SIEMENS

SINUMERIK 802D

Operation and Programming Milling

Valid for

Control system	*Software version*
SINUMERIK 802D	2

08/05 Edition

Safety information

This Manual contains information which you should carefully observe to ensure your own personal safety and the prevention of material damage. The notices are highlighted by a warning triangle and, depending on the degree of hazard, represented as shown below:

Danger

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

Warning

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

Caution

with a warning triangle indicates that minor personal injury can result if proper precautions are not taken.

Caution

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

Attention

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

If several hazards of different degrees occur, the hazard with the highest degree must always be given preference. If a warning note with a warning triangle warns of personal injury, the same warning note can also contain a warning of material damage.

Qualified personnel

Start-up and operation of the device/equipment/system in question must only be performed using this documentation. The start-up and operation of a device/system must only be performed by **qualified personnel**. Qualified personnel as referred to in the safety guidelines in this documentation are those who are authorized to start up, earth and label units, systems and circuits in accordance with the relevant safety standards.

Proper use

Please note the following:

Warning

The device may only be used for the applications described in the Catalog and only in combination with the equipment, components and devices of other manufacturers as far as this is recommended or permitted by Siemens. It is assumed that this product be transported, stored and installed as intended and maintained and operated with care to ensure that the product functions correctly and properly.

Trademarks

All designations marked with the copyright notice ® are registered trademarks of Siemens AG. Other names in this publication might be trademarks whose use by a third party for its own purposes may violate the rights of the registered holder.

Disclaimer of liability

Although we have checked the contents of this publication for agreement with the hardware and software described, since differences cannot be totally ruled out. Nonetheless, differences might exist and therefore we cannot guarantee that they are completely identical. The information given in this publication is reviewed at regular intervals and any corrections that might be necessary are made in the subsequent editions.

Siemens AG
Automation and Drives
Postfach 4848
90437 NÜRNBERG
GERMANY

6FC5698-2AA10-1BP5

Preface

SINUMERIK Documentation

The SINUMERIK Documentation is organized in 3 levels:

- General Documentation:
- User Documentation
- Manufacturer/Service Documentation:

For detailed information regarding further publications about SINUMERIK 802D, as well as for publications that apply for all SINUMERIK control systems (e.g. Universal Interface, Measuring Cycles...), please contact your Siemens branch office.

A monthly overview of publications with specification of the available languages can be found on the Internet at:
http://www.siemens.com/motioncontrol
Follow the menu items "Support"/"Technical Documentation"/"Overview of Publications".

The Internet edition of DOConCD – DOConWEB – can be found at:
http://www.automation.siemens.com/doconweb

Addressees of the documentation

The present documentation is aimed at the machine tool manufacturer. This publication provides detailed information required for the machine tool manufacturer to start up the SINUMERIK 802D control system.

Standard scope

The present Instruction Manual describes the functionality of the standard scope. Any amendments made by the machine manufacturer are documented by the machine manufacturer.

Other functions not described in this documentation can possibly also be performed on the control system. However, the customer is not entitled to demand these functions when the new equipment is supplied or servicing is carried out.

Hotline

If you have any questions, do not hesitate to call our hotline:
A&D Technical Support
Tel.: +49 (0) 180 / 5050 – 222
Fax: +49 (0) 180 / 5050 – 223
Internet: http://www.siemens.de/automation/support–request

If you have any questions (suggestions, corrections) regarding the Documentation, please send a fax to the following number or an e–mail to the following address:
Fax: +49 (0) 9131 / 98 – 63315
E–mail: motioncontrol.docu@siemens.com

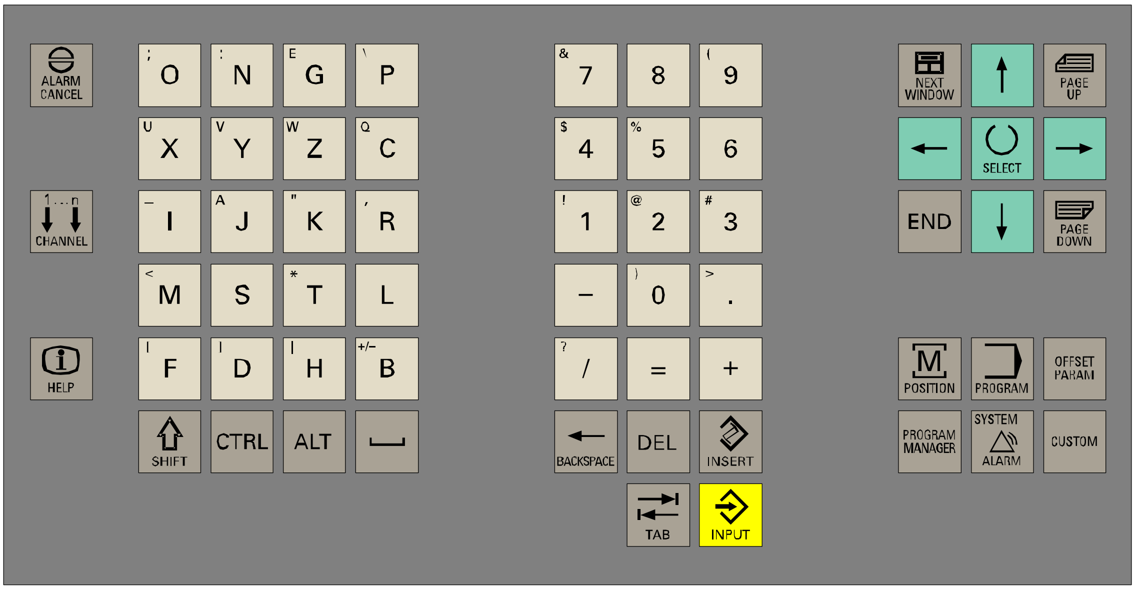
Fax form: see return fax form at the end of this publication

