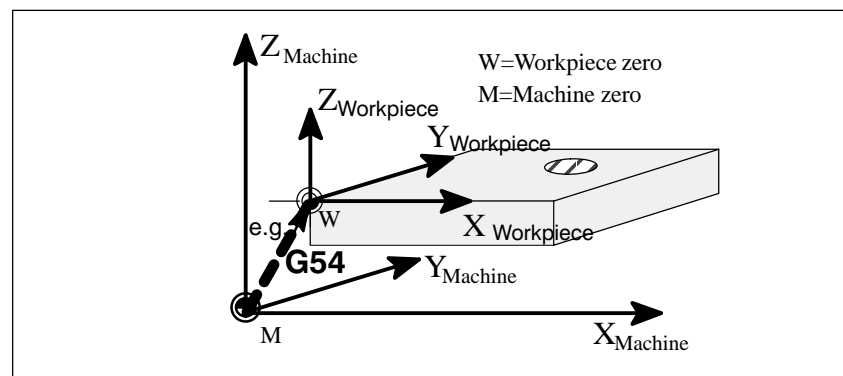


Fig. 8-8 Settable zero offset

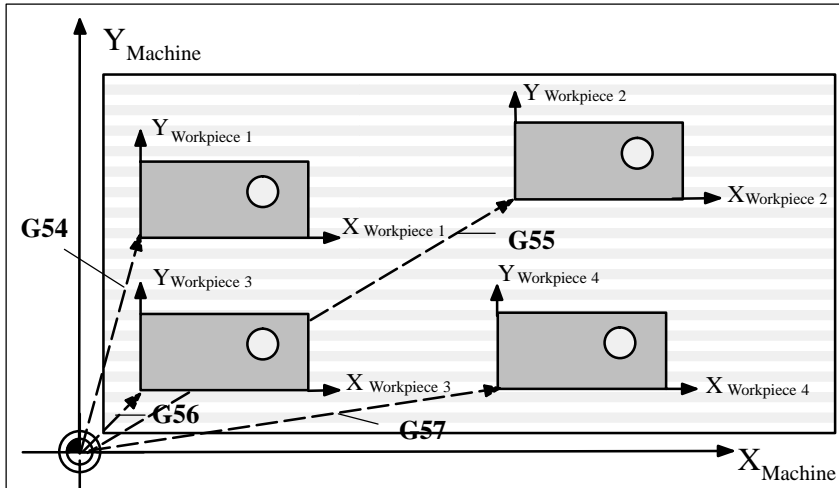


Fig. 8-9 Several workpiece clamping methods for drilling/milling

Programming example

```

N10 G54 ...           ;Call first settable zero offset
N20 L47              ;Machine workpiece 1, here as L47
N30 G55 ...         ;Call second settable zero offset
N40 L47              ;Machine workpiece 2, here as L47
N50 G56 ...         ;Call third settable zero offset
N60 L47              ;Machine workpiece 3, here as L47
N70 G57 ...         ;Call fourth settable zero offset
N80 L47              ;Machine workpiece 4, here as L47
N90 G500 G0 X...    ;Switch settable zero offset off
    
```

Subroutine call see Section 8.10 "Subroutine System"

8.3 Axis movements

8.3.1 Linear interpolation at rapid traverse: G0

Functionality

The rapid traverse motion G0 is used to position the workpiece rapidly, but **not to machine the workpiece directly**. All axes can be traversed simultaneously resulting in a linear path.

The maximum speed (rapid traverse) for each axis is set in the machine data. If only one axis is moving, it traverses at its own rapid traverse speed. If two axes are traversed simultaneously, then the path speed (resultant speed) is selected so as to obtain the **maximum possible path speed** based on the settings for all axes.

A programmed feed (F word) is irrelevant for G0. G0 remains effective until it is canceled by another instruction from the same group (G1, G2, G3,...).

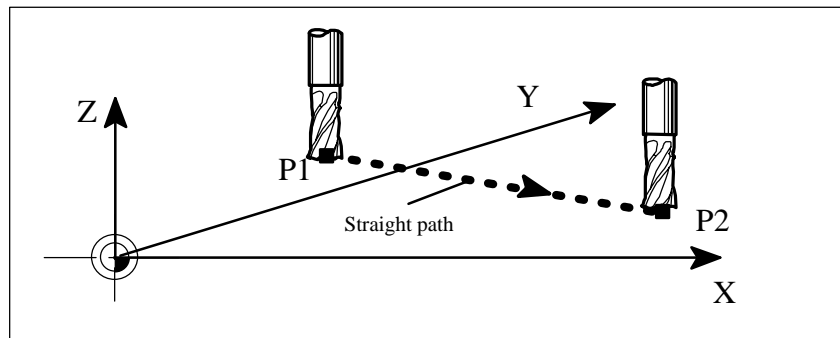


Fig. 8-10 Linear interpolation with rapid traverse from point P1 to P2

Programming example

```
N10 G0 X100 Y150 Z65
```

Information

A further group of G functions is provided for programming the approach to the position (see Section 8.3.12 "Exact Stop/Continuous Path Control: G60, G64"). G60 (exact stop) is linked to another group which allows various accuracy settings to be selected in a window. There is also a non-modal instruction, i.e. G9, for the exact stop function.

You should note these options when considering how to adapt the control to your positioning tasks.

8.3.2 Linear interpolation at feedrate: G1

Functionality

The tool moves from the starting point to the end point along a straight path. The **path speed** is defined by the programmed **F word**.

All axes can be traversed simultaneously.

G1 remains effective until it is canceled by another instruction from the same G group (G0, G2, G3, ...).

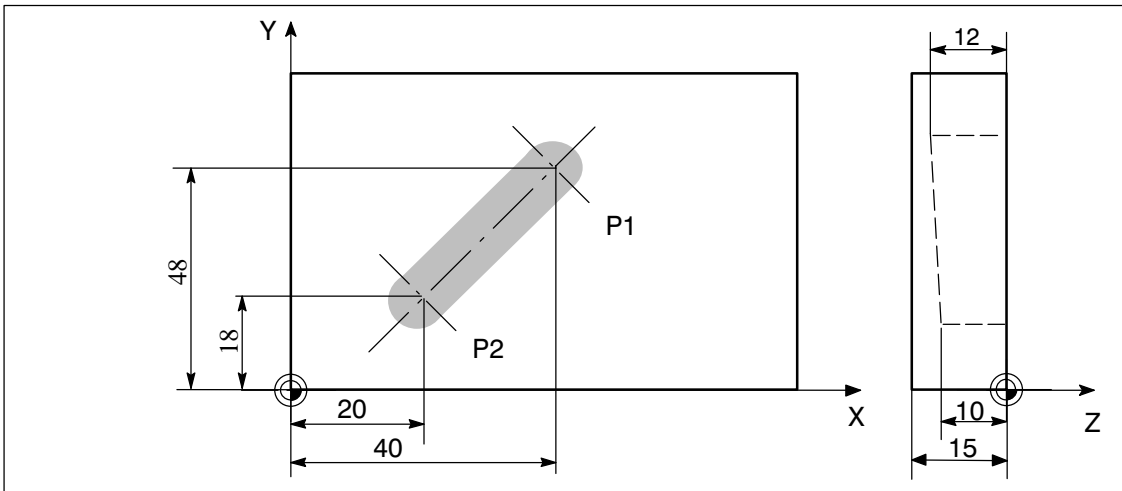


Fig. 8-11 Linear interpolation with three axes on a groove (example)

Programming example

N05 G0 G90 X40 Y48 Z2 S500 M3	;Tool is moving at rapid traverse to P1, 3 axes simultaneously, spindle speed = 500 rpm, CW rotation
N10 G1 Z-12 F100	;Infeed to Z-12, feed 100 mm/min
N15 X20 Y18 Z-10	;Tool is moving along a straight line in the space to P2
N20 G0 Z100	;Traverse clear with rapid traverse
N25 X-20 Y80	
N30 M2	;End of program

The spindle speed S and the direction of rotation of spindle M3/M4 must be entered for work-piece machining (see Section 8.4 "Spindle Movements").

8.3.3 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 – in clockwise direction
- G3 – in counterclockwise direction

The **path speed** is determined by the programmed **F word**. The required circle can be described in different ways:

- Center point and end point
- Circle radius and end point
- Center point and aperture angle
- Aperture angle and end point

G2/G3 remain effective until they are canceled by another instruction from the same G group (G0, G1, ...).

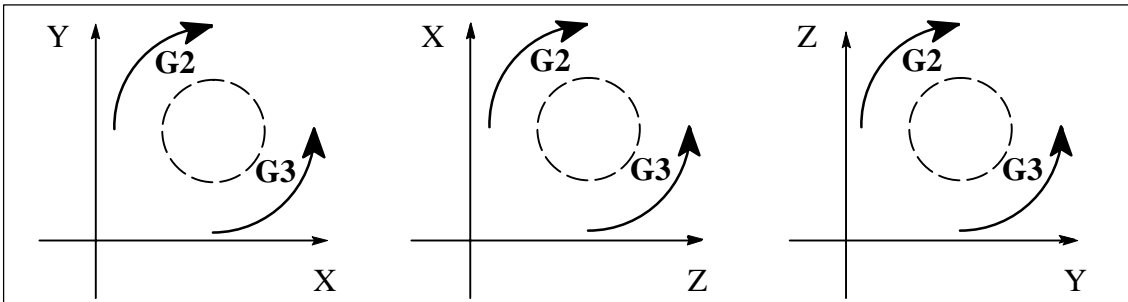


Fig. 8-12 Definition of direction of rotation around circle with G2/G3 in the 3 possible planes

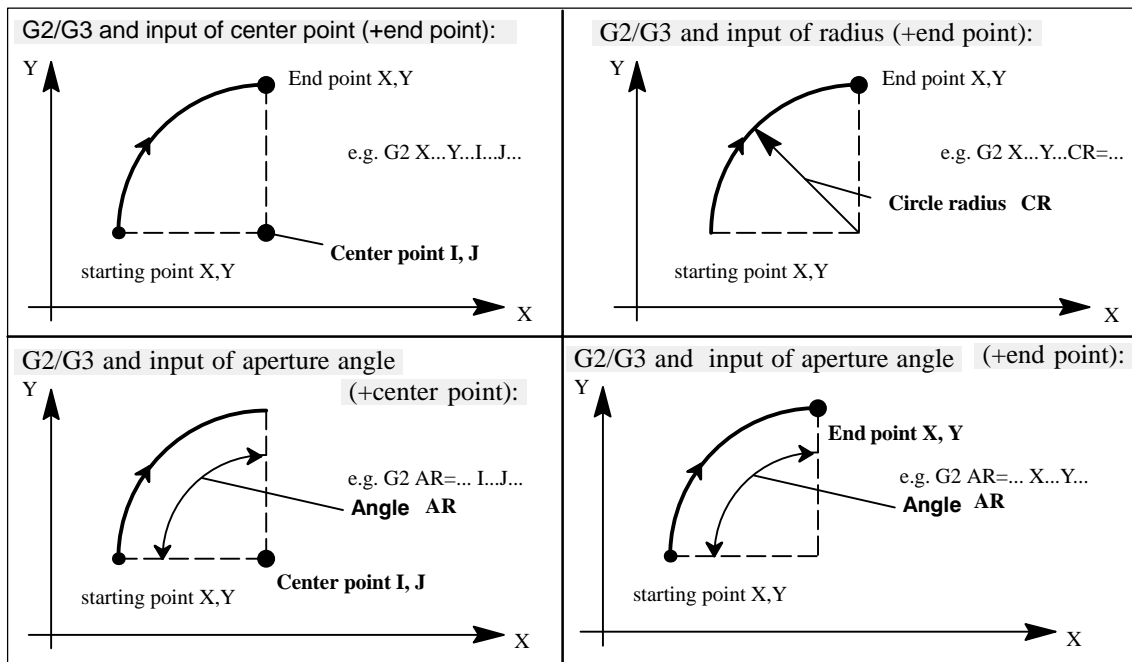


Fig. 8-13 Possible circle programming using G2/G3 with the X/Y axes

Information

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

The sign in front of CR=... defines the required circle type. 2 circles having the same start and end point, the same radius and the same direction are possible. The negative sign in front of CR=-... defines a circle whose circle segment is greater than a semi-circle; otherwise, a circle with the circle segment less than or equal to a semi-circle is defined as follows:

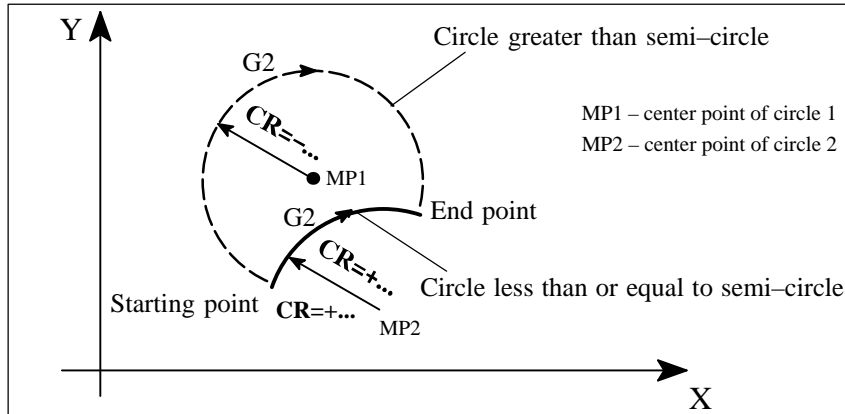


Fig. 8-14 Selection of circle from two possible circles with radius specification via the sign of CR=

Programming example

Center point and end point specification:

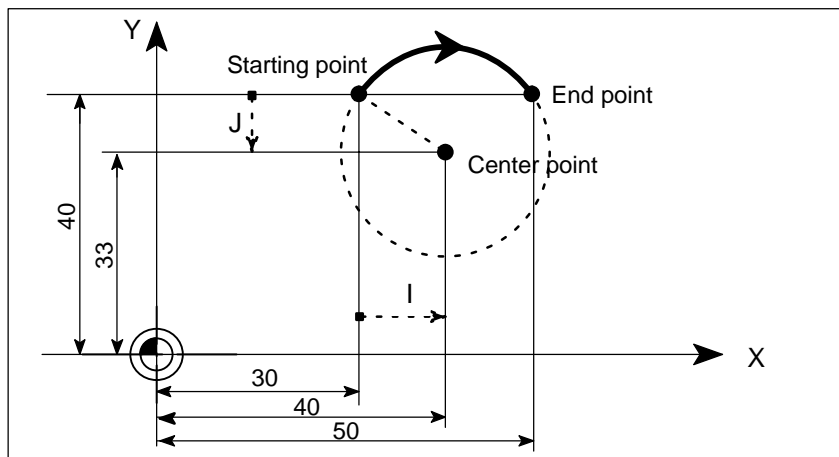


Fig. 8-15 Example of center and end point specification

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

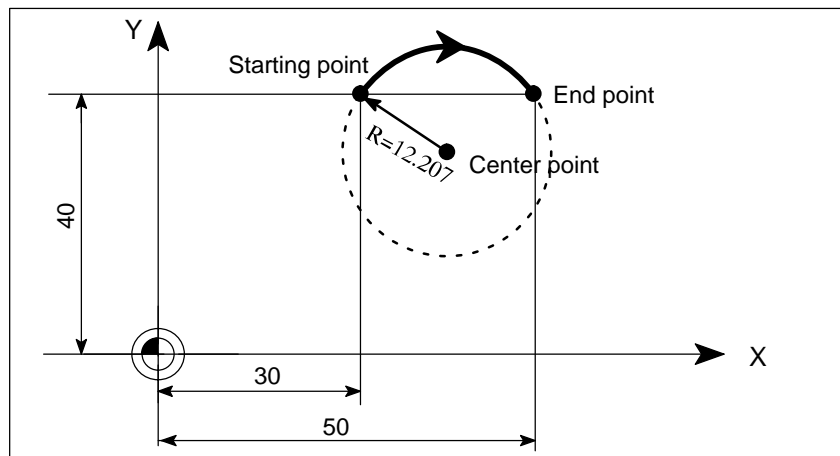
Programming example**End point and radius specification:**

Fig. 8-16 Example of end point and radius specification

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

Note: When the value for CR=... has a negative sign, a circle segment larger than a semi-circle is selected.

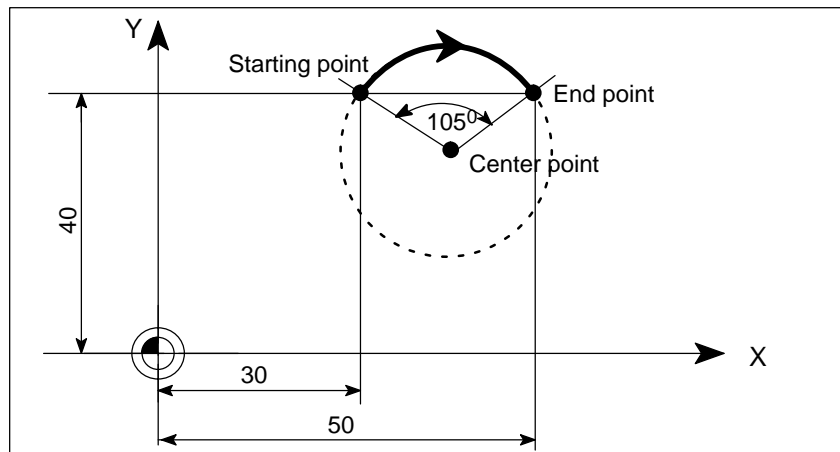
Programming example**End point and aperture angle:**

Fig. 8-17 Example of end point and aperture angle specification

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

Programming example

Center point and aperture angle:

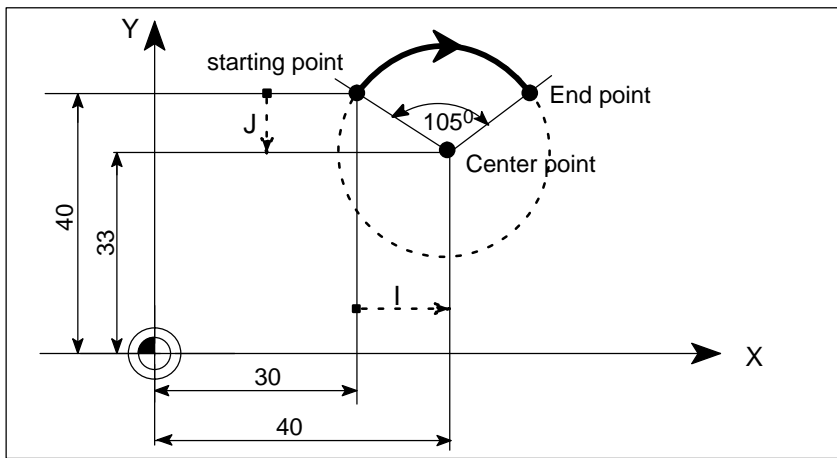


Fig. 8-18 Example of center point and aperture angle specification

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

Input tolerances for circle

The control system will only accept circles within a certain dimensional tolerance. The circle radius at the start and end points are compared for this purpose. If the difference is within the tolerance limits, the center point is set internally in the control. Otherwise, an alarm message is output..