Internet address

http://www.siemens.com/motioncontrol

Contents

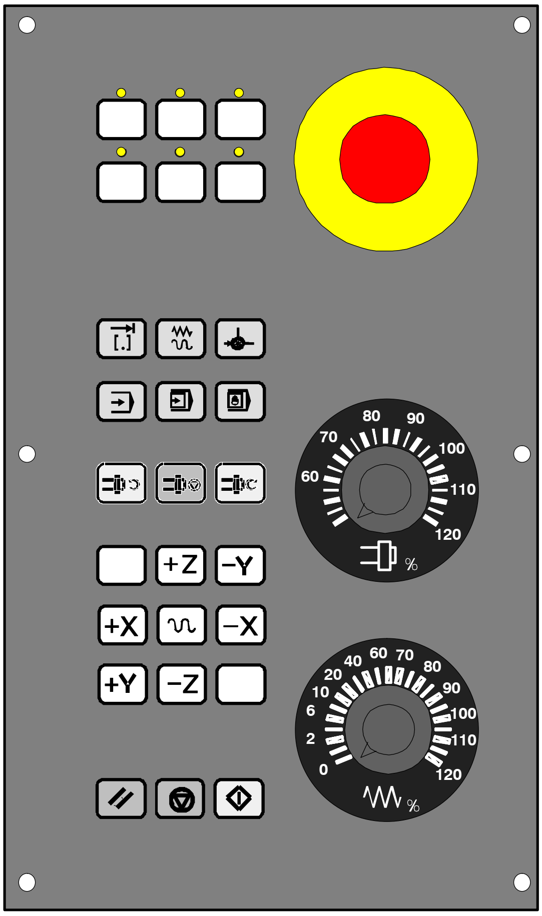
SINUMERIK 802D Key Definition



Key	Function	Key	Function
^	"Recall" key	POSITION	"Position" operating area key
>	ETC key	PROGRAM	"Program" operating area key
ALARM CANCEL	"Acknowledge alarm" key	OFFSET PARAM	"Parameter" operating area
CHANNEL	without function	PROGRAM MANAGER	"Program Manager" operating area
HELP	Info key	SYSTEM ALARM	"Alarm" / "System" operating areas (Shift+key)
SHIFT	Shift key	CUSTOM	
CTRL	Controlkey	NEXT WINDOW	not assigned
ALT	Altkey	PAGE UP / PAGE DOWN	PageUp / PageDown keys
SPACE	SPACE	↑ ← → ↓	Cursor keys
BACKSPACE	Backspace	SELECT	Selection key / toggle key
DEL	Clear key	END	
INSERT	INSERT key	J (A) / Z (W)	Alphanumeric keys Double assignment on the Shift level
TAB	Tabulator		
INPUT	ENTER / Input key		

0 9 Numeric keys
Double assignment on the Shift level

External Machine Control Panel



RESET

NC STOP

NC START

EMERGENCY STOP

Spindle override (option)

User–defined key with LED

User–defined key without LED

INCREMENT

JOG

REFERENCE POINT

AUTOMATIC

SINGLE BLOCK

MANUAL DATA

SPINDLE START LEFT

SPINDLE STOP

SPINDLE START RIGHT

RAPID TRAVERSE OVERRIDE

+X –X X axis

+Z –Z Z axis

Feedrate override

Introduction

1

1.1 Screen layout

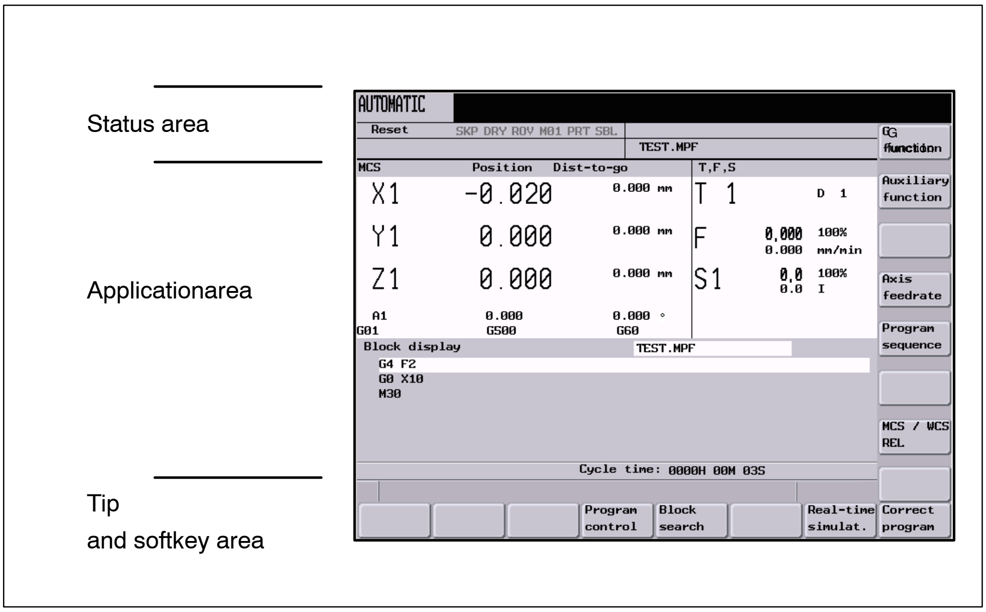


Fig. 1-1 Screen layout

The screen is divided into the following main areas:

- Status area
- Applicationarea
- Tip and softkey area

Status area

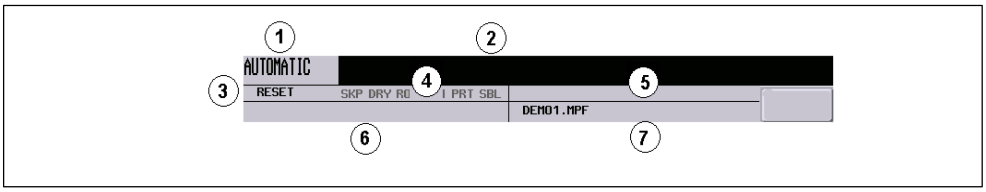


Fig. 1-2 Status area

Table 1-1 Explanation of the display elements in the status area

Screen Control	Display	Meaning
①	**Active operating area, active mode** Position JOG; 1 INC, 10 INC, 100 INC, 1000 INC, VAR INC (evaluation by increments in the JOG mode) MDA AUTOMATIC Offset Program Program Manager System Alarm Marked as an ”external language” using G291	
②	**Alarm and message line** In addition, the following is displayed: 1. Alarm number with alarm text, or 2. Message text	
③	**Program status**	
	RESET	Program canceled / default status
	RUN	Program running
	STOP	Program stopped
④	**Program controls in the AUTOMATIC mode**	
⑤	**Path**	N: – NC internal ”drive” D: – CF card
⑥	**NC messages**	
⑦	**Selected part program (main program)**	

Tip and softkey area

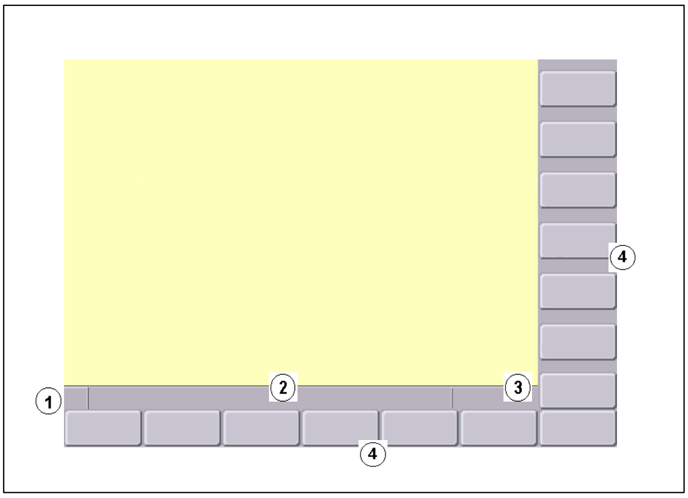
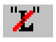
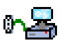


Fig. 1-3 Tip and softkey area

Table 1-2 Explanation of the screen elements in the tip and softkey area

Screen Control	Display	Meaning
①		**Recall symbol** Pressing the Recall key lets you return to the next higher menu level.
②		**Tip line** Displays tips for the operator
③	   	**MMC status information** **ETC** is possible (Pressing this key displays the horizontal softkey bar providing further functions.) Mixed notation active (uppercase/lowercase letters) Data transfer running Connection to the PLC programming tool active
④		**Softkey bar** vertical and horizontal

Standard softkeys

Use this softkey to quit the screenform.

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

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

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

1.2 Operating areas

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

	Position	Machine operation
	Offset/Parameters	Input of offset values and setting data
	Program	Creation of part programs
	Program Manager	Part program directory
 	System	Diagnosis, start–up
	Alarm	Alarm and message lists

To switch the operating area, use the relevant key (hard key).

Protection levels

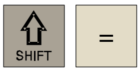
The input and modification of vital data in the control system is protected by passwords.

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

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

1.3 Accessibility options

1.3.1 Calculator



The calculator function can be activated from any operating area using ”SHIFT” and ”=”.

To calculate terms, the four basic arithmetic operations can be used, as well as the functions ”sine”, ”cosine”, ”squaring” and ”square root”. A bracket function is provided to calculate nested terms. The bracket depth is unlimited.

If the input field is already occupied by a value, the function will accept this value into the input line of the calculator.

When you press the **Input** key, the result is calculated and displayed in the calculator.

Selecting the **Accept** softkey enters the result in the input field at the current cursor position of the part program editor and closes the calculator automatically.

Note

If an input field is in the editing mode, it is possible to restore the original status using the ”Toggle” key.

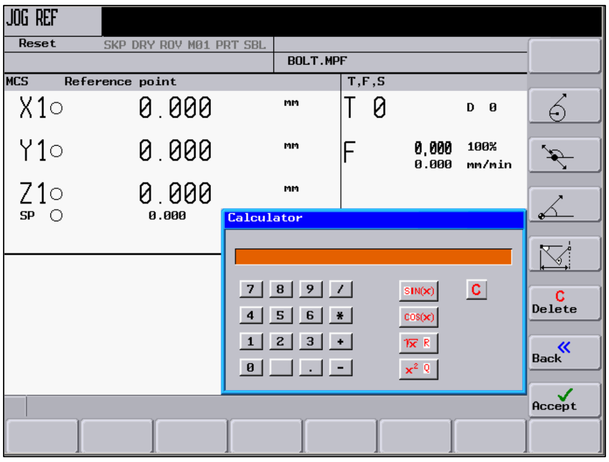


Fig. 1-4 Calculator

Characters permitted for input

+, – *, /	Basic arithmetic operations
S	Sine function The X value (in degrees) in front of the input cursor is replaced by the sin(X) value.
O	Cosine function The X value (in degrees) in front of the input cursor is replaced by the cos(X) value.
Q	Square function The X value in front of the input cursor is replaced by the X^2 value.

R Square root function
The X value in front of the input cursor is replaced by the √ value.

( ) Bracket function (X+Y)*Z

Calculation examples

Task	Input -> Result	
100 + (67*3)	100+67*3	-> 301
sin(45°)	45 S	-> 0.707107
cos(45°)	45 O	-> 0.707107
4^2	4 Q	-> 16
√4	4 R	-> 2
(34+3*2)*10	(34+3*2)*10	-> 400

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

- Calculating the tangential transition between a circle sector and a straight line
- Moving a point in the plane
- Converting polar coordinates to Cartesian coordinates
- Adding the second end point of a straight line/straight line contour section given from an angular relation

Softkeys

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

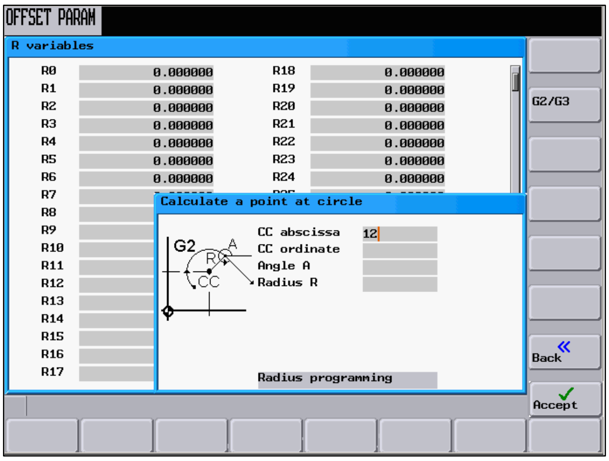


Fig. 1-5

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

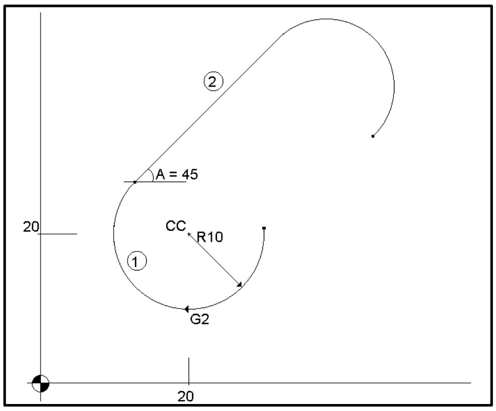
Use the G2 / G3 softkey to define the direction of rotation of the circle.