The tolerance value can be set via the machine data.

8.3.4 Circular interpolation via intermediate point: G5

Functionality

The direction of the circle in this case is determined by the position of the intermediate point (between start and end positions).

G5 remains effective until it is canceled by another instruction from the same G group (G0, G1, G2, ...).

Note: The dimension setting G90 or G91 applies to both the end point **and** intermediate point!

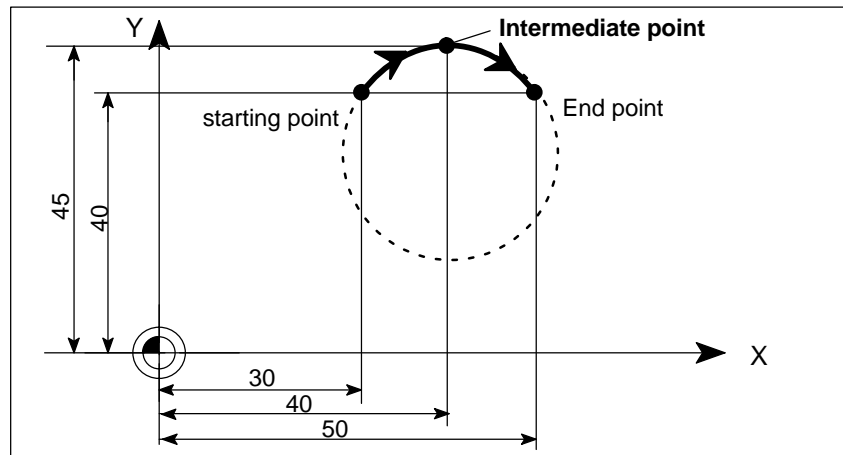


Fig. 8-19 Circle with end and intermediate point specification with G90 active

Programming example

```
N5 G90 X30 Y40 ;Circle starting point for N10
N10 G5 X50 Y40 IX=40 JY=45 ;End and intermediate points
```

8.3.5 Thread cutting with constant lead: G33

Functionality

This function requires a spindle with position measuring system.

Function G33 can be used to machine threads with constant lead. The appropriate tool provided,

tapping with compensating chuck

is possible.

The compensating chuck compensates any path differences to a certain degree.

The drilling depth is set by one of the axes X, Y or Z, and the lead is set by the associated I, J or K.

G33 remains effective until it is canceled by another instruction from the same G group (G0, G1, G2, G3, ...).

RH or LH threads

The direction of the thread, i.e. right-hand or left-hand, is determined by the setting for the direction of rotation of the spindle (M3 – clockwise rotation, M4 – counterclockwise rotation; see Section 8.4 “Spindle Movements”). To this aim, the speed setting must be programmed under address S, or a speed must be set.

Note:

A complete cycle for tapping with compensating chuck is provided by the standard cycle LCYC840 (see Section “Cycles”).

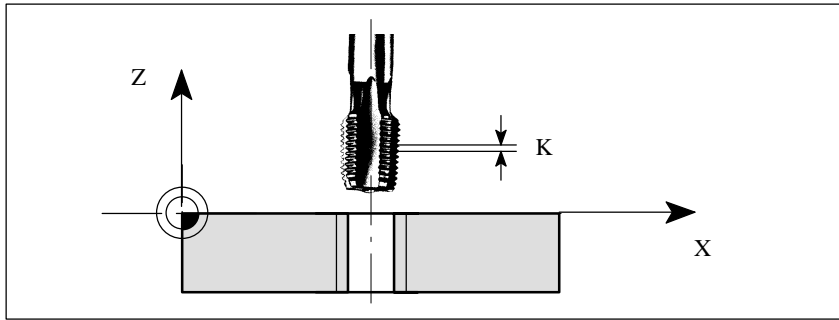


Fig. 8-20 Tapping with G33

Programming example

Metric thread 5,
 lead as per table: 0.8 mm/rev., drill hole already made:

```

N10 G54 G0 G90 X10 Y10 Z5 S600 M3           ;Approach starting point,
                                           ;CW rotation
N20 G33 Z-25 K0.8                           ;Tapping, end point – 25 mm
N40 Z5 K0.8 M4                               ;Retract, CCW rotation
N50 G0 X... Y... Z...
    
```

Velocity of the axes

For thread cuts with the G33 function, the speed of the axes for the thread length is determined by the spindle speed and the thread lead. **Feed F is not relevant in this respect.** However, it remains stored. The maximum speed defined in the machine data (rapid traverse) must not be exceeded.

Information

Important

- The setting of the spindle speed override switch (override spindle) should not be changed for thread machining operations.
- The feed override switch has no function in this block.

8.3.6 Tapping with compensating chuck: G63

Functionality

G63 can be used for tapping with compensating chuck. The programmed feed F must match with the spindle speed (S programmed or speed set) and with the lead of the drill:

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

The compensating chuck compensates any path differences to a certain degree.

The drill retraction is also carried out by means of G63, but with the opposite direction of rotation of spindle M3 ↔ M4.

G63 is non-modal. The G command of the Interpolation Type group (G0, G1, G2, ...) active prior to the block with G63 is activated again after the block with G63.

RH or LH threads

The direction of the thread, i.e. right-hand or left-hand, is determined by the setting for the direction of rotation of the spindle (M3 – clockwise rotation, M4 – counterclockwise rotation; see Section 8.4 “Spindle Movements”).

Note:

A complete cycle for tapping with compensating chuck (however, with G33 and the respective precondition) is provided by the standard cycle LCYC840 (see Section “Cycles”).

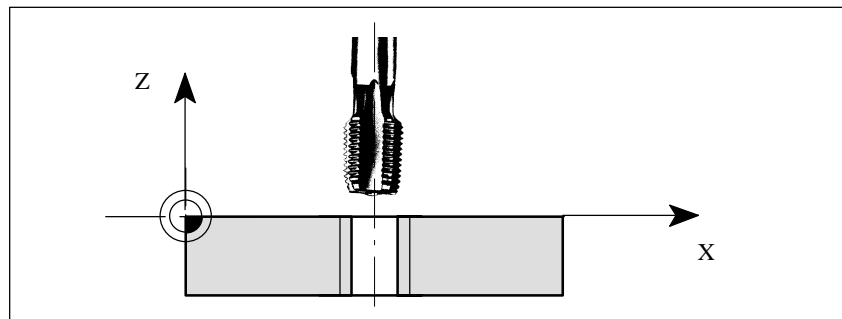


Fig. 8-21 Tapping with G63

Programming example

Metric thread 5,

lead as per Table: 0.8 mm/rev., drill hole already made:

N10 G54 G0 G90 X10 Y10 Z5 S600 M3 ;Approach starting point,
CW rotation

N20 G63 Z-25 F480 ;Tapping, end point -25 mm

N40 G63 Z5 M4 ;Retraction, CCW rotation

N50 X... Y... Z...

8.3.7 Thread interpolation: G331, G332

Functionality

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

G331/G332 can be used for tapping **without** compensating chuck, a respective spindle and axis dynamics provided.

If nonetheless a compensating chuck is used, the path differences to be compensated by the compensating chuck decrease. Tapping with higher spindle speed is thus possible.

G331 is used for drilling, and G332 for retracting the drill from the hole.

The drilling depth is set via one of the axes, and the lead via the associated I, J or K.

When G332 is active, the same lead as with G331 is programmed. The direction of rotation of spindle is automatically reversed.

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

Before tapping with G332, the spindle must be switched to position-controlled operation by SPOS=... (see also Section 8.4.3 "Positioning the Spindle").

RH or LH thread

The **sign of the lead** defines the direction of rotation of spindle:

positive: clockwise rotation (as with M3)

negative: counterclockwise rotation (as with M4)

Note:

A complete tapping cycle with thread interpolation is provided by the standard cycle LCYC84 (see Chapter "Cycles").

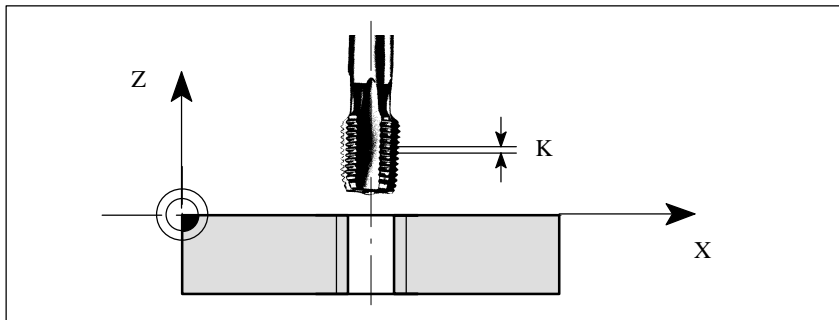


Fig. 8-22 Tapping with G331/G332

Programming example

Metric thread 5,

lead as per Table: 0.8 mm/rev., drill hole already made:

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 = CW rotation
of spindle, end point - 25 mm

N40 G332 Z5 K0.8 ;Retraction

N50 G0 X... Y... Z...

Velocity of the axes

When G331/G332 is active, the speed of the axis for the thread length results from the spindle speed and the lead. The **feed F is not relevant**. However, it remains stored. The maximum axis speed (rapid traverse) must not be exceeded. As a result, an alarm is output.

8.3.8 Fixed–point approach: G75

Functionality

G75 can be used to approach a fixed point on the machine, e.g. a tool change point. The position for all axes is fixed in machine data. No offset is active.

The speed of each axis is its rapid traverse.

G75 requires a separate block and is non–modal.

The G command for the Interpolation Type group (G0, G1, G2, ...) active prior to the block with G75 is activated again after the block with G75.

Programming example

```
N10 G75 X0 Y0 Z0
```

Note: The programmed numerical values for X, Y, Z are ignored.

8.3.9 Reference point approach: G74

Functionality

G74 is used to execute reference–point approach in the NC program. The direction and speed of each axis are stored in machine data.

G74 requires a separate block and is non–modal. The G command for Interpolation Type group (G0, G1, G2, ...) active prior to the block with G74 is activated again after the block with G74.

Programming example

```
N10 G74 X0 Y0 Z0
```

Note: The programmed numerical values for X, Y, Z are ignored.

8.3.10 Feedrate F

Functionality

The feedrate F is the **path speed** and represents the absolute value of the geometric total of the speed components of all axes involved. The individual axis speeds therefore result from the content of the axis path concerned in the total path.

The feedrate F is effective in interpolation modes G1, G2, G3, G5 and remains active until a new F word is inserted in the program.

Programming

F...

Note: Decimal points can be omitted in case of **integer values**, e.g. F300.

Unit for F- G94, G95

The unit of measurement for the F word is defined by G functions:

- G94 F as feedrate in **mm/min**
- G95 F as feedrate in **mm/rev.** of spindle
(only makes sense if spindle is in operation!)

Programming example

```
N10 G94 F310 ;Feed in mm/min...
N110 S200 M3 ;Spindle rotation
N120 G95 F15.5 ;Feed in mm/rev.
```

Note: Enter a new F word if you change from G94 to G95.

8.3.11 Feed override for circles: G900, G901

Functionality

When **tool radius compensation** (G41/G42, see Section 8.6.4) and **circle programming** are active, it is necessary to correct the feed in the cutter center point if the **programmed F value** is desired to be active in the circle contour.

Internal and external machining of the circle, as well as the current tool radius are automatically considered if the offsets are activated.

This compensation is not necessary for straight paths; the path speeds in the cutter center point and the programmed contour are identical, i.e. the programmed feed acts on the contour.

If you wish the programmed feed to active always in the cutter center point path, switch the feed override off using G900/G901 of the G group.

Programming

```
G900 ;Feed override OFF (programmed feed value
acts in the cutter center point)
G901 ;Feed override with circle ON
```

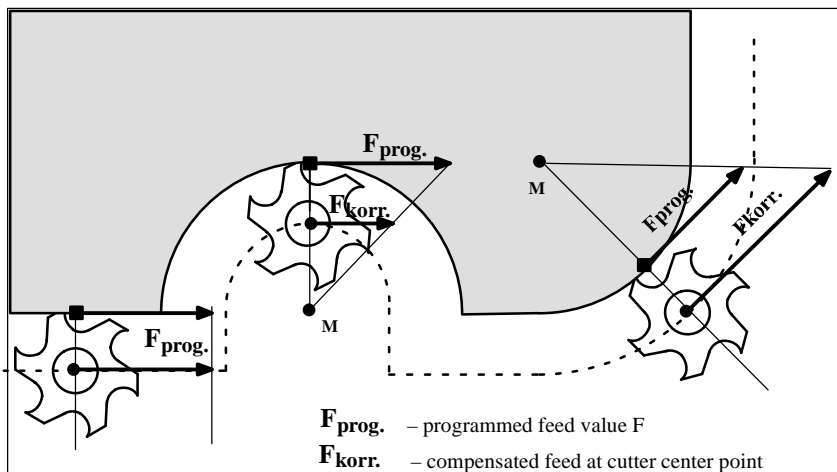


Fig. 8-23 Feed override G901 with internal/external machining

Corrected feedrate

–External machining:

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

–Internal circle machining:

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

r_{cont} : circle contour radius

r_{tool} : tool radius

Programming example

```

N10 G42 ...                ;Tool radius offset ON
N20 G901 ...              ;Feed override for circle ON
N30 G2 X... Y... I... J... F350 ;Feed value acts at contour
N40 G3 X... Y... I... J...   ;Feed value acts at contour
N70 G900                  ;Feed override OFF, programmed feed value acts in
                           cutter center point

```

8.3.12 Exact stop / continuous–path operation: G9, G60, G64**Functionality**

To set the traversing behavior at block limits and to control program advance, G functions are provided for optimum adaptation to the most varied requirements. Example: You want to position quickly with the axes or machine path contours over several blocks.

Programming	G60	;Exact stop – (modal)
	G64	;Continuous path control
	G9	;Exact stop – (non–modal)
	G601	;Exact stop window fine
	G602	;Exact stop window coarse

Exact stop G60, G9

If the exact stop function (G60 or G9) is active, the speed for reaching the exact target position is reduced towards zero at the end of the program block.

Another modally active G group can be set in conjunction with these functions to determine the moment at which the traversing motion in this block is finished so that processing of the next block can commence.

- G601 Exact stop window fine
Processing of the next block commences as soon as all axes have reached the “Exact stop window fine” (value in machine data).
- G602 Exact stop window coarse
Processing of the next block commences as soon as all axes have reached the “Exact stop window coarse” (value in machine data).

The selection of the exact stop window significantly affects the total machining time if many positioning operations need to be carried out. Fine adjustments require more time.

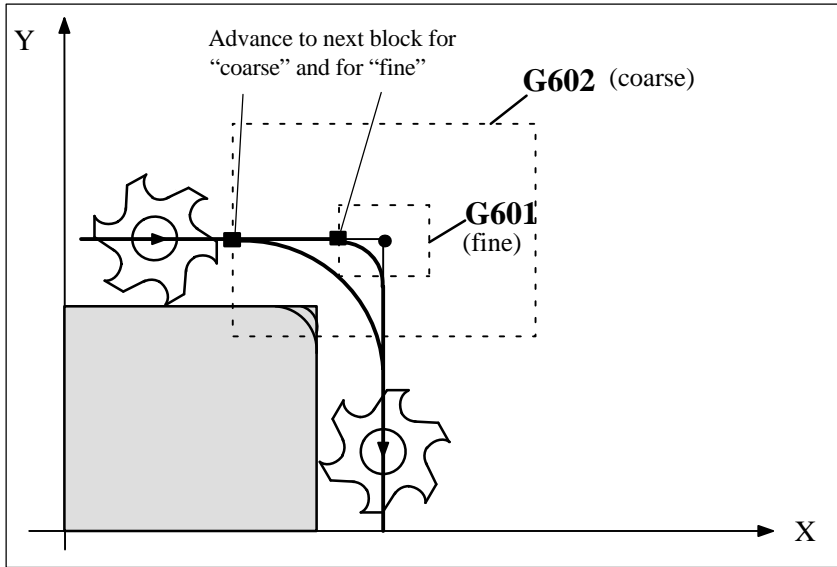


Fig. 8-24 Coarse or fine exact stop window, effective with G60/G9, zoomed view of window

Programming example

```

N5 G602 ;Exact stop window coarse
N10 G0 G60 X... ;Exact stop window modal
N20 X... Y... ;G60 continues to be active
...
N50 G1 G601 ... ;Exact stop window fine
N80 G64 X... ;Switchover to continuous path
operation
...
N100 G0 G9 X... ;Exact stop acts only in this block
N111 ... ;Return to continuous path
operation
...

```

Note: The G9 command generates an exact stop for the block in which it is programmed; in contrast, G60 remains active until it is canceled by G64.

Continuous path operation G64

The purpose of continuous path operation is to prevent braking of the axes at block limits and to change **to the next block** at the most **constant possible speed** (in the case of tangential transitions). The function operates with lookahead speed control to the next block.

In the case of non-tangential path transitions (corners), the speed is reduced to such an extent in some cases that none of the axes is capable of making a speed step change that is higher than the maximum acceleration rate. This results in **speed-dependent rounding at corners**.