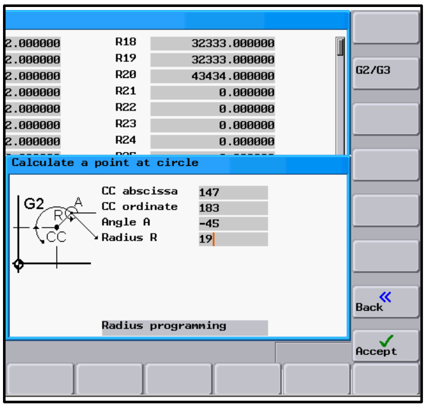
Use this softkey to calculate the abscissa and ordinate values. The abscissa is the first axis of the plane, and the ordinate is the second axis of the plane. The abscissa value is copied into the input field from which the calculator function has been called, and the value of the ordinate is copied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

Example: *If the G18 plane is active, the abscissa is the Z axis and the ordinate the X axis.*

Example: *Calculate the intersection point between the circle sector* ① *and the straight line* ②*.*

Given: *Radius: 10*
Circle center: Z 20 X 20
Connection angle of the straight line: 45°
Direction of rotation: G2





Result: *X = 12.928*
Y = 27.071

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

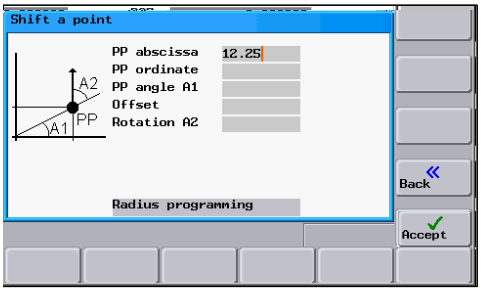


Fig. 1-6

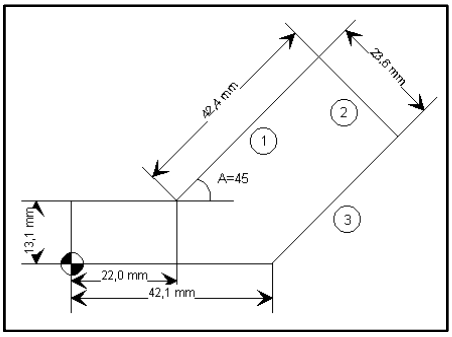
Enter the following coordinates or angles:

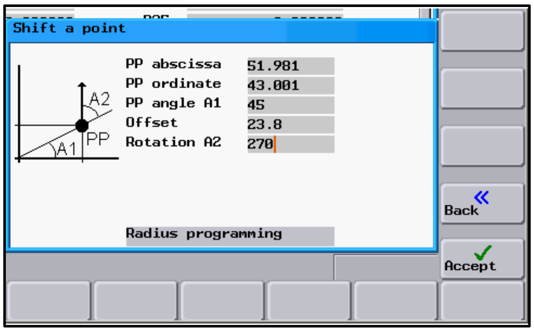
- the coordinates of the given point (PP)
- the slope angle of the straight line (A1)
- the distance of the new point with reference to PP(offset)
- the slope angle of the connecting straight line (A2) with reference to A1

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

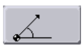
Example

Calculating the end point of the straight line (2). The straight line stands vertically on the end point of the straight line (1) (Coordinates: X = 51.981, Y = 43.081) (see example: "Converting polar coordinates into Cartesian coordinates"). The length of the straight lines is also given.





Result: *X = 68.668*
Y = 26.393



This function converts the given polar coordinates into Cartesian coordinates.

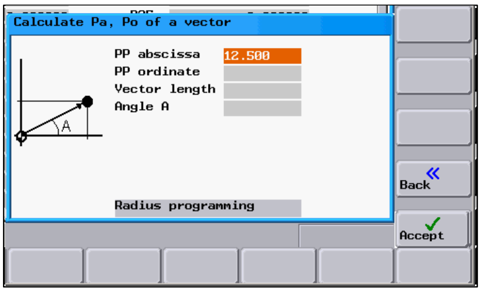


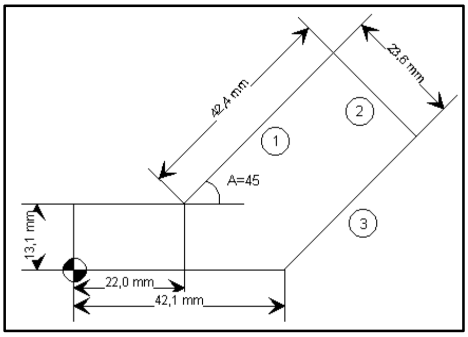
Fig. 1-7

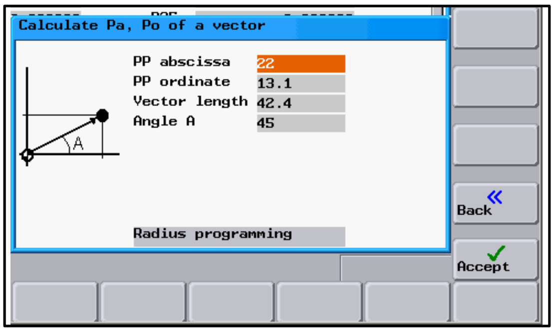
Enter the reference point, the vector length and the slope angle.

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

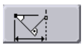
Example

Calculating the end point of the straight line ① *. The straight line is determined by the angle A=45° and its length.*





Result: *X = 51.981*
Y = 43.081



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

The following values of the straight line are known:

Straight line 1:
Starting point and slope angle

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

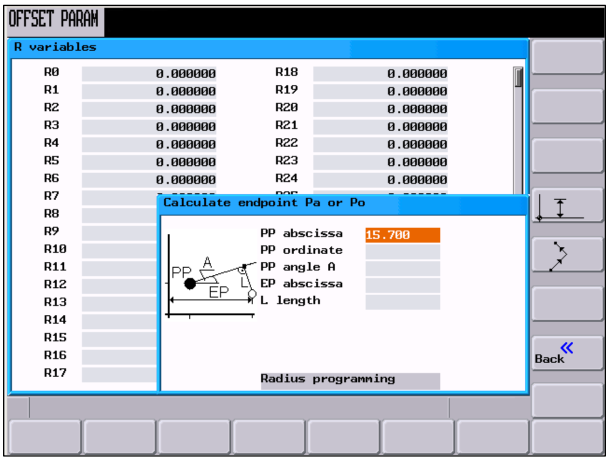
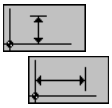
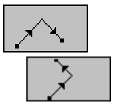


Fig. 1-8



This function is used to select the given coordinate of the end point.
The ordinate value or the abscissa value is given.



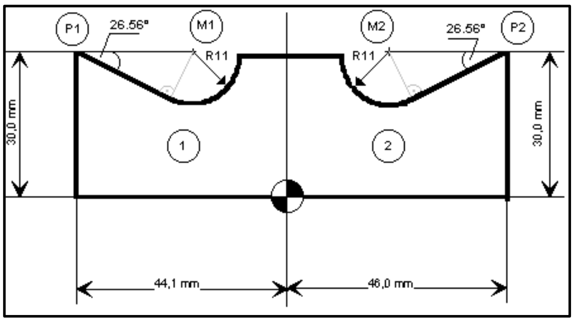
The second straight line is rotated in the CW direction or in the CCW direction by 90 degrees relative to the first straight line.