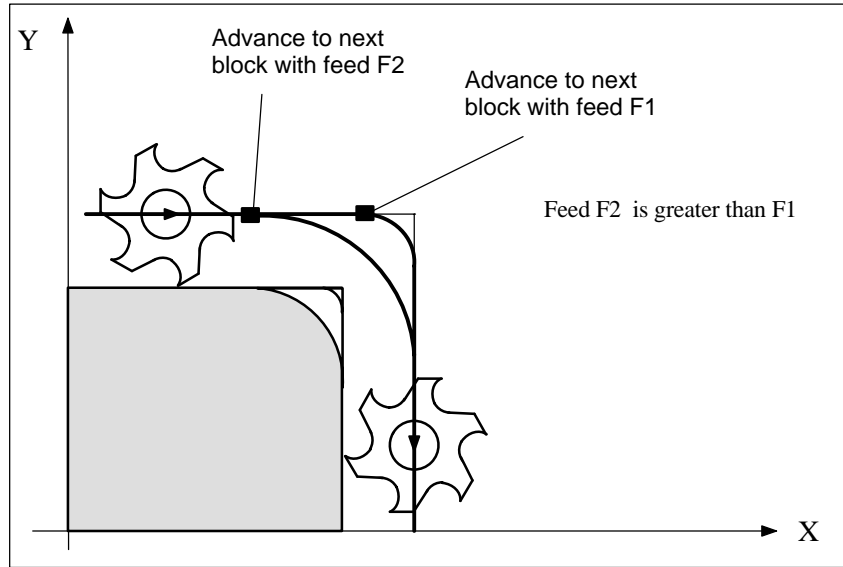


Fig. 8-25 Rounding at contour corners with G64

Programming example

```

N10 G64 G1 X... F...           ;Continuous path operation
N20 Y..                         ;Continuous path operation still
                                active
...
N180 G60 ...                    ;Switchover to exact stop
    
```

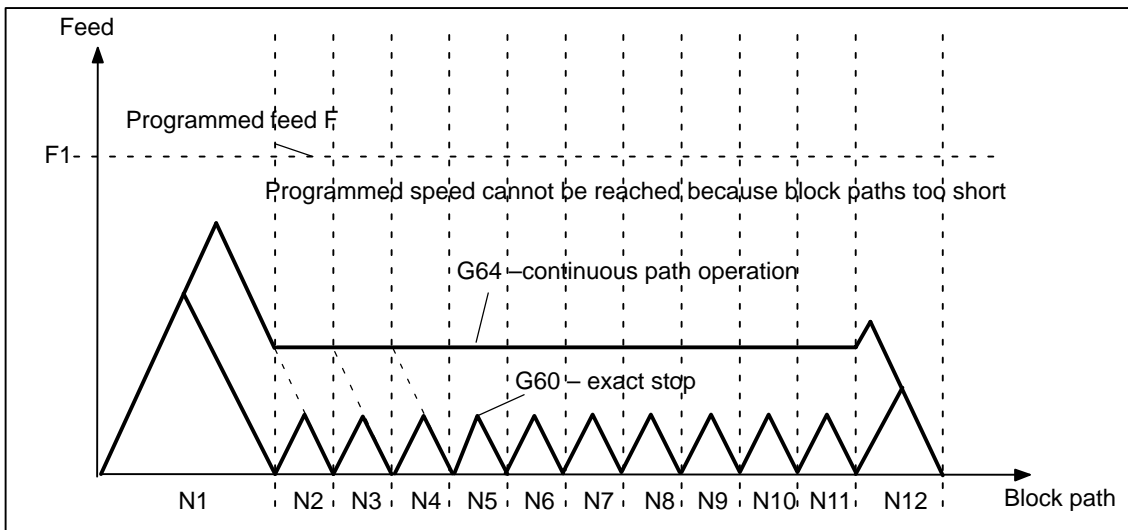


Fig. 8-26 Comparison between speed responses with G60 and G64 with short block paths

8.3.13 Dwell time: G4

Functionality

You can interrupt machining for a defined time by inserting a **separate block** with G4, e.g. for relief cutting operations.

The words with F... or S... are used for time specifications. Any previously programmed feed F or spindle speed S remain unaffected.

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 s
N20 Z70
N30 G4 S30 ;Dwell for 30 spindle revolutions,
           corresponds to t = 0.1 min with
           S=300 rev./min and 100 %
           speed override
N40 X... ;Feed and spindle speed values remain
         effective
```

Remark

G4 S.. can only be programmed if the machine has a controlled spindle (if the speed has also been programmed under address S..).

8.4 Spindle movements

8.4.1 Spindle speed S, directions of rotation

Functionality

The spindle speed is programmed under address S in revolutions per minute, if the machine has a controlled spindle. The direction of rotation and the start or end of the movement are specified by means of M commands (see Section 1.7 "Miscellaneous Function M").

M3	Clockwise rotation of spindle
M4	Counterclockwise rotation of spindle
M5	Spindle stop

Note: A decimal point may be omitted in the case of integer S values, e.g. S270.

Information

If you insert M3 or M4 in a **block with axis movements**, then the M commands will take effect **before** the axis movements.

Default setting: The axis movements will only start after the spindle has run up (M3, M4). M5 is also output prior to the axis movement. However, the axes will not wait for the spindle being stopped. The axis movements will already start before the spindle has come to a standstill. The spindle is stopped with program end or RESET.

Note:

Other settings can be configured via machine data.

Programming example

N10 G1 X70 Z20 F300 S270 M3	;Spindle powers up to 270 rev/min in clockwise rotation before axis traversal X, Z
...	
N80 S450 ...	;Speed change
...	
N170 G0 Z180 M5	;Z movement in block, spindle stop

8.4.2 Spindle speed limitation: G25, G26

Functionality

You can restrict the speed limit values that otherwise apply by programming a speed limit value using G25 or G26 and spindle address S. These functions also overwrite the values entered in the setting data.

G25 or G26 each requires a separate block. Any previously programmed speed S remains effective.

Programming

G25 S... ;Lower spindle speed limitation
 G26 S... ;Upper spindle speed limitation

Information

The maximum upper and lower spindle speed limits are set in a machine data. Setting data can be activated via the operator panel to limit the speed range still further.

Programming example

N10 G25 S12 ;Lower spindle limit speed: 12 rev/min
 N20 G26 S700 ;Upper spindle limit speed: 700 rev/min

8.4.3 Spindle positioning: SPOS

Functionality

Precondition: The spindle must be esigned such that open-loop position-control is possible.

The SPOS= function allows you to position the spindle in a specific **angular position**. It is then held in this position by a closed-loop position control function.

The **speed** of the positioning operation is defined in a machine data item.

The applicable **direction of rotation** is maintained from the M3/M4 movement until the positioning process is complete. When the spindle is positioned from standstill, the position is approached via the shortest possible path. In this case, the direction is determined by the start and end positions.

Exception: Initial movement of spindle, i.e. if the measuring system is not yet synchronized. In such cases, the direction is specified by a machine data.

The spindle movements are executed in parallel to any axis movements programmed in the same block. Processing of the block is complete after both movements have been executed.

Programming

SPOS=... ;Absolute position: 0 ... <360 degrees

Programming example

N10 SPOS=14.3 ;Spindle position 14.3 degrees
 ...
 N80 G0 X89 Z300 SPOS=25.6 ;Position spindle with axis
 movements.
 The block is complete once all
 movements have been executed.
 N81 X200 Z300;N81 block does not start until spindle
 position from N80 is reached.

8.5 Rounding, chamfer

Functionality

You can insert the elements “chamfer” and “rounding” at contour corners. The appropriate instruction, i.e. CHF=... or RND=... is programmed in the block with axis motions that leads into the corner.

Programming

CHF=... ;Insert chamfer, value: Length of chamfer
 RND=... ;Insert rounding, value: Rounding radius

Chamfer CHF=

A linear section is inserted between **linear and circular contours** in any combination. The edge is chamfered.

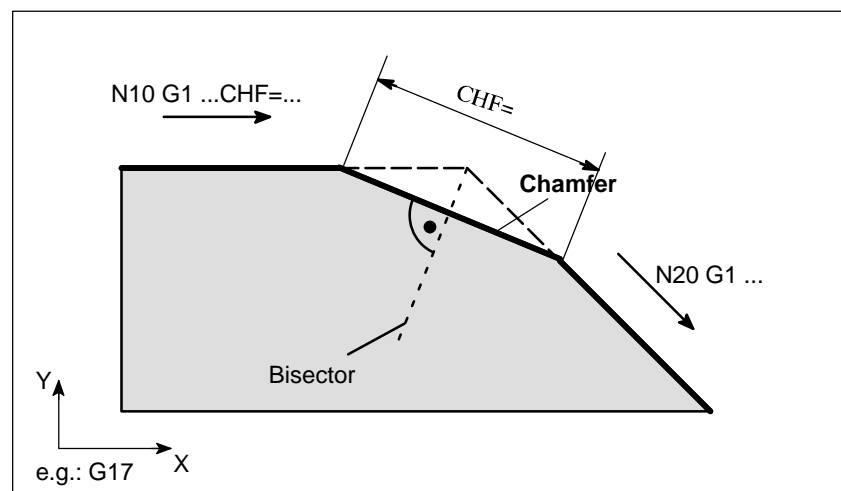


Fig. 8-27 Insertion of a chamfer between two linear contours (example)

Programming example for chamfer

N10 G1 Y... CHF=5 ;Insert 5 mm chamfer
 N20 X... Y...

Rounding RND=

A circular contour element is inserted with tangential transitions between **linear and circular contours** in any combination.

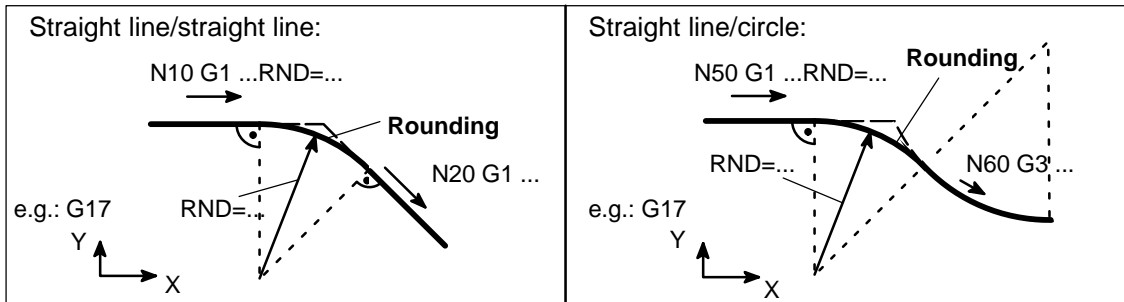


Fig. 8-28 Examples of rounding insertion

Programming example for rounding

```

N10 G1 Z... RND=8           ;Insert rounding with 8 mm radius
N20 X... Z...
...
N50 G1 Z... RND=7.3       ;Insert rounding with 7.3 mm radius
N60 G3 X... Z...
    
```

Information

The functions chamfer/rounding are carried out in the current plane G17 to G19.

Note:

The programmed value for the chamfer or rounding is automatically reduced when the contour programmed in one of the blocks involved is not sufficiently long.

No chamfer/rounding is inserted if

- more than three of blocks are programmed which does not contain any information about traversing in the plane;
- the plane is changed.

8.6 Tool and tool offset

8.6.1 General notes

Functionality

When you create programs for workpiece machining, you need not take into account tool lengths or cutter radii. Program the workpiece dimensions directly, e.g. as given in the workpiece drawing.

The tool data are entered separately in a special data area.

You merely call the required tool with its offset data in the program and activate the tool compensation if you wish to. The control system uses these data to carry out the required path compensations in order to produce the workpiece you have defined.

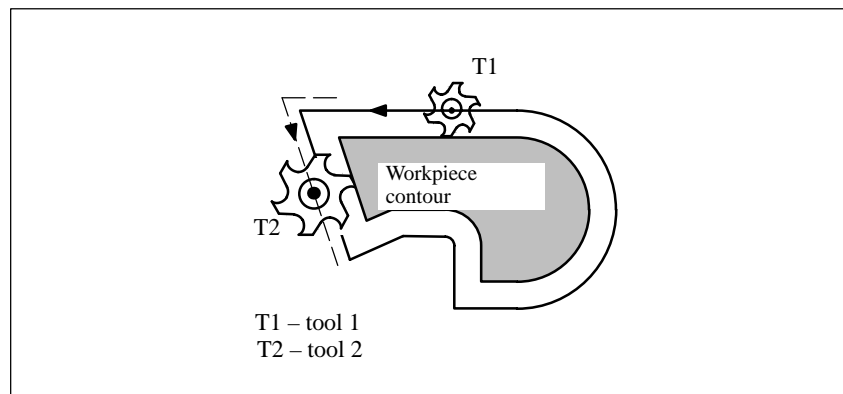


Fig. 8-29 Machining a workpiece with different tool radii

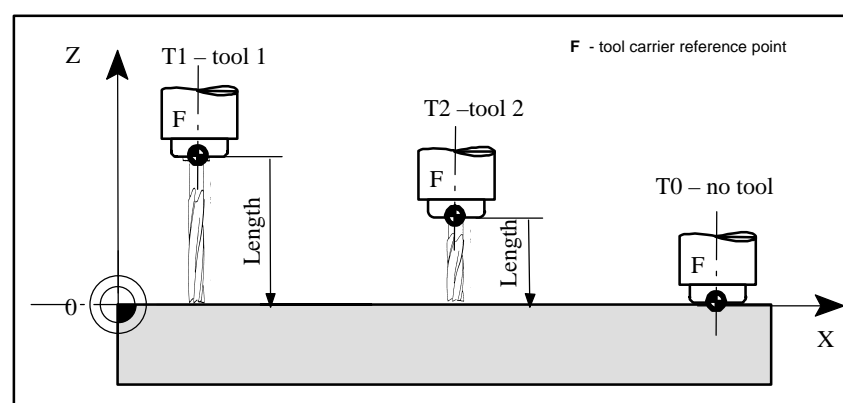


Fig. 8-30 Approaching the workpiece position Z0 – different length compensations

8.6.2 Tool T

Functionality

The required tool is selected by programming the T word. A machine data defines whether the T word represents a **tool change** or merely a **preselection**:

- Tool change (tool call) is carried out directly with the T word, or
- the change is carried out after the preselection with the T word by the additional instruction **M6** (see also Section 8.7 "Miscellaneous Function M").

Note:

If a certain tool has been activated, it remains stored even after the program has been executed and the control system has been switched off/on.

If you change the tool manually, the change must also be entered in the control system so that the CNC can recognize the tool. For example, you can start a block with the new T word in the MDA mode.

Programming

```
T...           ;Tool number: 1 ... 32 000, T0 – no tool
```

Note

A maximum of 15 tools can be stored in the control system at a time.

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        ;Carry out tool change, T14 is then active
```

8.6.3 Tool offset number D

Functionality

A certain tool can be assigned 1 to 9 data fields each with different tool offset blocks (for several tool edges). If a certain tool edge is required, it can be programmed with D and a respective number.

If no D word is entered, **D1** is **automatically** active.

Note

A maximum of 30 data fields (D numbers) with tool offset blocks can be stored in the control system at a time.

Programming

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

T1	D1	D2	D3	D9
T2	D1			
T3	D1			
T6	D1	D2	D3	
T9	D1	D2		
T...	D1	D2		

Fig. 8-31 Assignment of tool offset numbers to the tool (example)

Information

Tool length compensations take **immediate** effect when the tool is active; if no D number has been programmed, the values programmed under D1 are active for the length offsets. The compensation is performed with the first programmed traversal of the relevant length compensation axis. Pay attention to any active G17 to G19!

A **tool radius compensation** must additionally be activated by means of G41/G42.

Programming example

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

```
N5 G17 ;Defines axis assignment for offsets
N10 T1 ;Tool 1 is activated with associated D1
N11 G0 Z... ;When G17 is activated, Z is the length
            offset axis, the length offset compensation is
            superimposed here

N50 T4 D2 ;Change tool 4, D2 of T4 becomes active
...
N70 G0 Z... D1 ;D1 for tool 4 is active, only edge changed
```

Tool change **with M6** command:

```
N5 G17 ;Defines axis assignment for offsets
N10 T1 ;Tool preselection
...
N15 M6 ;Tool change, T1 is active with associated D1
N16 G0 Z... ;When G17 is activated, Z is the length offset
            axis, the length offset compensation is
            superimposed here
```

```

...
N20 G0 Z... D2 ;D2 for tool 1 is active
                When G17 is activated, Z is the length offset
                axis, the difference of the length offset D1→D2
                is superimposed here
N50 T4          ;Tool preselection T4,
                Note: T1 with D2 is still active !
...
N55 D3 M6      ;Tool change, T4 with the associated D3 is active
...
    
```

Contents of offset memory

Enter the following in the offset memory:

- Geometric quantities: length, radius
 These consist of several components (geometry, wear). The control system computes the components to a resultant quantity (e.g. total length 1, total radius). The total dimension calculated in each case takes effect when the offset memory is activated.
 The method used to compute these values in the axes depends on the tool type and the commands G17, G18, G19 (see Diagrams below).
- Tool type
 The tool type determines which geometry data are necessary and how they are computed (drill, cutter). The hundreds place is the only distinguishing digit:
 - Type 1xy : cutter
 - Type 2xy : drill

Tool parameters

The value for the relevant tool parameters is entered next to DP... . The tool type determines which parameters are required. Any tool parameters not needed must be set to “0”.

Tool type:	DP1= 100 (cutter)	
	Geometry	Wear
Length 1:	DP3	DP12
* Length 2:	DP4	DP13
* Length 3:	DP5	DP14
Radius:	DP6	DP15

Tool type:	DP1= 200 (drill)	
	Geometry	Wear
Length 1:	DP3	DP12
* Length 2:	DP4	DP13
* Length 3:	DP5	DP14

* Note: When tool types 1xy (cutter) and 2xy (drill) are selected, the parameters for length 2 and length 3 are only needed for special cases (e.g. multi-dimensional length offset for an angular head cutter).