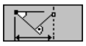
This function will select the relevant end position.

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

If the function has been called from the part program editor, the coordinates are saved with the axis names of the selected basic plane.

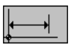
Example



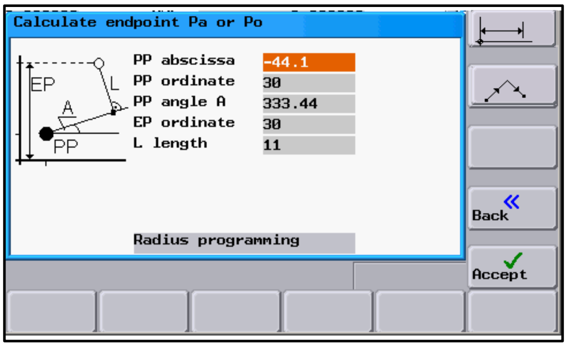
Add the present drawing by the values of the center circle to be able to calculate the points of intersection between the circle sectors. The missing coordinates of the center points are calculated using the calculator function,  *since the radius in the tangential transition stands vertically on the straight line.*

Calculating M1 in section 1:

In this section, the radius stands in the counterclockwise direction turned on the straight line section.

Use the softkeys  *and*  *to select the given configuration.*

Enter the coordinates of the pole (PP) P1, the slope angle of the straight line, the given ordinate value and the circle radius as the length.

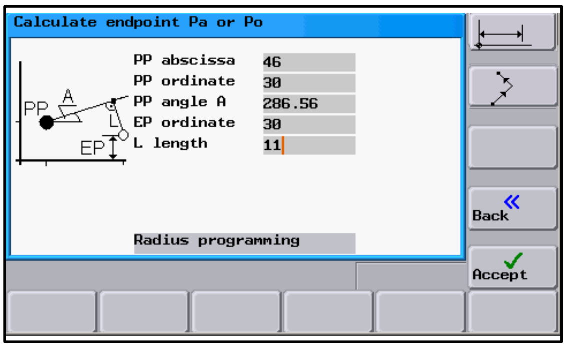


Result: *X = −19.449*
Y = 30

Calculating M2 in section 2:

In this section, the radius stands in the clockwise direction turned on the straight line section. Use the softkeys  *to select the given configuration.*

Enter the parameters in the screenform.



Result: *X = 21.399*
Y = 30

1.3.2 Editing Chinese characters

This function is only available in the Chinese language version.

The control system provides a function for editing Chinese characters in the program editor and in the PLC alarm text editor. After activation, type the phonetic alphabet of the searched character in the input field. The editor will then offer various characters for this sound, from which you can choose the desired one by entering either of the digits 0 to 9.

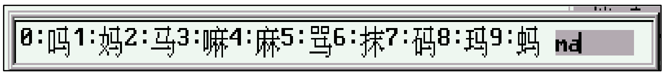


Fig. 1-9 Chinese editor

Alt S Is used to turn on / turn off the editor

1.3.3 Hotkeys

This operator control can be used to select, copy, cut and delete texts using special key commands. These functions are available both for the part program editor and for input fields.

CTRL	C	Copy
CTRL	B	Select
CTRL	X	Cut
CTRL	V	Paste
Alt	L	Switches between uppercase and lowercase letters
Alt or Info key	H	Help system

1.4 The help system

To activate the help system, use the Info key. It offers a brief description for all important operating functions.

In addition, the help offers the following topics:

- Overview of the NC commands with a brief description
- Cycle programming
- Explanation of the drive alarms

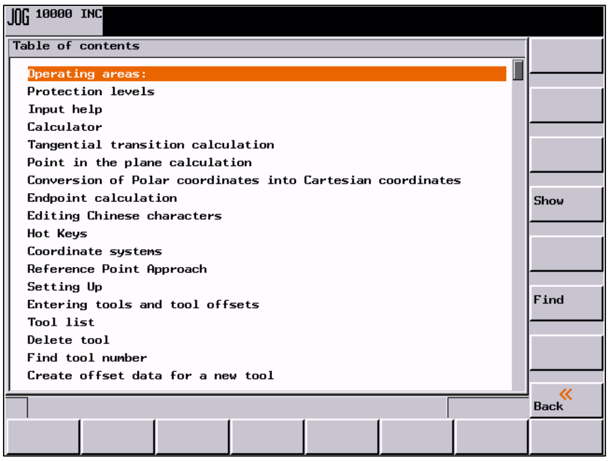


Fig. 1-10 Table of contents of the help system

Show

This function opens the selected topic.

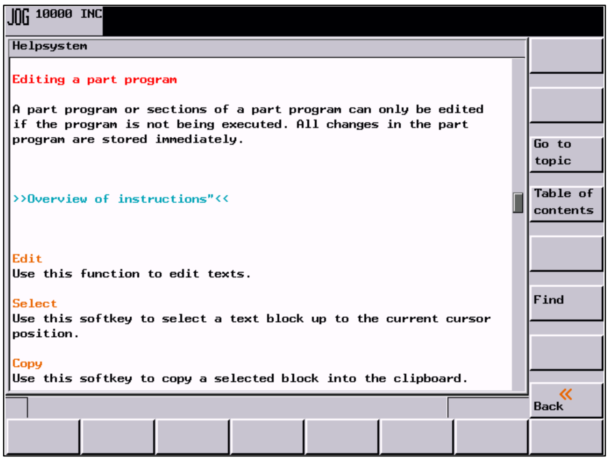


Fig. 1-11 Description for a help topic

Go to topic

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

Back to topic

If you select a cross reference, in addition, the **Back to topic** softkey is displayed. This function lets you return to the previous screenform.

Find

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

Help in the ”Program editor” area

The system offers an explanation for each NC instruction. To display the help text directly, position the cursor after the appropriate instruction and press the Info key. This possibility will only function if the NC instruction is written using uppercase letters.

1.5 Coordinate systems

For machine tools, right–handed, right–angled coordinate systems are used.
The movements on the machine are described as a relative movement between tool and workpiece.

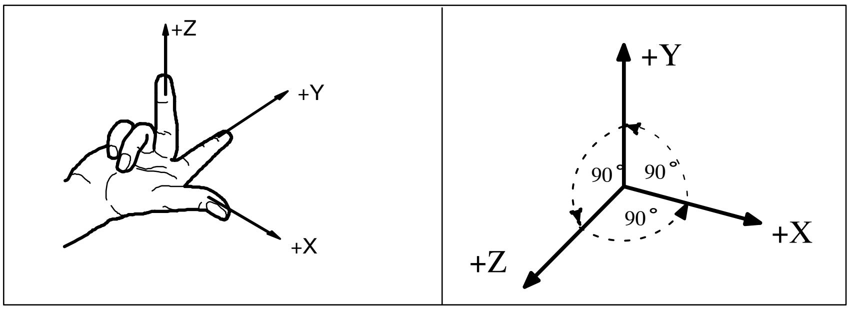


Fig. 1-12 Definition of the directions of the axes one to another; right–angled coordinate system

Machine coordinate system (MCS)

How the coordinate system is located with reference to the machine, depends on the machine type concerned. It can be rotated in different positions.

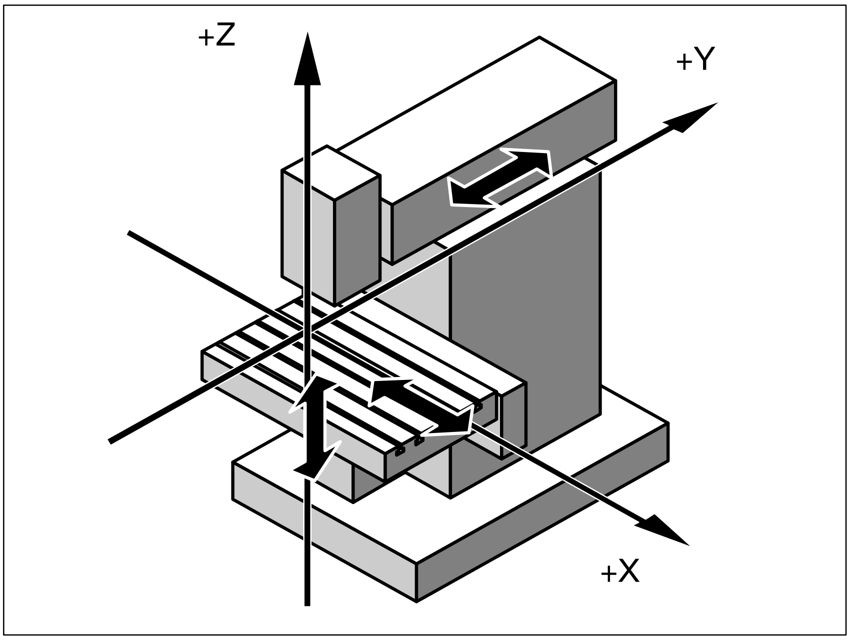


Fig. 1-13 Machine coordinates/machine axes using the example of a milling machine

The origin of the coordinate system is the **machine zero.**
All axes have zero position. This point only represents a reference point defined by the machine manufacturer. It need not be approachable.

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

Workpiece coordinate system (WCS)

The coordinate system described above (see Fig. 1-12) 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 to know the real motion relations on the machine, i.e. he need not to know whether the workpiece or the tool moves. Furthermore, it can be different from axis to axis. The directions are always defined such if the workpiece would be resting and the tool would move.

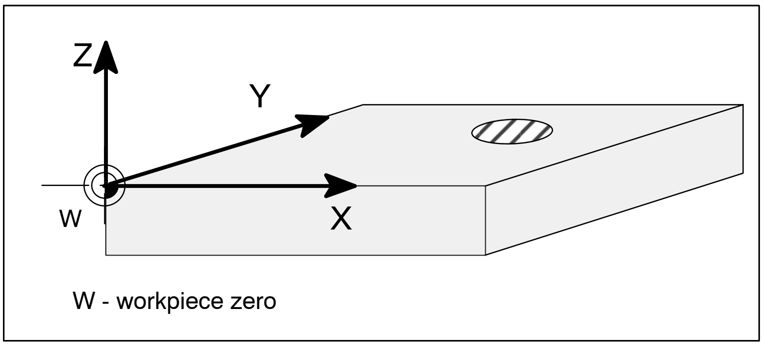


Fig. 1-14 Workpiece coordinate system

Relative coordinate system

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

Clamping the workpiece

For machining, the workpiece is clamped on the machine. The workpiece must be aligned such that the axes of the workpiece coordinate system run in parallel with those of the machine. Any resulting offset of the machine zero with reference to the workpiece zero is determined for each axis individually and entered in the relevant data areas intended for the **settable work offset**. In the NC program, this offset is activated, e.g. using a programmed G54 (see also Section ”Workpiece clamping – settable work offset, ...”).

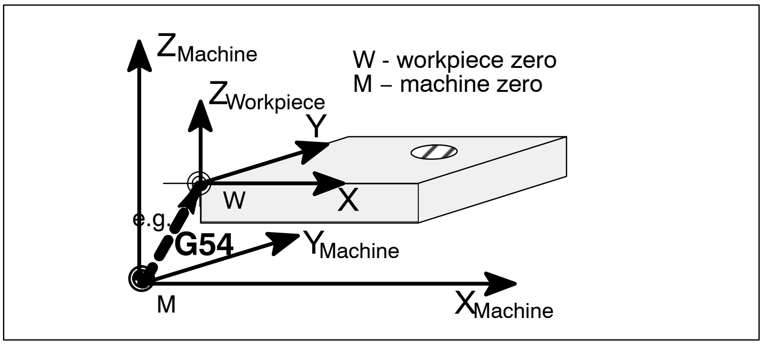


Fig. 1-15 Workpiece on the machine

Current workpiece coordinate system

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

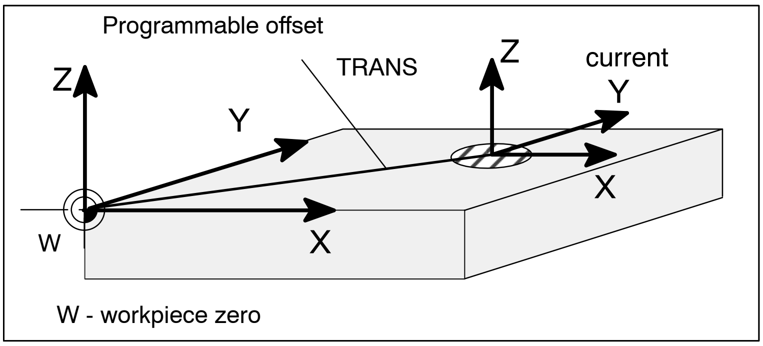


Fig. 1-16 Coordinates on the workpiece; current workpiece coordinate system

This sheet has been left empty for your notes

Turning On and Reference Point Approach

2

Note

When you turn on the SINUMERIK 802D and the machine, please also observe the Machine Documentation, since turning on and reference point approach are machine–dependent functions.

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

Operating sequence

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

The *”Reference point approach”* window is active.