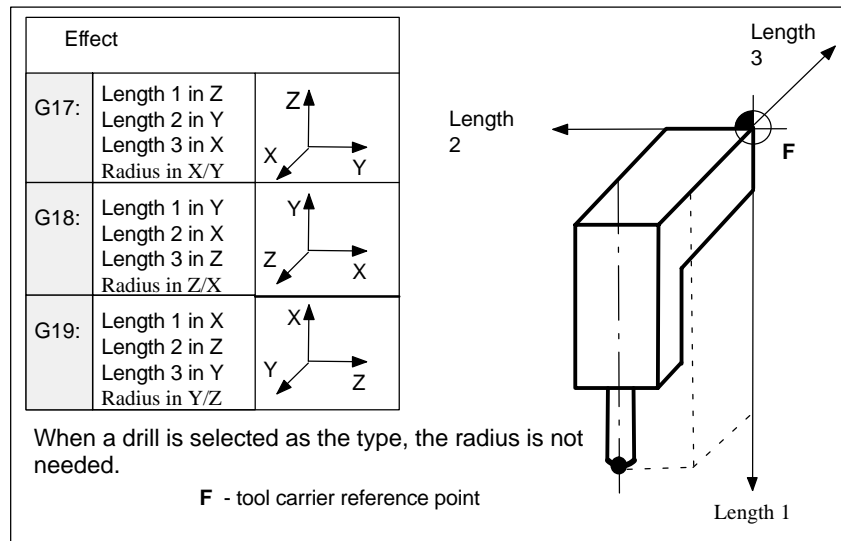


Fig. 8-32 Effect of the tool length offsets for 3D contouring control (special case)

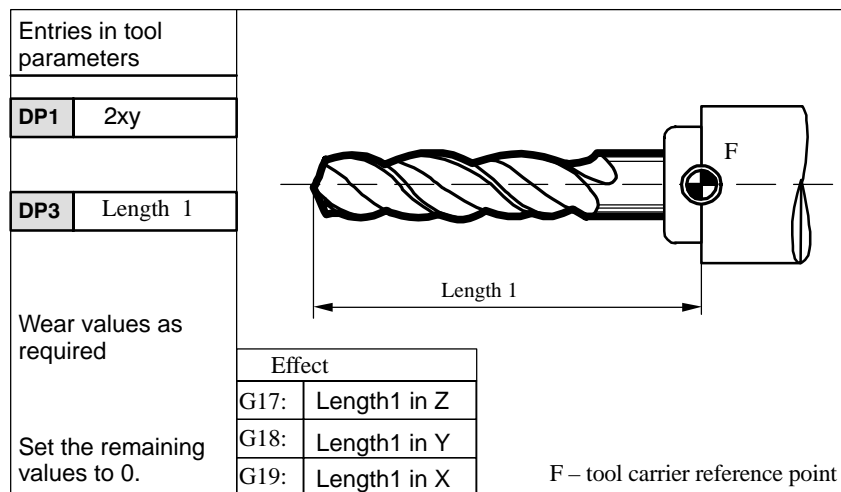


Fig. 8-33 Offset data required for a drill

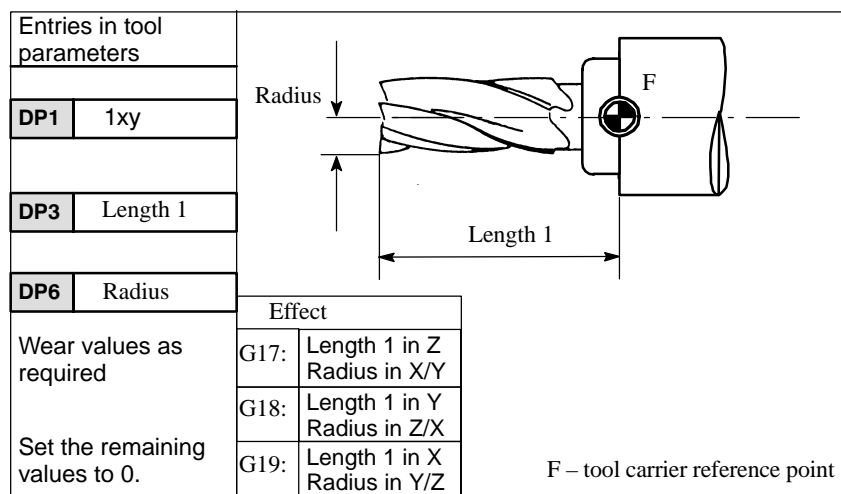


Fig. 8-34 Offset data required for a cutter

8.6.4 Selection of tool radius offset: G41, G42

Functionality

The control system is working with tool radius offset in the selected planes G17 to G19. A tool with a corresponding D number must be active. The tool radius offset is activated by G41/G42. The control system uses this value to calculate the required tool path equidistant for the currently active tool radius along the programmed contour automatically.

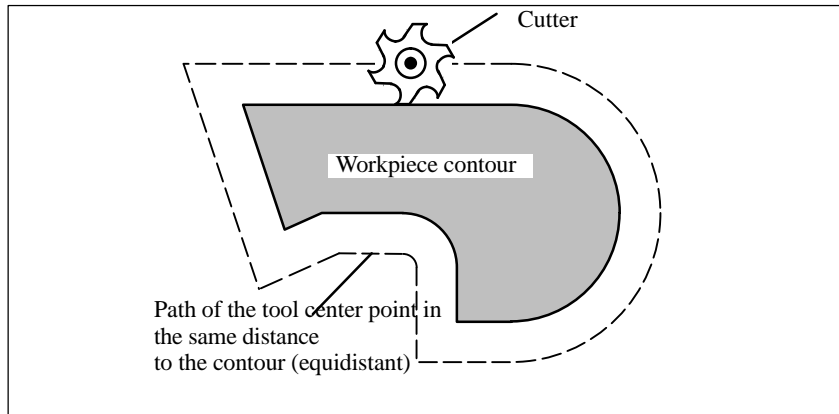


Fig. 8-35 Tool radius offset

Programming

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

Note: This function can only be selected for linear interpolation (G0, G1). Program both axes of the plane (e.g. X, Y for G17). If you specify only one axis, the last programmed value is automatically set for the second axis.

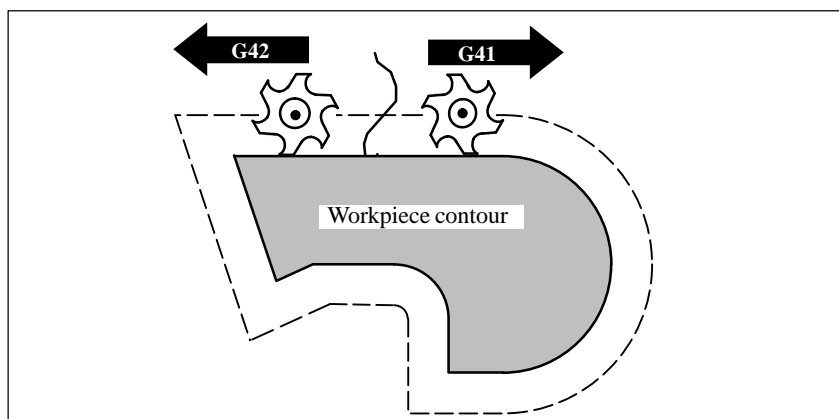


Fig. 8-36 Offset right/left of contour

Start compensation

The tool approaches the contour along a straight line and positions itself perpendicular to the path tangent at the starting point of the contour.

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

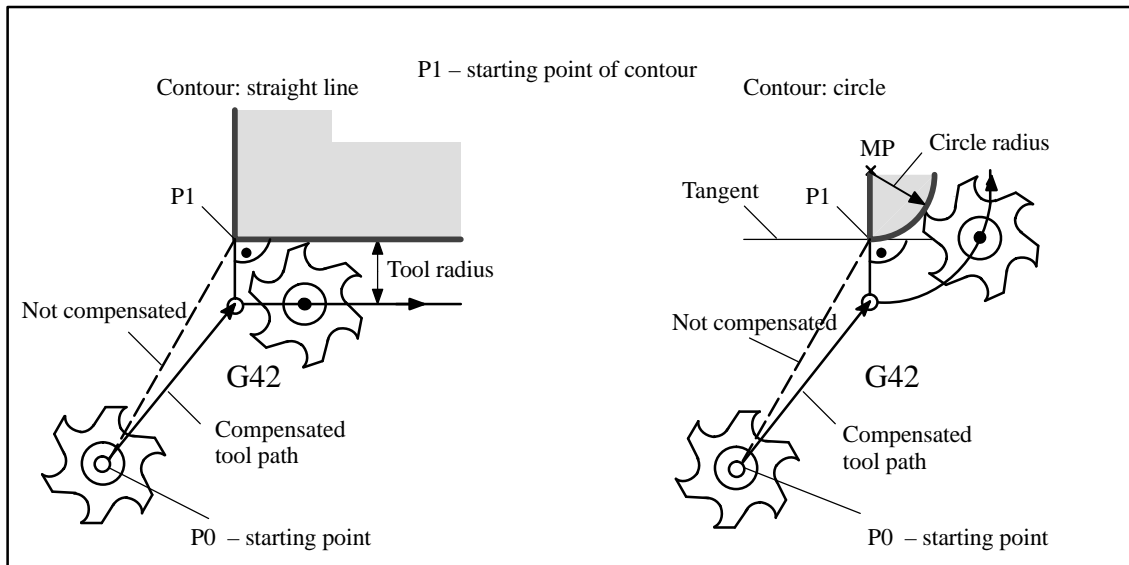


Fig. 8-37 Start of tool radius offset, example shows G42

Information

The block containing G41/G42 is generally followed by the first block with the workpiece contour. The contour definition may, however, be interrupted by an intermediate block that does not contain any information on the contour path in the plane, e.g. only M commands or infeed movements.

Programming example

```

N10 T...
N20 G17 D2 F300 ;Offset no. 2, feed 300 mm/min
N25 X... Y... ;starting point P0
N30 G1 G42 X... Y... ;Selection right of contour, P1
N31 X... Y... ;Initial contour, circle or straight line

```

After the offset has been selected, it is also possible to carry out blocks containing infeed movements or M commands.:

```

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

```

8.6.5 Behavior at corners: G450, G451

Functionality

The functions G450 and G451 can be used to set the response in the case of discontinuous transition from one contour element to another (behavior at corners) when G41/G42 is active. The control system itself detects inner and outer corners. The point at which the equidistant paths intersect is always approached in the case of inner corners.

Programming

```
G450      ;Transition circle
G451      ;Intersection
```

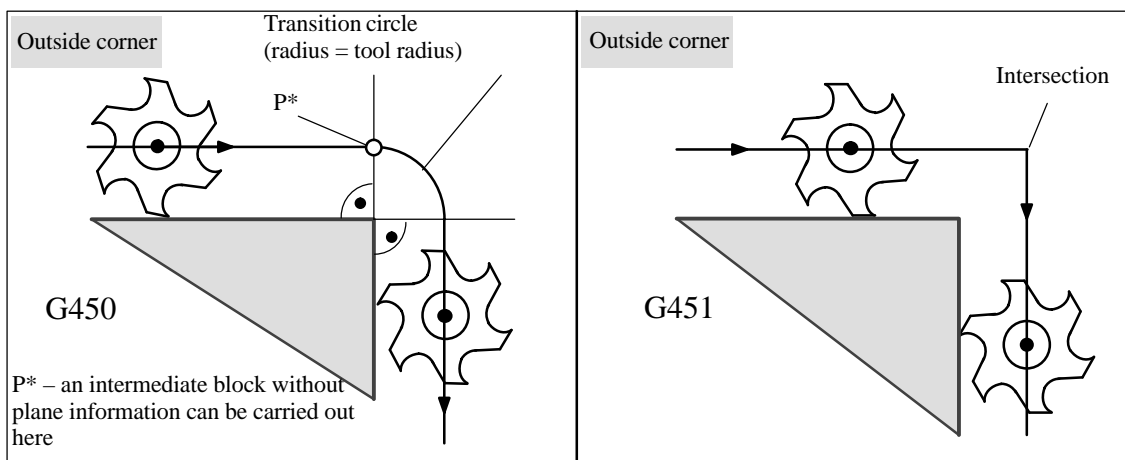


Fig. 8-38 Behavior at an outside corner

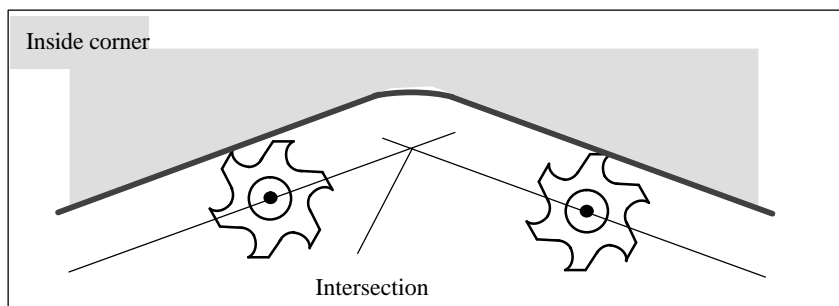


Fig. 8-39 Behavior at an inside corner

Transition circle G450

The tool center point traverses round the workpiece outer corner along an arc with the same radius as the tool radius. In data-processing terms, the transition circle belongs to the next block which contains traversing movements, e.g. with respect to the feed value.

Intersection G451

With function G451 (intersection of equidistant paths), the tool approaches the point at which the center point paths (circle or straight line) intersect.

When the intersection is active, acute contour angles can result in idle motions of the tool, depending on the tool radius.

The control system automatically changes to this block for transition circle when the angle value set in machine data is reached.

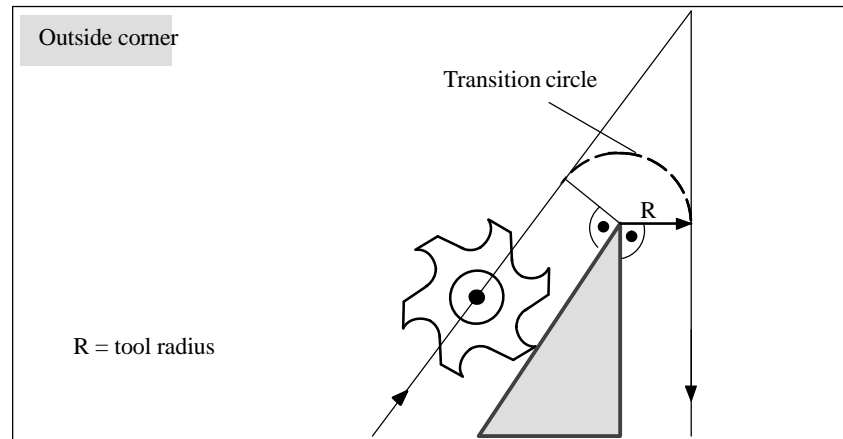


Fig. 8-40 Acute contour angle and change to transition circle

8.6.6 Tool radius compensation OFF: G40

Functionality

The compensation mode activated with G41/G42 is deactivated by G40. This G function is also the default position when the program starts.

The tool completes the **block prior to G40** in normal position (compensation vector perpendicular to tangent at end point), independently of the retraction angle.

The tool center point reaches the programmed end point in the block with G40.

Always select the end point of the G40 block such that collision-free traversing is ensured.

Programming

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

Note: The compensation mode can only in linear interpolation (G0, G1) be deactivated.