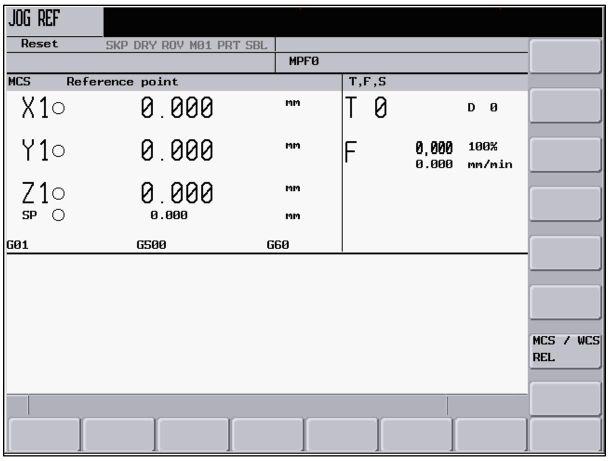
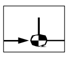


Fig. 2-1 The ”Jog–Ref” start screen



Use the **Ref** key on the machine control panel to activate ”reference point approach”.

The ”Reference point approach” window (Fig. 2-1) displays whether or not the axes have a reference point.

○ Axis must be referenced

◕ Axis has reached its reference point

+X ... -Z

Press a direction key.

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

Approach the reference points for each axis one after the other.
Quit the function by switching the mode (**MDA**, **AUTOMATIC** or **Jog**).

Note

”Reference point approach” is only possible in the **Jog** mode.

Setting Up

3

Preliminary remarks

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

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

3.1 Entering tools and tool offsets

Functionality

The tool offsets consist of several data describing the geometry, the wear and the tool type. Depending on the tool type, each tool is assigned a defined number of parameters. Tools are identified by a number (T number).

See also Section 8.6 ”Tool and tool compensation”

Operating sequences

Tool
List

Use this softkey to open the ”Tool offset data” window which contains a list of the tools created. Use the cursor keys and the PageUp / PageDown keys to navigate in this list.

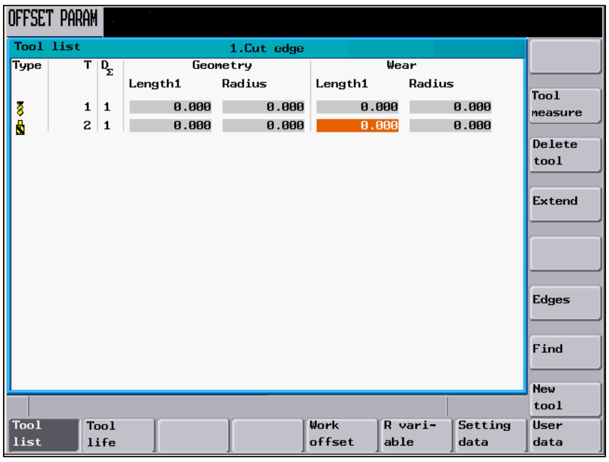


Fig. 3-1

Enter the offsets by positioning the

- cursor bar on the input field to be changed,
- enter the value(s)

INPUT

and either press **Input** or use a cursor key to confirm.

For special tools, the Extend softkey function is provided which offers a complete parameter list which can be filled out.

Softkeys

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

Measure manually — Use this softkey to determine the tool compensation data manually.

Measure auto — Use this softkey to determine the tool offset data semi–automatically (only applies in conjunction with a sensing probe).

Calibrate probe — Use this softkey to calibrate the sensing probe.

Delete tool — Selecting this softkey will delete the tool offset data of all edges of the tool.

Extend — Use this function to display all parameters of a tool.

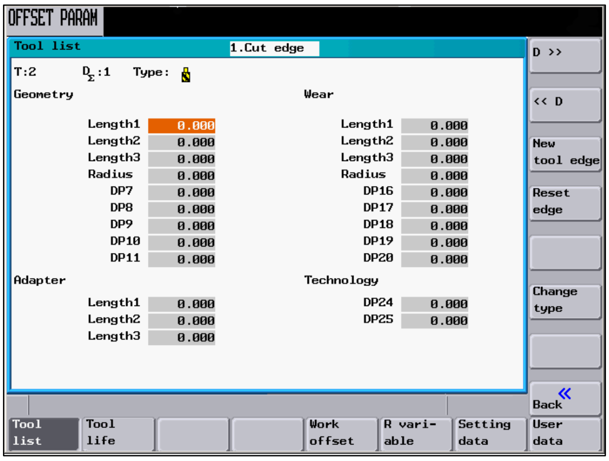


Fig. 3-2 Input screen for special tools

For the meanings of the parameters, please refer to the Section ”Programming”.

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

D >> — Use this softkey to select the next higher edge number.

<< D — Use this softkey to select the next lower edge number.

New tool edge — Use this softkey to create a new edge.

Reset edge — Use this softkey to reset all compensation values of the edge to zero.

Change type — This function is intended to change the tool type. select the tool type using the appropriate softkey.

Find — Find tool number
Type the number of the tool you are looking for and select the **OK** softkey to start searching. If the tool you are looking for exists, the cursor is positioned on the appropriate line.

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

3.1.1 Use this softkey to create a new tool.

Operating sequence

New tool — This function offers another two softkey functions to select the tool type. After selecting the tool type, type the desired tool number in the input field.

Fig. 3-3 The "New tool" window

Input of the tool number

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

3.1.2 Determining the tool offsets (manually)

Functionality

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

Prerequisite

The relevant tool is loaded. In the JOG mode, you will approach the **edge** of the tool to a machine point whose **machine coordinate values** are known. This can be a workpiece with a known position.

Procedure

Enter the reference point in the appropriate field X0, Y0 or Z0.

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

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

Note: You can also use a zero already determined (e.g value of G54). In this case, use the edge of the tool to approach the workpiece zero point. If the edge is positioned directly at workpiece zero, the reference point is zero.

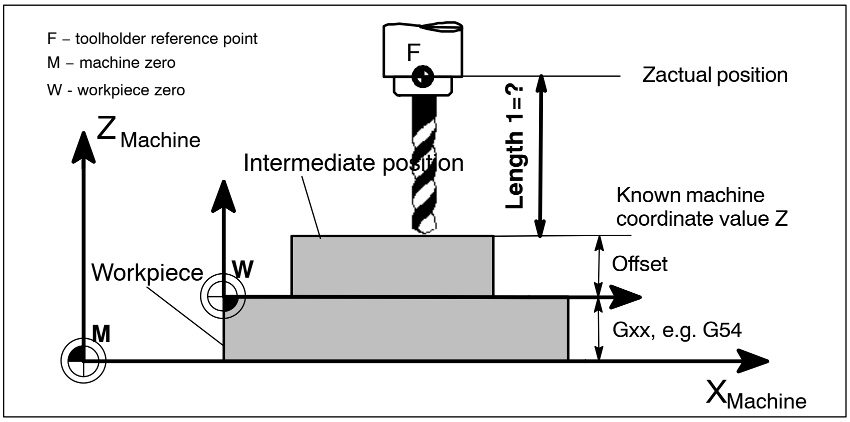


Fig. 3-4 Determination of the length offset using the example of a drill: Length 1 / Z axis

Operating sequence

Tool Measur.	Select this softkey. The *Measure tool* window is opened. You will automatically get to the "Position" operating area.