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

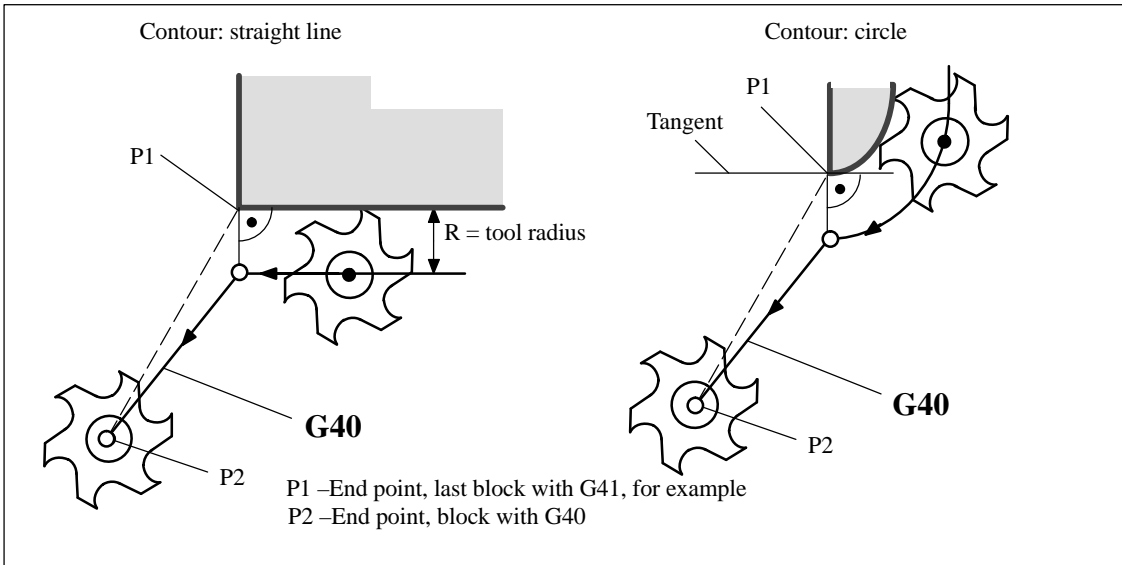


Fig. 8-41 Ending tool radius compensation

Programming example

```

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

8.6.7 Special cases of tool radius compensation

Change in compensation direction

The compensation direction G41 <-> G42 can be changed without inserting 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 start of compensation (normal position at starting point).

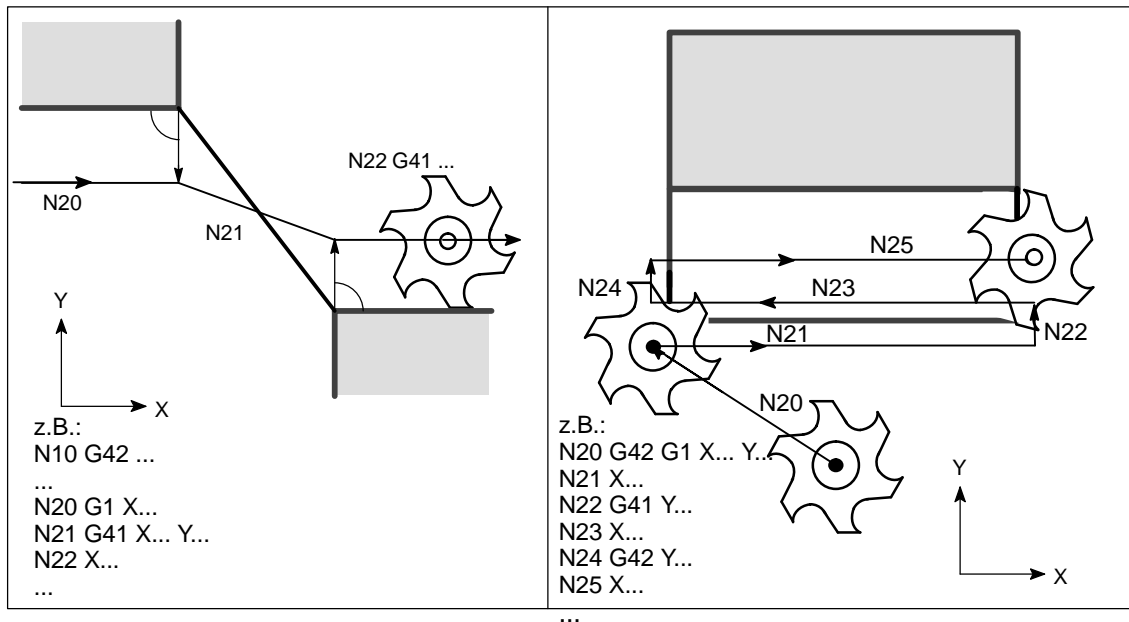


Fig. 8-42 Change in compensation direction

Repetition of G41, G41 or G42, G42

The same contour can be programmed again without inserting G40.

The last block prior to calling the new compensation ends with normal position of the compensation vector in the end point. The new compensation is carried out as a start of compensation (the same behavior as described for change in compensation direction).

Change in offset number D

The offset number D can be changed in compensation mode. In this case, an altered tool radius takes effect already in the beginning of the block containing the new D number. Its full change is only reached at the end of the block. The change is thus implemented continuously over the entire block. This also applies to circular interpolation.

Cancellation of compensation using M2

If compensation mode is canceled by means of M2 (end of program) without programming the command G40, the last block ends with the coordinates of the plane (G17 to G19) in normal position of the compensation vector. **No** axis movement is performed. The program ends with this tool position.

Critical machining operations

When programming machining operations, watch out for cases where the contour path at inner corners is smaller than the tool radius and, with two consecutive inner corners, smaller than the diameter.

This type of programming error must be avoided!

Check sequences of several blocks to make sure that the contour does not contain any "bottlenecks". When you carry out a test/dry run, use the largest available tool radius.

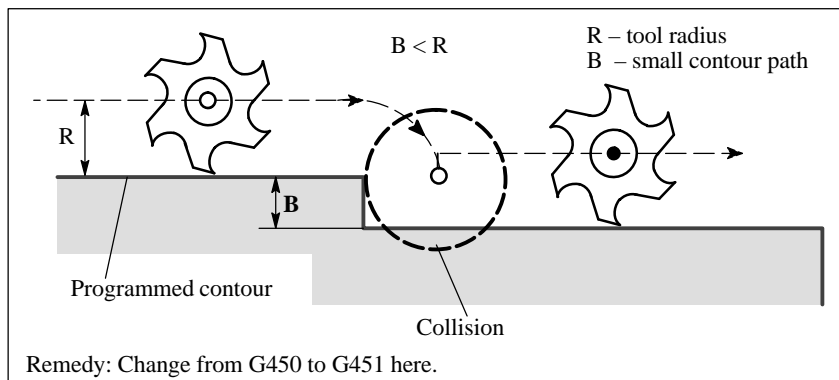


Fig. 8-43 Critical machining operation, example shows transition circle

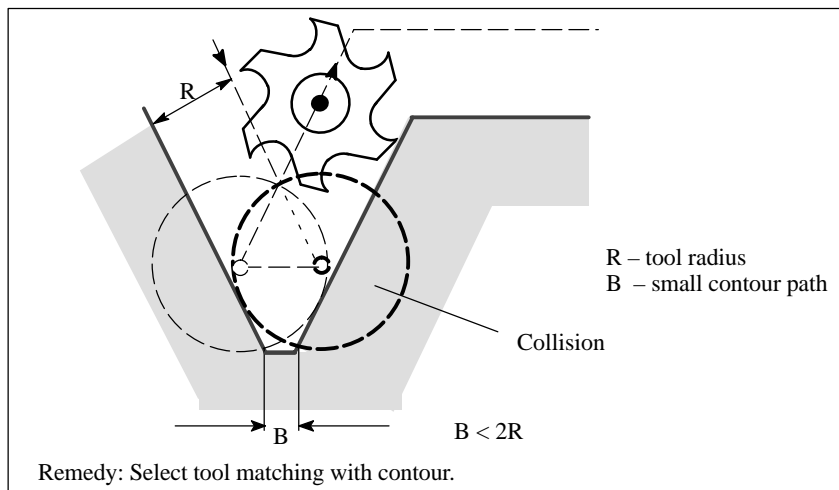


Fig. 8-44 Critical machining operation on inside corner/inside corner (example)

8.6.8 Example of tool radius compensation

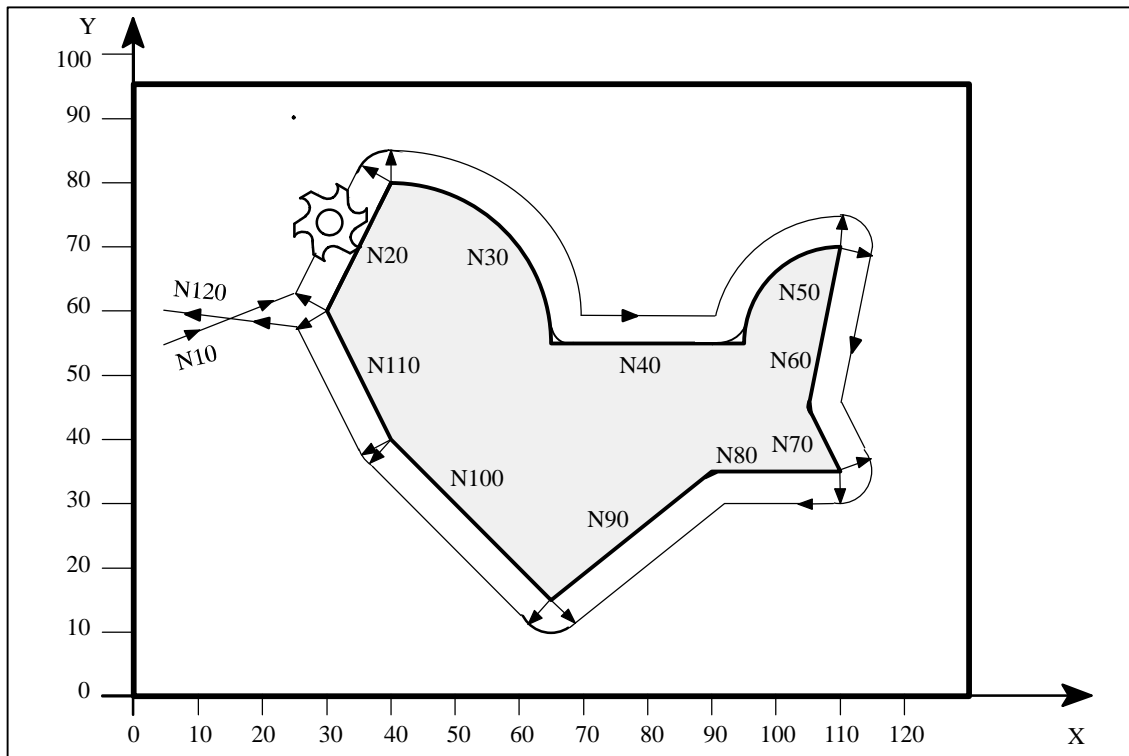


Fig. 8-45 Example of tool radius compensation

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 left of 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 ;End compensation mode
N130 G0 Z50 M2

```

8.7 Miscellaneous function M

Functionality

The miscellaneous function M can be used, for example, to initiate switching operations such as “Coolant ON/OFF”, among other tasks.

The control system manufacturer preassigns certain functions to a small number of the M functions. The others can be freely assigned to functions by the user.

A block may contain a maximum of 5 M functions.

Note:

An overview of the miscellaneous M functions as used and reserved in the control system is provided in Section 8.1.5 “Overview of Instructions”.

Programming

M...

Activation

Activation in blocks with axis movements:

If functions **M0, M1 and M2** are programmed in a block which includes axis movements, then they **take effect after the traversing movements** have been executed.

The functions **M3, M4 and M5** are transferred to the internal interface control **before the traversing movements**. The axis movements are not executed until the spindle has run up in M3 or M4. In the case of M5, however, the axis movements commence before the spindle has reached a standstill.

All the other M functions are transferred to the internal interface control **at the same time** as the traversing movements.

If you wish to program an M function specifically before or after an axis movement, then insert a separate block with the M function.

Remember: This block will interrupt G64 continuous path control mode and generate an exact stop!

Programming example

```
N10 S...
N20 X... M3           ;M function in block with axis movement
                       Spindle runs up before X axis movement
N180 M78 M67 M10 M12 M37
                       ;Max. 5 M functions in block
```

8.8 Arithmetic parameters R

Functionality

If you want an NC program in which you can vary the values to be processed, or if you simply need to compute arithmetic values, then you can use R (arithmetic) parameters. The control system will calculate or set the values you need when the program is executed.

An alternative method is to input the arithmetic parameter values directly. If the R parameters already have value settings, then they can be assigned in the program to other NC addresses that have variable values.

Programming

```
R0=...
  to
R249=...
(to R299=..., if there are no machining cycles)
```

Explanation

250 arithmetic parameters with the following classification are available:

R0 ... R99 – for free assignment

R100 ... R249 – transfer parameters for machining cycles.

R250 ... R299 – internal arithmetic parameters for machining cycles.

If you do not intend to use the machining cycles (see Section 9.2 “Machining Cycles”), then this range of arithmetic parameters are also available for your use.

Value assignment

You can assign values in the following range to the R parameters:

$\pm(0.000\ 0001 \dots 9999\ 9999)$
(8 decimal places and sign and decimal point).

The decimal point can be omitted for integer values. A positive sign can also be omitted.

Example:

R0=3.5678 R1=-37.3 R2=2 R3=-7 R4=-45678.1234

You can assign an extended numerical range using **exponential notation**:

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

The value of the exponent is typed after the characters **EX**. Maximum number of characters: 10 (including sign and decimal point).

Value range of EX: -300 to +300.

Example:

R0=-0.1EX-5 ;Meaning: R0 = -0,000 001

R1=1.874EX8 ;Meaning: R1 = 187 400 000

Note: Several assignments (including arithmetic expressions) can be programmed in one block.

Assignment to other addresses

You can obtain a flexible NC program by assigning arithmetic parameters or arithmetic expressions with R parameters to other NC addresses. Values, arithmetic expressions or R parameters can be assigned to any NC address with the **exception of addresses N, G and L**.

When making assignments of this kind, type the character "=" after the address character. Assignments with a negative sign are also permitted.

If you wish to make assignments to axis addresses (traversal instructions), then you must do so in a separate program block.

Example:

N10 G0 X=R2 ;Assignment to X axis

Arithmetic operations/functions

Operators/arithmetic functions must be programmed using the normal mathematical notation. Processing priorities are set by means of round brackets. Otherwise the "multiplication/division before addition/subtraction" rule applies. Degrees are specified for trigonometric functions.

Programming example: R parameter

N10 R1= R1+1 ;The new R1 is product of old R1 plus 1
 N20 R1=R2+R3 R4=R5-R6 R7=R8·R9 R10=R11/R12
 N30 R13=SIN(25.3) ;R13 is the sine of 25.3 degrees
 N40 R14=R1·R2+R3 ;"Multiplication/division before
 addition/subtraction" rule
 R14=(R1·R2)+R3
 N50 R14=R3+R2·R1 ;Result as for block N40
 N60 R15=SQRT(R1·R1+R2·R2)

;Meaning: $R15 = \sqrt{R1^2 + R2^2}$

Programming example: Assignment to axes

N10 G1 G91 X=R1 Z=R2 F300
 N20 Z=R3
 N30 X=-R4
 N40 Z=-R5
 ...

8.9 Program branches

8.9.1 Labels – destination for program branches

Functionality

Labels are used to mark blocks as the branch destination for branches in the program sequence.

Labels can be selected freely, but must have a minimum of 2 and a maximum of 8 letters or digits. However the **first two characters** must be **letters** or underscore characters.

Labels **end in a colon** in the block that is to act as a branch destination. They are always positioned at the beginning of the block. If the block also has a block number, then the label is positioned **after the number**.

Labels must be unique within the same program.

Programming example

```
N10 MARKE1: G1 X20           ;MARKE1 is label, branch destination
...
TR789: G0 X10 Z20           ;TR789 is label, branch destination
                          No block number
```

8.9.2 Unconditional program branches

Functionality

NC programs process the blocks they contain in the same order as they were typed by the programmer.

The processing sequence can be altered through the insertion of program branches.

The only possible branch destination is a block with label. This block must be included in the program.

An unconditional branch instruction must be programmed in a separate block.

Programming

```
GOTOF Label           ;Branch forwards
GOTOB Label           ;Branch backwards
```

STL	Explanation
GOTOF	Branch direction forwards (towards last block in program)
GOTOB	Branch direction backwards (towards first block in program)
Label	Selected character string for label

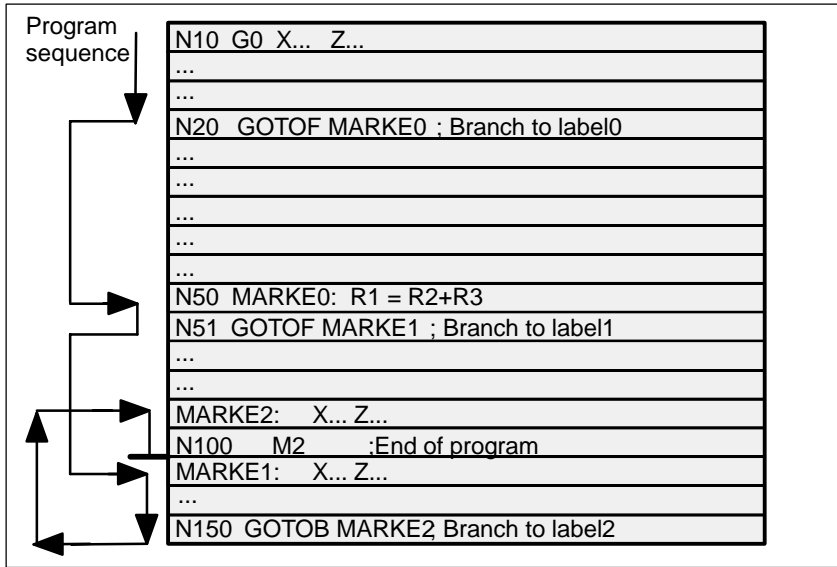


Fig. 8-46 Example of unconditional branches

8.9.3 Conditional branches

Functionality

Branch conditions are formulated after the **IF instruction**. If the branch condition is fulfilled (**value not equal to zero**), then the program branches. The branch destination can only be a block with corresponding label. This block must be contained within the program.

Conditional branch instructions must be programmed in a separate block. Several conditional branch instructions can be programmed in the same block.

You can reduce program processing times significantly by using conditional program branches.

Programming

IF *condition* GOTOF *Label* ;Branch forwards
 IF *condition* GOTOB *Label* ;Branch backwards

STL	Explanation
GOTOF	Branch direction forwards (towards last block in program)
GOTOB	Branch direction backwards (towards first block in program)
Label	Selected character string for label
IF	Introduction of branch condition
Condition	Arithmetic parameter, arithmetic expression in comparison for formulation of 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 are used to formulate branch conditions. Arithmetic expressions can also be compared.

The result of comparison operations is either "fulfilled" or "not fulfilled". "Not fulfilled" is equivalent to a value of 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)2

```

Programming example

```

N10 IF R1 GOTOF MARKE1    ;If R1 is not zero,
                          branch to block with MARKE1
...
N100 IF R1>1 GOTOF MARKE2 ;If R1 is greater than 1,
                          branch to block with MARKE2
...
N1000 IF R45==R7+1 GOTOB MARKE3
                          ;If R45 is equal to R7 plus 1,
                          branch to block with MARKE3

```

Several conditional branches in block:

```

...
N20 IF R1==1 GOTOB MA1 IF R1==2 GOTOF MA2 ...
...

```

Note: The program branches at the first fulfilled condition.

8.9.4 Example of program with branches

Objective of program

Approach points on an circle segment:

Let us assume the following values:

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 point in Z:	50 mm	in R5
Position of circle center point in X:	20 mm	in R6

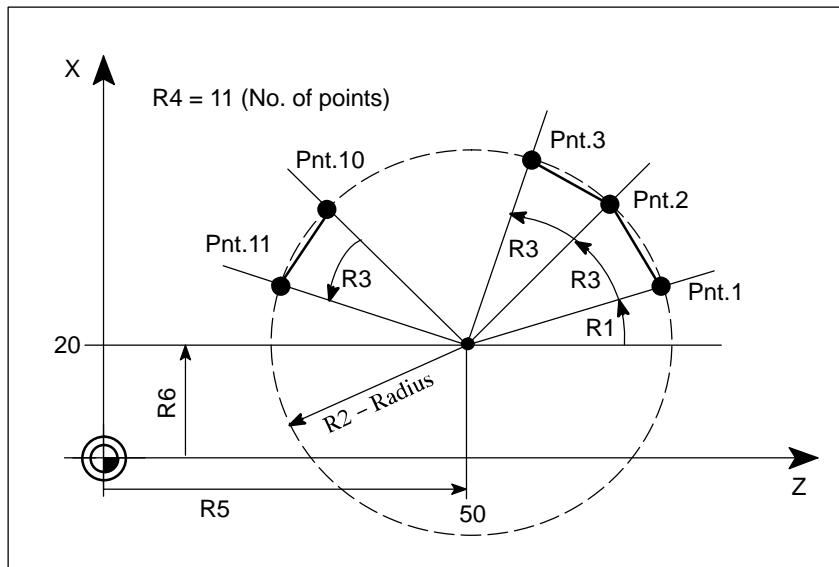


Fig. 8-47 Approaching points along a circle segment

Programming example

```

N10 R1=30 R2=32 R3=10 R4=11 R5=50 R6=20
    ;Assignment of start values
N20 MA1: G0 Z=R2 *COS (R1)+R5 X=R2*SIN(R1)+R6
    ;Computation and assignment to axis addresses
N30 R1=R1+R3 R4= R4-1
N40 IF R4 > 0 GOTOB MA1
N50 M2
    
```

Explanation

The initial conditions are assigned to the appropriate arithmetic parameters in block N10. The coordinates in X and Z are calculated in N20 and processed.

In N30, R1 is increased by the angle R3 and R4 is decremented by 1. If R4 > 0, N20 is processed again. Otherwise the program continues with N50 and end of program.

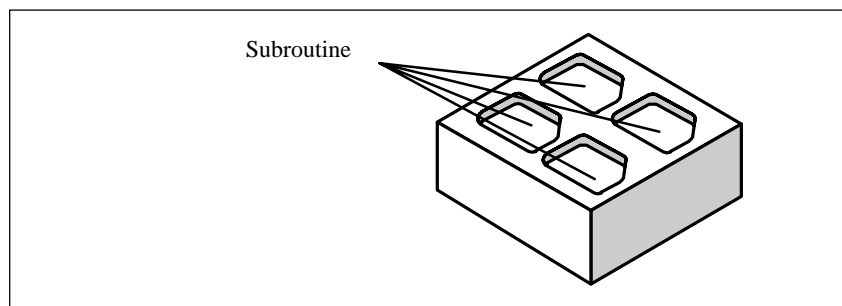
8.10 Subroutine technique

Application

There is no essential difference between a main program and a subroutine.

Subroutines contain frequently recurring machining sequences, for example, certain contour shapes. This type of subroutine is called at the appropriate locations in the main program and then processed.

One type of subroutine is the **machining cycle**. Machining cycles contain generally applicable machining operations (e.g. drilling, tapping, groove cutting, etc.). By supplying these cycles with values by means of the arithmetic parameters provided, you can adapt the program to your specific application (see Section "Machining Cycles").



Structure

Subroutines are structured in exactly the same way as main programs (see Section "Program structure").

M2 (end of program)

is programmed in the last block of the subroutine sequence in exactly the same way as for main programs. In this case, program end means a return to the program level that called the subroutine.

End of program

The M2 end-of-program instruction can be substituted by the end instruction **RET** in subroutines.

RET must be programmed in a separate block.

An RET instruction must be used when it is necessary to avoid an interruption in continuous path control mode G64 when the program branches back to main program level from the subroutine. If an M2 instruction is programmed, G64 mode is interrupted and an exact stop generated.

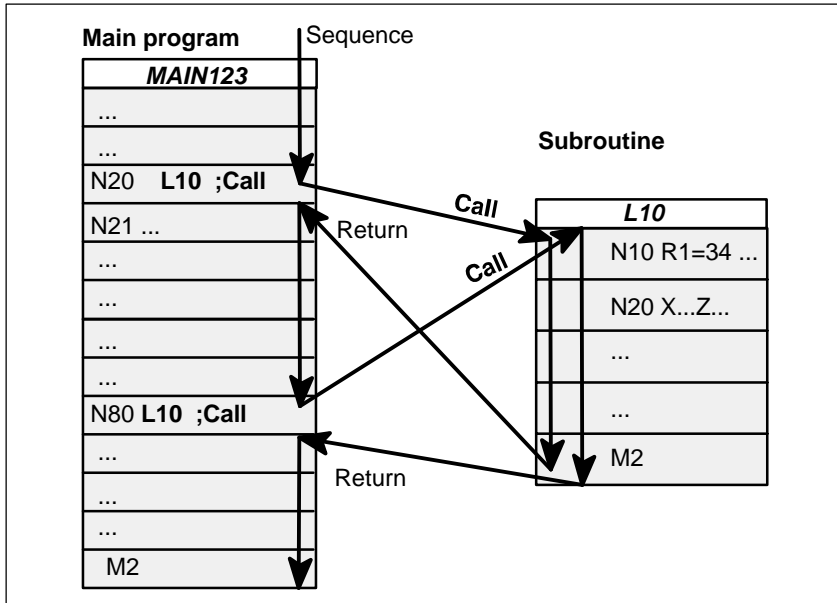


Fig. 8-48 Example of program sequence in which subroutine is called twice

Subroutine name

A subroutine is given its own specific name so that it can be selected from all the others. The name can be **chosen freely** subject to the following conditions when the subroutine is generated:

- The first two characters must be letters
- The others may be letters, digits or underscore
- Maximum of 8 characters in total
- No dashes (see Section "Character set")

The same rules apply as for main program names.

Example: **LFRAME7**

There is the additional option of using the address word **L...** for subroutines. This value may have 7 decimal places (integers only).

Please note: Leading zeros are interpreted as distinguishing digits in the L address.

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

These are 3 different subroutines!

Subroutine call

Subroutines are called by their name in a program (main program or subroutine). These calls must be programmed in separate blocks.

Example:

```
N10 L785 ;Call subroutine L785
N20 LFRAME7 ;Call subroutine LFRAME7
```

Program repetition P...

If a subroutine must be repeated several times in succession, then enter the number of runs under **address P** after the subroutine name in the block containing the subroutine call. A maximum of **9999 runs** can be programmed (P1 ... P9999).

Example:

N10 L785 P3 ;Call subroutine L785, 3 runs

Nesting depth

It is not only possible to call subroutines in main programs, but also in other subroutines. There is a total of **4 program levels** (including the main program level) available for programming this type of nested call.

Note: If you are working with machining cycles, please remember that these also need one of the four program levels.

The same applies to the arithmetic (R) parameters. Make sure that the arithmetic parameters you are using in the upper program levels do not change to different settings in lower levels.

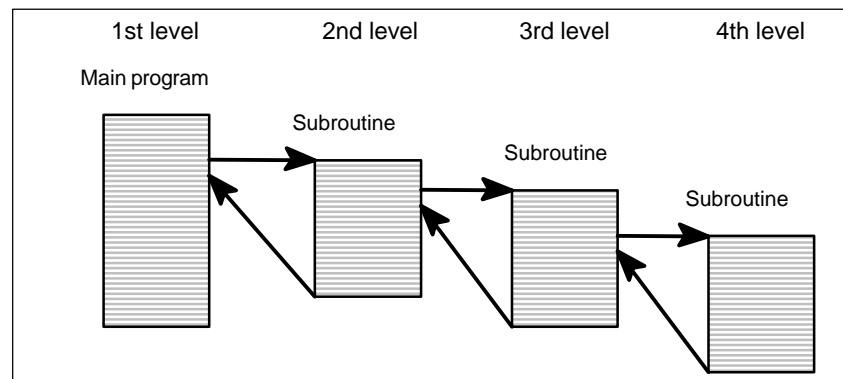


Fig. 8-49 Sequence with four program levels

Information

It is possible to change modal G functions, e.g. G90 → G91, in subroutines. Make sure that all modal functions are set in the way you require when the program branches back to the level on which the subroutine was called.

The same applies to the arithmetic parameters R. Make sure that the values of your arithmetic parameters used in the upper program levels are not changed unintentionally.

Cycles

Preliminary remark

Cycles are process-related subroutines that support general implementation of specific machining processes, such as drilling, stock removal or thread cutting. The cycles are adapted to the specific problem in hand by means of supply parameters.

Standard cycles for drilling and milling applications are provided in the system.

9.1 General information about standard cycles

9.1.1 Overview of cycles

LCYC82	Drilling, spot facing
LCYC83	Deep hole drilling
LCYC840	Tapping with compensating chuck
LCYC84	Tapping without compensating chuck
LCYC85	Boring_1
LCYC60	Row of holes
LCYC61	Row of holes
LCYC75	Cutting square pockets, slots and circular pockets

Defining parameters

The arithmetic parameters from R100 to R149 are used as supply parameters for the cycles.

Before a cycle is called, values must be assigned to its transfer parameters. Any parameters not needed must be loaded with zero. The values of these transfer parameters are unchanged after the cycle has been executed.

Arithmetic parameters

The cycles use the parameters R250 to R299 as internal arithmetic parameters. These are deleted when calling the cycles.

Call and return conditions

The drilling cycles are programmed independently of the particular axis names. The drilling position must be approached prior to calling the cycle in the higher-level program. The required values for feed, spindle speed and direction of rotation of the spindle must be programmed in the part program, if there are no supply parameters in the drilling cycle.

G0 G90 G40 are always effective at the end of a cycle.

Recompilation of cycles

The cycle can only be recompiled if the set of parameters stands immediately before the cycle call. The parameters may not be separated by NC statements or comments.

Plane definition

All drilling and milling cycles assume that the current workpiece coordinate system in which machining is to be performed is defined by selecting a plane G17, G18 or G19 and activating a programmed frame (zero offset, rotation). The drilling axis is always the 3rd axis of this system.

Prior to the call, a tool with tool offset of this plane must be active. This remains active even after the cycle has been completed.

9.1.2 Error messages and error handling in cycles

Error handling in cycles

Alarms with numbers between 61000 and 62999 are generated in the cycles. This number range is subdivided again into alarm reactions and reset criteria.

Table 9-1 Alarm numbers, reset criteria, alarm reactions

Alarm Number	Reaction	Program Continued by
61000...61999	Block preparation in NC is aborted	NC RESET
62000...62999	Block preparation is interrupted, can be continued by NC Start after alarm reset	Reset key

The error text displayed at the same time as the alarm number provides further details about the error cause.

Overview of cycle alarms

The following Table provides an overview of errors which can occur in cycles, of the location of their origin and guidance on how to eliminate them.

Table 9-2 Cycle alarms

Alarm Number	Alarm Text	Source	Remedial Action
61000	No tool offset active	LCYC75	Program a tool with tool offset in the calling program.
61001	Thread lead incorrectly defined	LCYC84, LCYC840	Check parameter R106 (R106=0).
61002	Machining type incorrectly defined	LCYC75	Check parameter R127 (only values 1 or 2 permitted).
61003	3rd geometry axis missing	LCYC82, 83, 84, 840, 84, 85	Check machine configuration (connect 3rd geometry axis).
61004	Geometry axis incorrectly configured	LCYC60,61,75	Check machine configuration (one geometry axis missing).
61101	Reference plane incorrectly defined	LCYC82, 83, 84, 840, 85, 75	Check parameters R101, R103, R104 – R103 = R104, or R103 is not between R101 and R104.
61102	No spindle direction defined	LCYC840	Value in parameter R107 is greater than 4 or less than 3.
61103	Number of drill holes is zero	LCYC60, 61	Parameter R119=0
61105	Cutter radius too much	LCYC75	The cutter diameter in the tool offset memory is greater than the pocket or slot width (R119 or R118) – either use a smaller cutter or alter the pocket width.
61106	Number or distance of circle elements too much	LCYC61	R119 or R120 incorrectly parameterized; allocation of circle elements on a full circle is not possible
61107	First drilling depth incorrectly defined	LCYC83	First drilling depth is opposite to the entire drilling depth
61108	Cutting direction incorrectly defined	LCYC75	Incorrect parameter R126 – greater than 3 or less than 2.
61109	Parameterized cycle number incorrectly defined	LCYC60, 61	Incorrect parameter R115 – the cycle number programmed there does not exist
62101	Finishing allowance reduced to possible value	LCYC75	The cutter diameter is greater than the pockets (slot width minus finishing allowance); if desired either use a smaller cutter or reduce the finishing allowance or enlarge the pocket slot width.

9.2 Drilling cycles

This Section describes

- the general preconditions for calling SIEMENS standard cycles;
- the parameters for each cycle;
- the motional sequence in the cycle;
- the call of the machining cycles using an example.

9.2.1 Drilling, spot facing – LCYC82

Function

The tool drills with the spindle speed and feedrate programmed down to the entered final depth. When the final drilling depth is reached, a dwell time can be programmed. The drill is retracted from the drill hole at rapid traverse rate.

Call

LCYC82

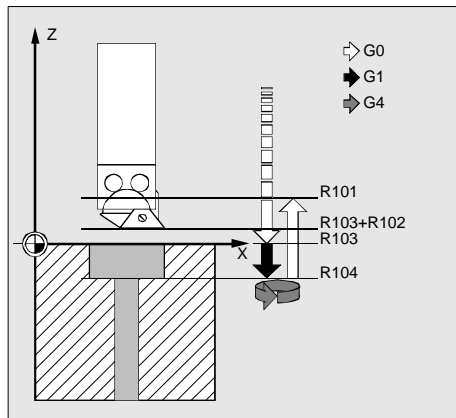


Fig. 9-1 Motional sequence and parameters in the cycle

Precondition

The spindle speed and the direction of rotation, as well as the feed of the drilling axis must be defined in the higher-level program.

The drilling position must be approached before calling the cycle in the higher-level program.

The required tool with tool offset must be selected before calling the cycle.

Parameters

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time in seconds

Information

- R101** The retract plane determines the position of the drilling axis at the end of the cycle.
- R102** The safety clearance acts on the reference plane, i.e. the reference plane is shifted forward by an amount corresponding to the safety clearance.
The direction in which the safety clearance acts is automatically determined by the cycle.
- R103** The starting point of the drill hole shown in the drawing is programmed under the reference plane parameter.
- R104** The drilling depth is always programmed as an absolute value with refer to workpiece zero.
- R105** The dwell time at drilling depth (chip breakage) is programmed in seconds under R105.

Motional sequence

Position reached prior to beginning of cycle:
last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0.
2. Traverse to final drilling depth with G1 and the feedrate programmed in the higher-level program.
3. Execute dwell time to final drilling depth.
4. Retract to retract plane with G0.

Example**Drilling – counter boring**

The program produces a 27 mm deep drill hole in the position X24 Y15 in the XY plane using the cycle LCYC82. The dwell time is 2 s, and the safety clearance in the drilling axis (here: Z) amounts to 4 mm. On completion of the cycle, the tool stands on X24 Y15 Z110.

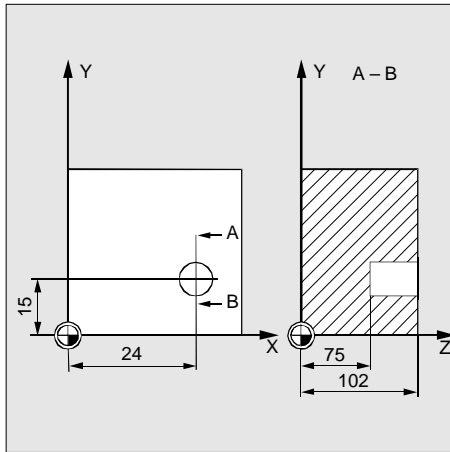


Fig. 9-2 Example drawing

```

N10 G0 G17 G90 F500 T2 D1 S500 M4 ; Define technology values
N20 X24 Y15 ; Approach drilling position
N30 R101=110 R102=4 R103=102 R104=75 ; Supply parameters
N35 R105=2 ; Supply parameters
N40 LCYC82 ; Call cycle
N50 M2 ; End of program

```

9.2.2 Deep hole drilling – LCYC83**Function**

The deep-hole drilling cycle produces center holes down to the final drilling depth by repeated, step-by-step deep infeed whose maximum amount can be parameterized. The drill can be retracted either to the reference plane for swarf removal after each infeed depth or by 1 mm in each case for chip breakage.

Call

LCYC83

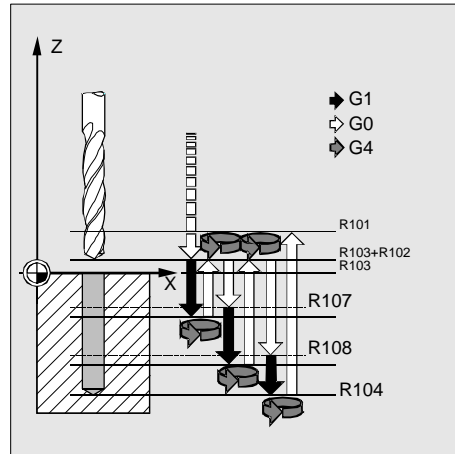


Fig. 9-3 Motional sequence and parameters in the cycle

Precondition

The spindle speed and the direction of rotation must be defined in the higher-level program.

The drilling position must be approached before calling the cycle in the higher-level program.

Before calling the cycle, a tool offset for the drill must be selected.

Parameters

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance, enter without sign
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time to drilling depth (chip breakage)
R107	Feed for drilling
R108	Feed for first drilling depth
R109	Dwell time at starting point and for swarf removal
R110	First drilling depth (absolute)
R111	Absolute degression, enter without sign
R127	Machining type: Chip breakage = 0 Swarf removal = 1

Information

R101 The retract plane determines the position of the drilling axis at the end of the cycle.

The cycle is programmed on the assumption that the retract plane positioned in front of the reference plane, i.e. its distance to the final depth is greater.

R102	The safety clearance acts on the reference plane, i.e. the reference plane is shifted forward by an amount corresponding to the safety clearance. The direction in which the safety clearance acts is automatically determined by the cycle.
R103	The starting point of the drill hole shown in the drawing is programmed under the reference plane parameter.
R104	The drilling depth is always programmed as an absolute value regardless of how G90/91 is set prior to cycle call.
R105	The dwell time at drilling depth (chip breakage) is programmed in seconds under R105.
R107, R108	The feed for the first drilling stroke (under R108) and for all subsequent drilling strokes (under R107) are programmed via these parameters.
R109	A dwell time at the starting point can be programmed in seconds under parameter R109. The dwell time at the starting point is executed only for the "with swarf removal" variant.
R110	Parameter R110 determines the depth of the first drilling stroke.
R111	Parameter R111 for the degression value determines the amount by which the current drilling depth is reduced with subsequent drilling strokes. The second drilling depth corresponds to the stroke of the first drilling depth minus the absolute degression value provided that this value is greater than the programmed absolute degression value. Otherwise, the second drilling depth also corresponds to the absolute degression value. The next drilling strokes correspond to the absolute degression value provided that the remaining degression depth is still greater than twice the absolute degression value. The remainder is then distributed evenly between the last two drilling strokes. If the value for the first drilling depth is in opposition to the total drilling depth, the error message 61107 "First drilling depth incorrectly defined" is displayed, and the cycle is not executed.
R127	Value 0: The drill travels 1 mm clear for chip breakage after it has reached each drilling depth. Value 1: The drill travels to the reference plane, which is shifted forward by an amount corresponding to the safety clearance for swarf removal after each drilling depth.

Motional sequence

Position reached prior to beginning of cycle:
last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0.
2. Traverse to first drilling depth with G1; the feedrate results from the feedrate programmed prior to cycle call after it has been computed with the setting in parameter R109 (feedrate factor).
Execute dwell time at drilling depth (parameter R105).

With chip breakage selected:

Retract by 1 mm from the current drilling depth with G1 for chip breakage.

With swarf removal selected:

Retract for swarf removal to reference plane shifted forward by an amount corresponding to the safety clearance with G0 for swarf removal, executing the dwell time at starting point (parameter R106),

approach last drilling depth minus clearance distance calculated in the cycle using G0,

3. Traverse to next drilling depth with G1 and the programmed feed; this motional sequence is continued as long as the final drilling depth is reached.
4. Retract to retract plane with G0.

Example: Deep-hole drillin

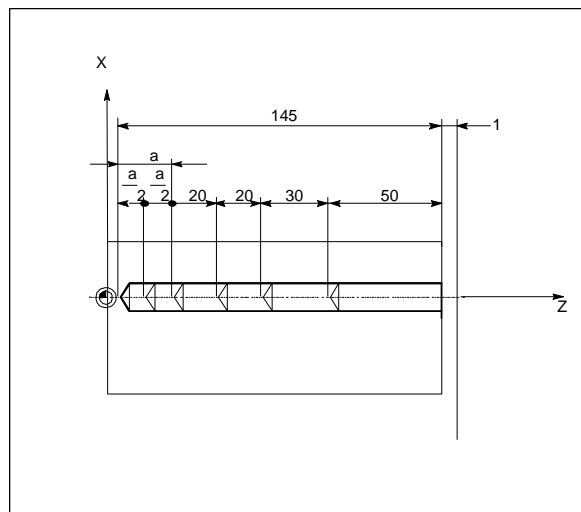


Fig. 9-4 Example drawing

Deep-hole drilling

```

;This program executes the cycle LCYC83 ;atposition X70.
N100 G0 G18 G90 T4 S500 M3           ;Define technology values
N110 Z155
N120 X70                               ;Approach first drilling position
R101=155 R102=1 R103=150
R104=5 R105=0 R109=0 R110=100        ;Parameter supply
R111=20 R107=500 R127=1 R108=400
N140 LCYC83                            ;1st call of cycle
N199 M2

```

9.2.3 Tapping without compensating chuck – LCYC84

Function

The tool drills with the programmed spindle speed and spindle direction down to the entered thread depth. Compared with the LCY840, this cycle is faster and more exactly. Nevertheless, a compensating chuck should be used for manufacture. The feed of the drilling axis results from the spindle speed. The direction of rotation is reversed in the cycle automatically. The retract can be carried out with a separate speed.

Call

LCYC84

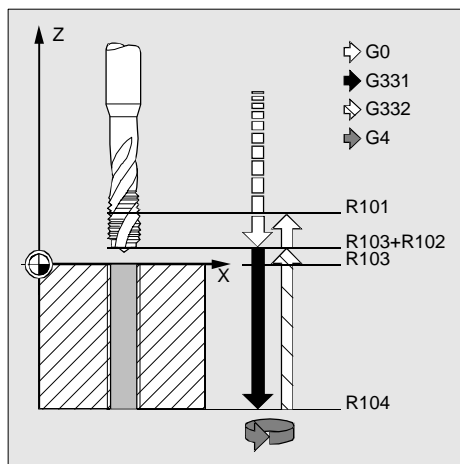


Fig. 9-5

Precondition

This cycle can only be used if the spindle is technologically able to switch over to **position-controlled** spindle operation (**actual-value encoder**). The cycle does not check whether the actual-value encoder for the spindle really exists.

The drilling position must be approached before calling the cycle in the higher-level program.

The required tool with tool offset must be selected before calling the cycle.

Depending on the spindle machine data settings and the accuracy of the drives, a compensating chuck should be used.

Parameters

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time at thread depth in seconds
R106	Thread lead as value Value range: 0.001 2000.000 mm -0.001 -2000.000 mm
R112	Speed for tapping
R113	Speed for retraction

Information

Parameters R101 – R105 see LCYC82

R106 Distance from one thread to the next thread as numerical value. The sign defines the direction of rotation of tapping. A positive sign means tapping in clockwise direction (as M3), and a negative sign tapping in counterclockwise direction (as M4).

R112 Parameter R112 contains the spindle speed for the tapping block.

R113 R113 is used to program the spindle speed for the retract movement. If this value is zero, the retract is carried out with the spindle speed programmed under R112.

Motional sequence

Position reached prior to beginning of cycle:
last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0.
2. Spindle stop at 0 degree and switching over the spindle to axis operation.
3. Tapping to final drilling depth with G331 and the speed programmed under R112. The direction of rotation results from the sign of the thread lead (R106).
4. Dwell time at thread depth
5. Retract to reference plane shifted forward by an amount corresponding to the safety clearance using G332 and the speed programmed under R113.
6. Retract to retract level with G0; the axis mode of the spindle is canceled.

Example

A thread is tapped without compensating chuck in the position X30 Y35 in the XY plane; the Z axis is the drilling axis. No dwell time is programmed. A negative lead is programmed, i.e. counterclockwise direction.

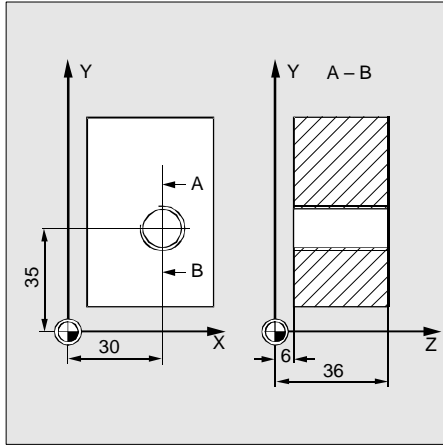


Fig. 9-6 Example drawing

```

N10 G0 G90 G17 T4 D4 ; Define technology value
N20 X30 Y35 Z40 ; Approach drilling position
N30 R101=40 R102=2 R103=36 R104=6 R105=0 ; Load parameters
N40 R106=-0.5 R112=100 R113=500 ; Load parameters
N50 LCYC84 ; Call cycle
N60 M2 ; End of program

```

9.2.4 Tapping with compensating chuck – LCYC840

Function

The tool drills with the programmed spindle speed and direction of rotation down to the entered thread depth. The feed of the drilling axis results from the spindle speed. This cycle can be used for tapping with compensating chuck and spindle actual-value encoder. The direction of rotation is automatically reversed in the cycle. The retract can be carried out at a separate speed.

Call

LCYC840

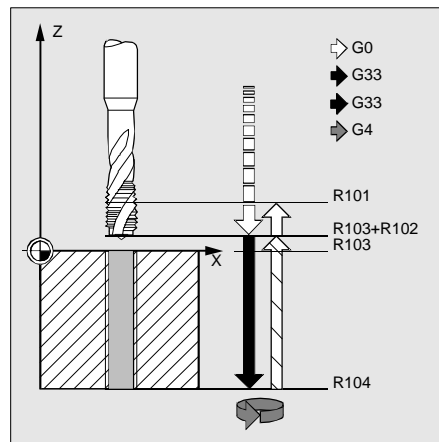


Fig. 9-7

Precondition

This cycle can only be used with a **speed-controlled** spindle with **position encoder**. The cycle does not check whether the actual-value encoder for the spindle really exists.

The spindle speed and the direction of rotation must be defined in the higher-level program.

The drilling position must be approached before calling the cycle in the higher-level program.

The required tool with tool offset must be selected before calling the cycle.

Parameters

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R106	Thread lead as value value range: 0.001 2000.000 mm
R126	Direction of rotation of spindle for tapping Value range: 3 (for M3), 4 (for M4)

Information

R101–R104 See LCYC84

R106 Thread lead as value

R126 The tapping block is executed with the direction of rotation of spindle programmed under R126. The direction of rotation is automatically reversed in the cycle.

Motional sequence

Position reached prior to beginning of cycle:

– last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0
2. Tapping down to final drilling depth with G33
3. Retract to reference plane shifted forward by an amount corresponding to the safety clearance with G33
4. Retract to retract plane with G0

Example

This program is used for tapping in the position X35 Y35 in the XY plane; the Z axis is the drilling axis. The parameter for the direction of rotation R126 must be parameterized. A compensating chuck must be used for machining. The spindle speed is defined in the higher-level program.

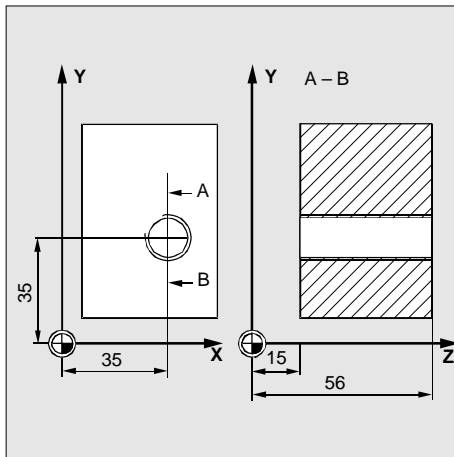


Fig. 9-8 Example drawing

N10 G0 G17 G90 S300 M3 D1 T1	; Define technology values
N20 X35 Y35 Z60	; Approach drilling position
N30 R101=60 R102=2 R103=56 R104=15 R105=1	; Parameter supply
N40 R106=0.5 R126=3	; Parameter supply
N40 LCYC840	; Cycle call
N50 M2	; End of program

9.2.5 Boring – LCYC85

Function

The tool drills with the spindle speed and feedrate programmed down to the entered final drilling depth. When the final drilling depth is reached, a dwell time can be programmed. The approach and retract movements are carried out with the feedrates programmed under the respective parameters.

Call

LCYC85

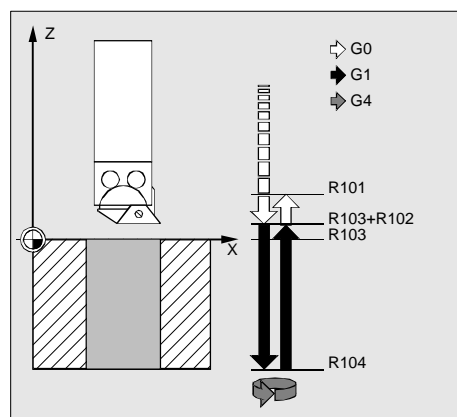


Fig. 9-9 Motion sequence and parameters of the cycle

Precondition

The spindle speed and the direction of rotation must be defined in the higher-level program.
The drilling position must be approached before calling the cycle in the higher-level program.
Before calling the cycle, the respective tool with tool offset must be selected.

Parameters

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time at drilling depth in seconds
R107	Feed for drilling
R108	Feed when retracting from drill hole

Information

Parameters R101 – R105	see LCYC82
R107	The feed value defined here acts for drilling.
R108	The feed value entered under R108 acts for retracting from the drill hole.

Motional sequence

Position reached prior to beginning of cycle:
last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0
2. Traverse to final drilling depth with G1 and the feed programmed under parameter R106.
3. Execute dwell time at final drilling depth.
4. Retract to reference plane shifted forward by an amount corresponding to the safety clearance with G1 and the retract feed programmed under R108.

Example

The cycle LCYC85 is called in Z70 and X50 in the ZX plane. The Y axis is the drilling axis. No dwell time is programmed. The workpiece upper edge is at Y=102.

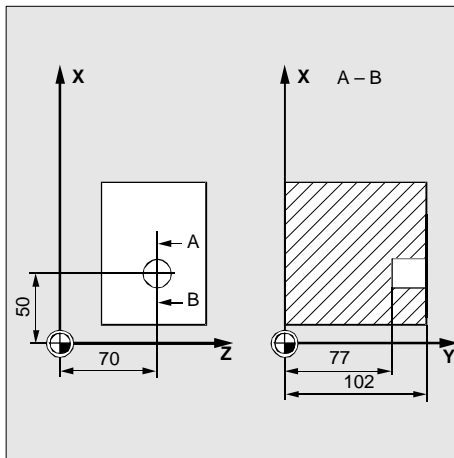


Fig. 9-10 Example drawing

N10 G0 G90 G18 F1000 S500 M3 T1 D1	; Define technology values
N20 Z70 X50 Y105	; Approach drilling position
N30 R101=105 R102=2 R103=102 R104=77	; Define parameters
N35 R105=0 R107=200 R108=400	; Define parameters
N40 LCYC85	; Call drilling cycle
N50 M2	; End of program

9.3 Drilling patterns

The LCYC60 and LCYC61 cycles can be used to produce drill holes and threads with a certain geometry whereby existing drilling and thread cutting cycles can be used.

9.3.1 Drilling a row of holes – LCYC60

Function

This cycle can be used to produce a certain number of drill or tapped holes arranged along a straight line. The type of the drill or tapped hole is defined by means of a parameter.

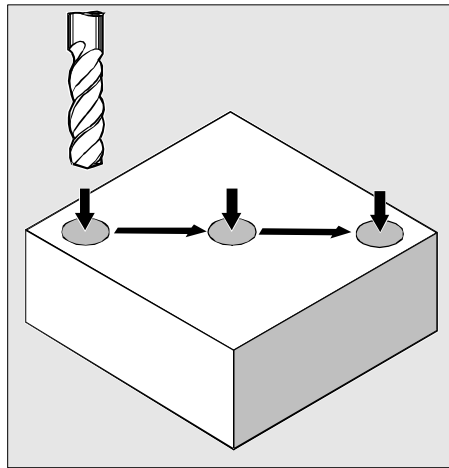


Fig. 9-11 Motional sequence

Call

LCYC60

Precondition

The spindle speed and the direction of rotation, as well as the feed along the drilling axis must be programmed depending on the parameterized drilling or tapping cycle in the higher-level program.

The chosen drilling or tapping cycle must also be parameterized before calling the drilling pattern cycle.

The required tool with tool offset must be selected before calling the cycle.

Parameters

Parameter	Meaning, Value Range
R115	Number of drilling or tapping cycle Values: 82 (LCYC82), 83 (LCYC83), 84 (LCYC84), 840 (LCYC840), 85 (LCYC85)
R116	Abscissae reference point
R117	Ordinate reference point
R118	Distance from first hole to reference point
R119	Number of holes
R120	Angle to the position of the row of holes in the plane
R121	Distance between the holes

Information

- R115 Use this parameter to determine the number of the drilling or tapping cycle with which the drill holes / tapped holes is to be executed.
- R116/R117 A point along the straight line of the row of holes is programmed, which is to be considered as the reference point to determine the distances between the holes. The distance to the first hole (R120) is specified from this point.
- R118 This parameter is used to program the distance from the first hole with refer to the reference point specified in R116/R117.
- R119 This parameter defines the number of drill/tapped holes.
- R120 This parameter is used to program the angle of the straight line to the ab scissa.
- R121 This parameter specifies the distance between the drill/tapped holes.

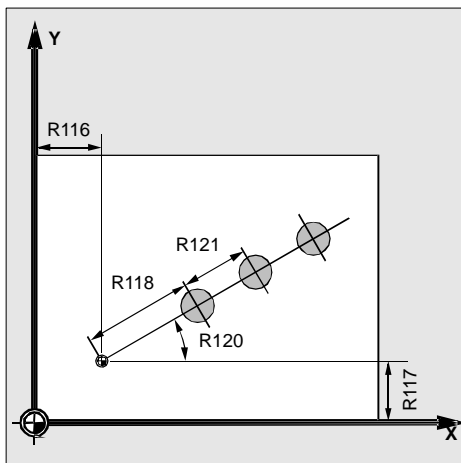


Fig. 9-12

Motional sequence

Starting position:

Any position from which the first drilling position can be approached without collision.

The cycle approaches the first drilling position and produces the hole using the cycle programmed under R115. All further drilling positions are approached with rapid traverse and drilled with the cycle parameterized.

Example

Row of holes

This program can be used to produce tapped holes arranged along a line in X direction in the ZX plane. The starting point is Z30 X20, with the first hole located 20 mm away from this point. The distance between the individual holes is also 20 mm. First, cycle LCYC83 is used to produce the drill hole, and cycle LCYC84 is then used for tapping (without compensating chuck) with positive lead (clockwise rotation of spindle). The holes have a depth of 80 mm.

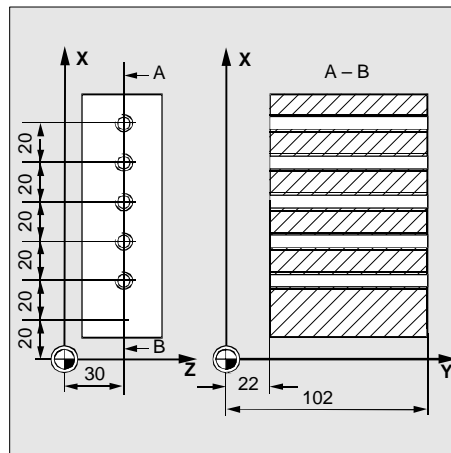


Fig. 9-13 Example drawing

```

N10 G0 G18 G90 S500 M3 T1 D1 ; Define technology values
N20 X50 Z50 Y110 ; Approach starting position
N30 R101=105 R102=2 R103=102 R104=22, ; Parameterize drilling cycle
N40 R107=100 R108=50 R109=1 ; Parameterize drilling cycle
N50 R110=90 R111=20 R127=1 ; Parameterize drilling cycle
N60 R115=83 R116=30 R117=20 R119=0 R120=20 R121=20 ; Parameterize cycle for row of holes
N70 LCYC60 ; Call cycle for row of holes
N80 ..... ; Tool change
N90 R106=0.5 R112=100 R113=500 ; Parameterize tapping cycle
; (program only parameters that
; changes compared with the drilling
; cycle)
N100 R115=84 ; Parameterize cycle for row of holes
; (R116 – R121 correspond to 1st
; call)
N110 LCYC60 ; Cycle for row of holes
N120 M2

```

Grid of holes

This program can be used to machine a grid of holes consisting of 5 lines with 5 holes each located in the XY plane, with a distance of 10 mm between them. The starting point of the grid of holes is X30 Y20. The cycle LCYC85 is used for boring. The spindle speed and the direction of rotation are defined in the higher-level program by means of R106/R107. The feed results from the parameters.

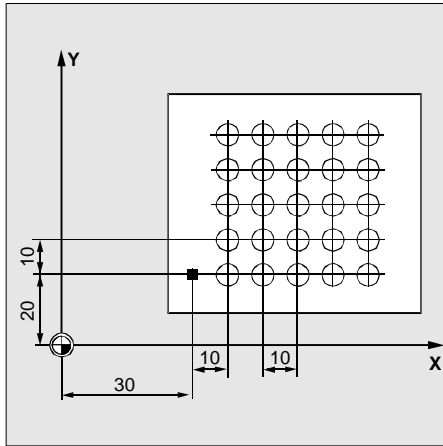


Fig. 9-14 Example drawing

```

N10 G0 G17 G90 S500 M3 T2 D1      ; Define technology values
N20 X10 Y10 Z105                  ; Approach starting position
N30 R1=0, R101=105, R102=2, R103=102,
                                   ; Parameterize drilling cycle, initialize
                                   counter of track of punched holes
                                   (R1)
N40 R104=30 R105=2 R107=100 R108=300
                                   ; Parameterize drilling cycle
N50 R115=85 R116=30 R117=20 R120=0 R119=5
                                   ; Parameterize cycle for track of
                                   punched holes
N60 R120=10 R121=10                ; Parameterize cycle for track of
                                   punched holes
N70 MARKE1: LCYC60                 ; Call cycle for track of punched
                                   holes
N80 R1=R1+1 R117=R117+10          ; Increment counter of row of
                                   holes + define new
                                   reference point
N90 IF R1<5 GOTOB MARKE1           ; Return to mark 1 if condition met
N100 G0 G90 X10 Y10 Z105          ; Approach starting position
N110 M2                             ; End of program

```


9.3.2 Hole circle – LCYC61

Function

This cycle can be used to produce drill or tapped holes arranged along a circle. The type of drill/tapped hole is defined by means of a parameter.

Call

LCYC61

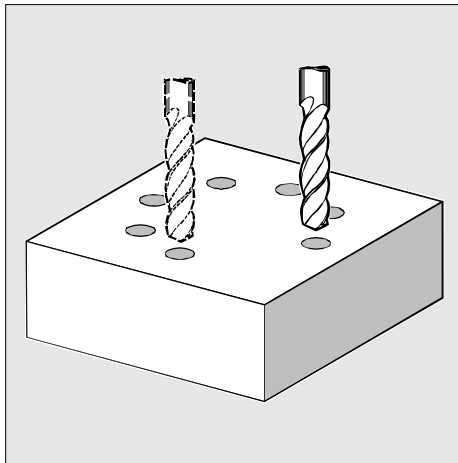


Fig. 9-15

Precondition

The chosen drilling/tapping cycle must also be programmed before calling the drilling pattern cycle.

The required tool with tool offset must be selected before calling the cycle.

Parameters

Parameter	Meaning, Value Range
R115	Number of drilling/tapping cycle Values: 82 (LCYC82), 83 (LCYC83), 84 (LCYC84), 840 (LCYC840), 85 (LCYC85)
R116	Center point of hole circle abscissa (absolute)
R117	Center point of hole circle ordinate (absolute)
R118	Hole circle radius
R119	Number of holes
R120	Starting angle Value range: $-180 < R120 < 180$
R121	Indexing angle

Information

R115	see LCYC60
R116/R117/R118	The position of the hole circle in the machining plane is defined via center point (parameters R116/R117) and radius R118). Only positive values are permitted for the radius.
R119	see LCYC61
R120/R121	These parameters define the arrangement of the holes along the circle. Parameter R120 specifies the angle of rotation between the positive abscissa and the first hole, and R121 defines the angle of rotation between the holes. If parameter R121 is zero, the indexing angle is calculated in the cycle from the number of holes such that these are evenly distributed along the circle.

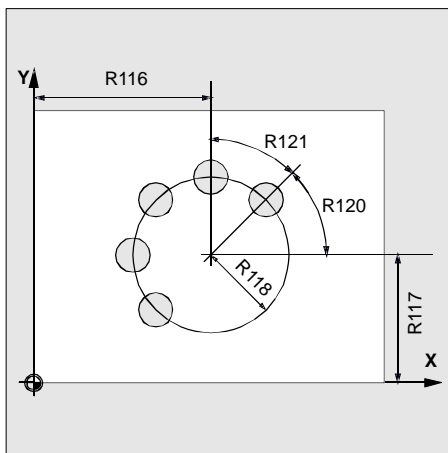


Fig. 9-16

Motional sequence

Starting position:

Any position from which the first drilling position can be approached without collision.

The cycle approaches the first drilling position and produces the hole using the cycle programmed under R115. All further drilling positions are approached with rapid traverse and drilled with the cycle parameterized.

Example

This program is used in conjunction with cycle LCYC82 to produce 4 holes having a depth of 30 mm each. The circle is defined by the center point X70 Y60 and the radius 42mm in the XY plane. The starting angle amounts to 33 degrees. The safety clearance in the drilling axis Z is 2 mm. Spindle speed and direction of rotation, as well as the feed are defined in the higher-level program.

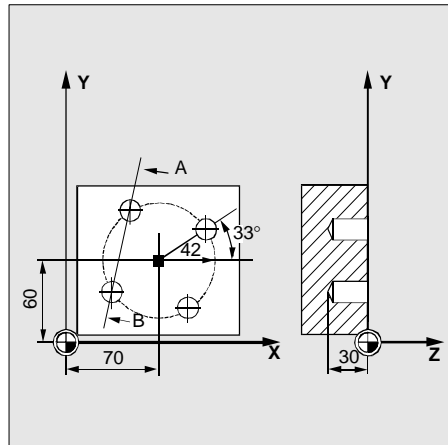


Fig. 9-17

```

N10 G0 G17 G90 F500 S400 M3 T3 D1 ; Define technology parameters
N20 X50 Y45 Z5 ; Approach starting position
N30 R101=5 R102=2 R103=0 R104=-30 R105=1
; Parameterize drilling cycle
N40 R115=82 R116=70 R117=60 R118=42 R119=4
; Parameterize hole circle cycle
N50 R120=33 R121=0 ; Parameterize hole circle cycle
N60 LCYC61 ; Call hole circle cycle
N70 M2 ; End of program

```

9.4 Milling cycles

9.4.1 Cutting square pockets, slots and circular pockets – LCYC75

Function

Provided the correct parameters are assigned, this cycle can be used to cut a square pocket or slot with parallel axes, or a circular pocket. The cycle provides roughing and finishing operations. The parameters pocket length = pocket width and corner radius = pocket length/2 are used to cut a circular pocket with a diameter corresponding to the parameterized pocket length or width. When setting the pocket width to the corner radius = pocket width/2, a slot is cut. The milling cut is always carried out in the 3rd axis, and the down feed is always carried out to the center. This allows cutters to be used which cannot cut across center when rough-boring in this position.

Call

LCYC75

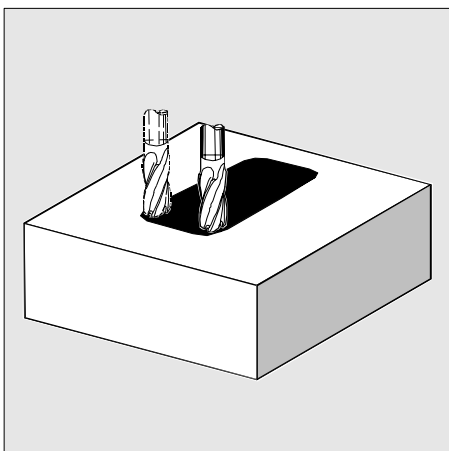


Fig. 9-18

Precondition

When not rough-boring, the cycle requires a cutter with radial tooth cutting across center (DIN844).

The spindle speed and the direction of rotation must be defined in the higher-level program.

The required tool with tool offset must be selected prior to calling the cycle.

Parameters

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Pocket depth (absolute)
R116	Pocket center point abscissa
R117	Pocket center point ordinate
R118	Pocket length
R119	Pocket width
R120	Corner radius
R121	Max. infeed depth
R122	Feed for down infeed
R123	Feed for traversing movement in the plane
R124	Finishing allowance in the plane
R125	Finishing allowance for depth

Parameter	Meaning, Value Range
R126	Cutting direction (G2 or G3) Value range: 2 (G2), 3 (G3)
R127	Machining type 1 – roughing 2 – finishing

Information

R101/R102/R103 see LCYC82

R104 This parameter is used to program the distance between reference plane and pocket base (depth).

R116/R117 The parameters R116 and R117 define the center point of the pocket in abscissa and ordinate.

R121 Use this parameter to determine the maximum infeed depth. The infeed in the cycle is carried out by even steps. Using R121 and R104, the cycle calculates an infeed depth, which is between the 0.5*max. infeed depth and the max. infeed depth. To calculate the individual infeed depths, the distance between reference plane + safety clearance and the pocket depth is used. When using R121=0, the infeed is carried out to the pocket depth immediately. The down infeed starts from the reference plane shifted forward by an amount corresponding to the safety clearance.

R122 This feed acts for all infeeds perpendicular to the machining plane.

R123 This parameter defines the feed for roughing and finishing in the plane.

R124 The finishing allowance of the contour is programmed under R124, with roughing selected. When finishing is selected (R127=2), the parameters R124 and R125 are used to select whether only the contour or contour and depth are to be finished.

Only contour : R124>0 and R125=0

Contour and depth: R124>0 and R125>0
R124=0 and R125=0
R124=0 and R125>0

R125 The finishing allowance programmed under R125 acts with down infeed and roughing selected. When finishing is selected (R127=2), the parameters R124 and R125 are used to select whether only the contour or contour and depth are to be finished.

Only contour : R124>0 and R125=0

Contour and depth: R124>0 and R125>0
R124=0 and R125=0
R124=0 and R125>0

R126 This parameter defines the machining direction of the pocket.

R127 This parameter defines the machining type.

1 – Roughing

The pocket is broached with the programmed parameters down to the finishing allowance.

2- Finishing

The cycle requires a pocket broached down to the finishing allowance so that only the finishing allowance must be broached out. The cycle assumes that the remained finishing allowance is less than the tool diameter.

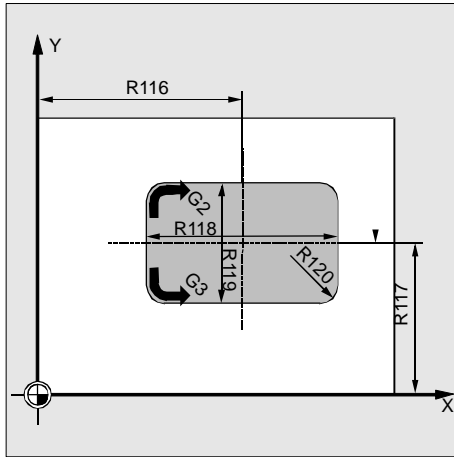


Fig. 9-19

Motional sequence

Starting position:

Any position from which the pocket center point on the level of the retract plane can be approached without collision.

1. Roughing R127=1

G0 is used to approach the pocket center point on the level of the retract plane; G0 is also used to approach the reference plane in this position, which is shifted forward by an amount corresponding to the safety clearance. To machine the pocket, the following steps are carried out:

- Infeed to pocket center point position (next machining depth) with the feed programmed in R122 and the spindle speed acting prior to cycle call;
- Cutting the pocket down to the finishing allowance with the feed programmed in R123 and the spindle speed acting prior to cycle call. If the cutter diameter is greater than the pocket slot width minus finishing allowance, or if the cutter radius is equal to the pocket slot width, the finishing allowance is reduced if possible and a slot is produced by an oscillatory movement.
- The direction of machining depends on the value programmed under R126.
- On completion of the machining, the tool is moved down to the pocket center point to the retract plane, and the cycle is completed.

2. Finishing R127=2

- When several infeed operations are necessary, only the last infeed operation to the final depth with feed and in the pocket center point (R122) is carried out. The infeed operations prior to this are carried out with rapid traverse and, in order to reduce the number of idle motions and a respective size of the pocket provided, with an offset to the pocket center point. The parameters R124 and R125 are used to select whether only the contour or contour and depth are to be machined.

Only contour: R124>0 and R125=0

Contour and depth: R124>0 and R125>0

R124=0 and R125=0

R124=0 and R125>0

The traversing movements in the plane are carried out with a value parameterized under R123. The infeed down to the final depth is carried out with the value parameterized under R122.

- The machining direction is selected by means of the value parameterized under R126.
- On completion of machining of the pocket, the tool is moved to the pocket center point to the retract plane, and the cycle is completed.

Example

Cutting a pocket

This program can be used to produce a pocket with a length of 60 mm, a width of 40 mm, a corner radius of 8 mm and a depth of 17.5 mm. A cutter is used which **cannot** cut across center. To this aim, it is necessary to rough-bore to the pocket center (LCYC82). The finishing allowance is 0.75mm for the pocket margins and 0.5mm in the depth; the safety clearance in the Z axis which is added to the reference plane is 0.5 mm. The pocket center point is in X60 and Y40, and the maximum down infeed is 4 mm. Both roughing and finishing are to be carried out.

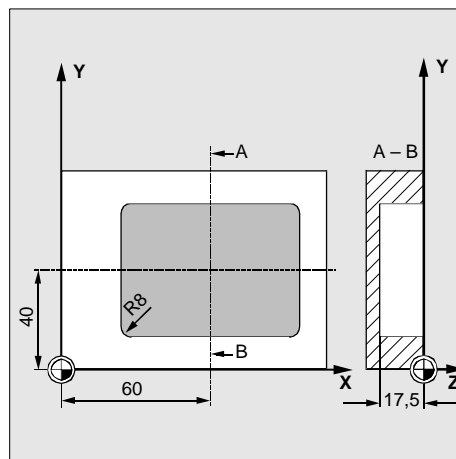


Fig. 9-20 Example drawing

N10 G0 G17 G90 F200 S300 M3 T4 D1 ; Define technology values

N20 X60 Y40 Z5 ; Approach drilling position

N30 R101=5 R102=2 R103=0 R104=-17.5 R105=2

; Parameterize drilling cycle

```

N40 LCYC82 ; Call drilling cycle
N50 ..... ; Tool change
N60 R116=60 R117=40 R118=60 R119=40 R120=8
; Parameterize pocket rough-cutting
; cycle
N70 R121=4 R122=120 R123=300 R124=0.75 R125=0.5
; R101– R104 do not change
N80 R126=2 R127=1 ; in comparison with the drilling cycle
N90 LCYC75 ; Call roughing cycle
N100 ..... ; Tool change
N110 R127=2 ; Parameterize pocket finish-cutting
; cycle (the remaining parameters do not change)
N120 LCYC75 ; Call finishing cycle
N130 M2 ; End of program

```

Cutting a circular pocket

Use this program to produce a circular pocket in the ZY plane. The center point is defined with Z50 Y50, and the depth of the pocket amounts to 20mm. The X axis is the infeed axis for down infeed. No finishing allowance is specified, i.e. the pocket is cut to the final dimensions by roughing. A cutter that cuts across center is used.

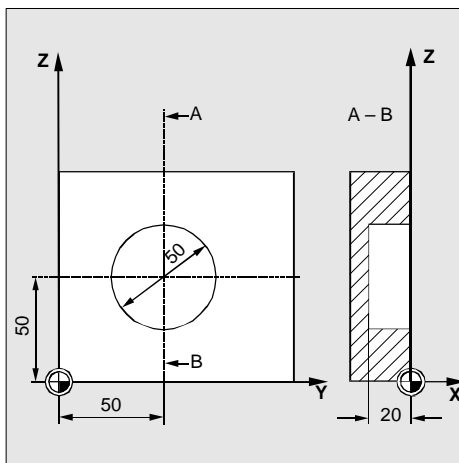


Fig. 9-21 Example drawing

```

N10 G0 G19 G90 S200 M3 T1 D1 ; Define technology values
N20 Z60 X40 Y5 ; Approach starting position
N30 R101=4 R102=2 R103=0 R104=-20 R116=50 R117=50
; Parameterize pocket cutting cycle
N40 R118=50 R119=50 R120=25 R121=4 R122=100
; Parameterize pocket cutting cycle
N50 R123=200 R124=0 R125=0 R126=0 R127=1
; Parameterize pocket cutting cycle
N60 LCYC75 ; Cycle call
N70 M2 ; End of program

```


Cutting a slot

The program realizes an arrangement of 4 slots on a circle in the ZY plane, which are arranged at an angle of 90 degrees to one another and have a starting angle of 45 degrees (see Fig.). The coordinate system is rotated and moved in the higher-level program. The slots have the following dimensions: length 30 mm, width 15 mm, depth 23 mm. The safety clearance is 1 mm, the cutting direction is G2, and the maximum down infeed is 6 mm. The slots are machined down to the final depth by roughing (finishing allowance is zero) using a cutter cutting across center.

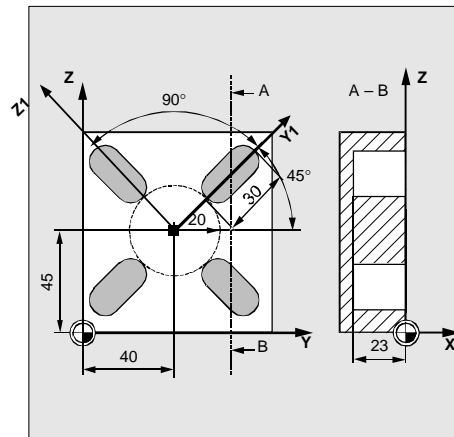


Fig. 9-22 Example drawing

```

N10 G0 G19 G90 T10 D1 S400 M3 ; Define technology values
N20 Y20 Z50 X5 ; Approach starting position
N30 R101=5 R102=1 R103=0 R104=-23 R116=35 R117=0
; Parameterize cutting cycle
N40 R118=30 R119=15 R120=15 R121=6 R122=200
; Parameterize cutting cycle
N50 R123=300 R124=0 R125=0 R126=2 R127=1
; Parameterize cutting cycle
N60 G158 Y40 Z45 ; Set up coordinate system Z1 – Y1
; 1st offset from
; Z45 Y40
N70 G259 RPL=45 ; 2nd rotation of coordinate system
; by 45 Grad
N80 LCYC75 ; Cycle call for cutting the 1st slot
N90 G259 RPL=90 ; Additive rotation of coordinate system
; Z1–Y1 by 90 degrees for cutting the
; 2nd slot
N100 LCYC75 ; Cycle call for cutting the 2nd slot
N110 G259 RPL=90 ; Additive rotation of coordinate system
; Z1–Y1 by 90 degrees for cutting the
; 3rd slot
N120 LCYC75 ; Cut 3rd slot
N130 G259 RPL=90 ; Additive rotation of coordinate system
; Z1–Y1 by 90 degrees for cutting the
; 4th slot
N140 LCYC75 ; Cut 4th slot

```

9.4 Milling cycles

N150 G259 RPL=45	; Restore initial coordinate system
	; Rotate 1st to 0 degrees
N160 G158 Y-40 Z-45	; Undo 2nd offset
N170 Y20 Z50 X5	; Approach starting position
M2	; End of program

Index

A

Address, 8-100

B

Basic principles, 1-22
Block Search, 5-49
Block structure, 8-100

C

Character set, 8-102

D

Data transfer, 7-81
Determining the Tool Offsets, 2-30
Diagnosis, 7-81
Diagnostic functions, 7-87
"Diagnostics" operating area, 7-87

E

Entering Tools and Tool Offsets, 2-27
Entering/Modifying the Zero Offset, 2-32
Error messages, 9-166
Error Treatment, 9-166

F

Fundamentals of NC programming, 8-99

I

Interface parameterization, 7-85
Interface parameters, 7-84

J

Jog mode, 4-39

M

"Machine" operating area, 4-39
Manual Data Input, 4-43
MDA Mode, 4-43

O

Operating Areas, 1-14

P

Part Program
 Selecting/Starting, 5-48
 Stopping/Aborting, 5-49
Password, 7-95

R

R Parameters, 2-38
Repositioning after Interruption, 5-50

S

Screen Layout, 1-11
Setting Data, 2-36
Special Functions, 7-85
Start-up functions, 7-91

T

Tool Offset Data, 2-29

V

V24 interface, 7-81

W

Word structure, 8-100

SIEMENS AG
 A&D MC BMS
 Postfach 3180
 D-91050 Erlangen

(Tel. +49 180 / 5050 – 222 [Hotline]
 Fax +49 9131 / 98 – 2176 [Documentation]
 Mailto: motioncontrol.docu@erlf.siemens.de)

<p>From</p> <p>Name _____</p> <p>Company/dept. _____</p> <p>Street _____</p> <p>Zip code: _____ City: _____</p> <p>Telephone: _____ / _____</p> <p>Telefax: _____ / _____</p>	<p>Suggestions</p> <p>Corrections</p> <p>for Publication/Manual:</p> <p>SINUMERIK 802S/802C</p> <p>User Documentation</p> <p>Operation and Programming – Milling</p> <p>Order No.: 6FC5598–3AA10–0BP2 Edition: 01.02</p> <p>Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.</p>
--	---

Suggestions and/or corrections

Document Structure SINUMERIK 802S and 802C

General Documentation: **Catalog**



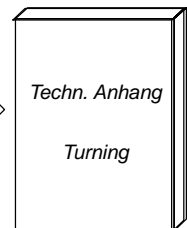
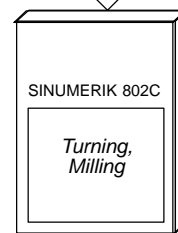
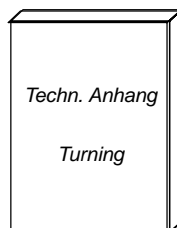
User Manual: **Operation and Programming**



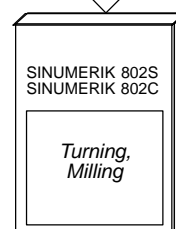
User Manual: **Diagnostics Guide**

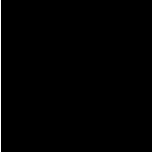


Technical Manual: **Start-up**



Technical Manual: **Description of Functions**





Siemens AG

Automatisierungs- und Antriebstechnik

Motion Control Systems

Postfach 3180, D – 91050 Erlangen

Bundesrepublik Deutschland

www.ad.siemens.de

© Siemens AG 2002
Subject to change without prior notice
Order No.: 6FC5598-3AA10-0BP2

Printed in the Federal Republic of Germany