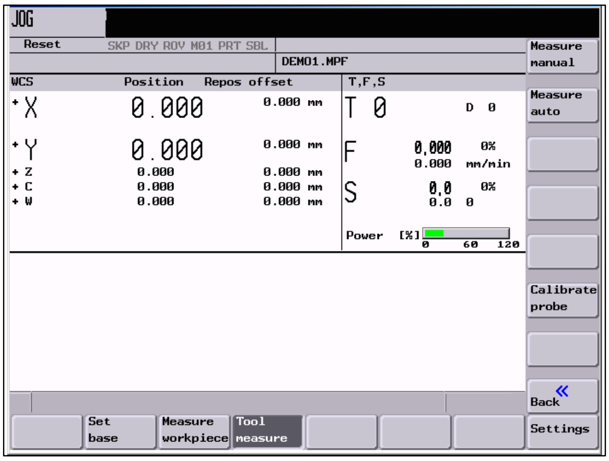


Fig. 3-5 Selecting manual or semiautomatic measuring

Measure manually

The *Measure tool* window is opened.

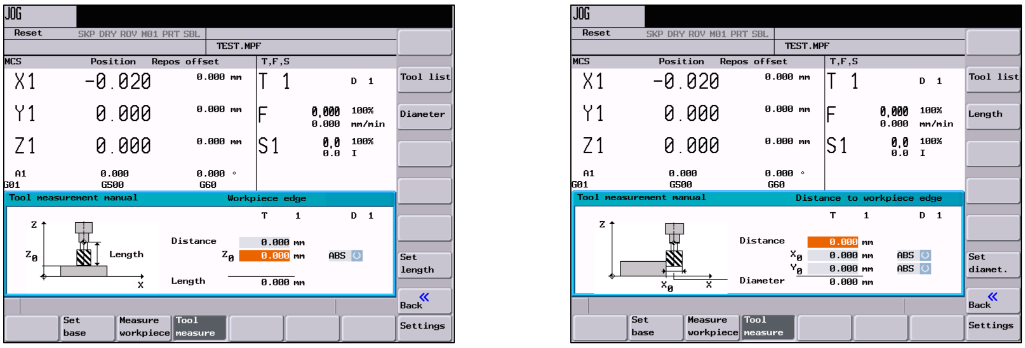


Fig. 3-6 ”Offset values” window; measuring the length and the tool diameter

- Enter the reference point in the field X0, Y0 or Z0. This can be either the current machine coordinate (absolute) or a value from the work offsets (base, G54 – G59). If any other values are used, the offset value will refer to the specified position.
- After selecting the **Set length** or **Set diameter** softkey, the control system will calculate the searched geometry length 1 or the diameter according to the preselected axis. The offset value determined will be stored.
- If a spacer is inserted between the tool and the workpiece, its thickness can be entered in the ”Clearance” field.

3.1.3 Determining tool compensations using a probe

Operating sequence

Tool Measur. | Measure auto

Use this softkey to open the *Measure tool* window.

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

This setting can be changed in the **Probe data settings** screenform (Section 3.1.4).

Note

To create the measuring program, the ”Safety clearance” parameters from the ”Settings” screenform and the feedrate from the ”Probe data” screenform are used.

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

Measuring the tool length

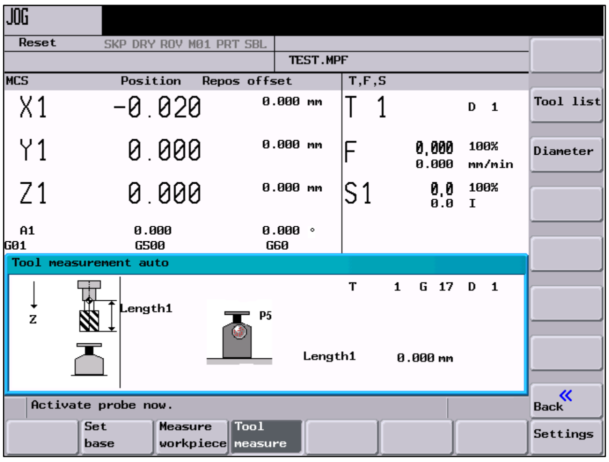
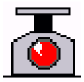
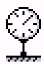


Fig. 3-7 The ”Offset values” window; measuring the tool length

Use the feed axis to traverse to the probe.

After the ”Probe triggered” has  appeared, release the traversing key and wait until the measuring process is completed. A dial gauge  symbolizing the active measuring process is displayed on the animated screen during the automatic measurement.

Measuring the tool diameter

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

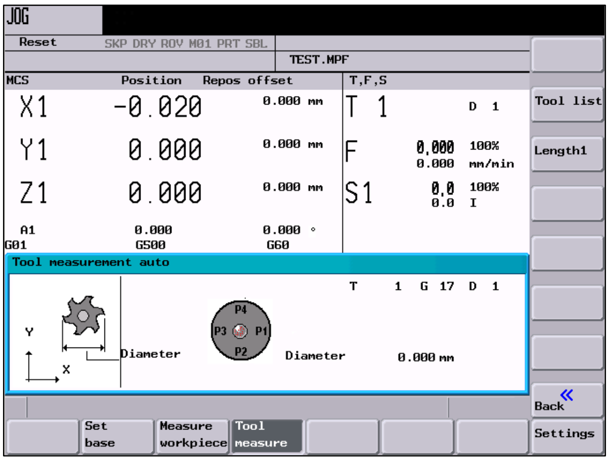
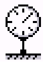


Fig. 3-8 The ”Offset values” window; measuring the diameter

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

After the ”Probe triggered” has  appeared, release the traversing key and wait until the measuring process is completed. A dial gauge  symbolizing the active measuring process is displayed on the animated screen during the automatic measurement.

Warning

The spindle will rotate at the speed defined in the probe data!

3.1.4 Probe settings

Settings | Data probe

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

- Plane of the probe
- Axis feedrate
- Speed and direction of rotation of the spindle
  The direction of rotation of the spindle must be opposite to the cutting direction of the cutter.

All position values refer to the machine coordinate system.

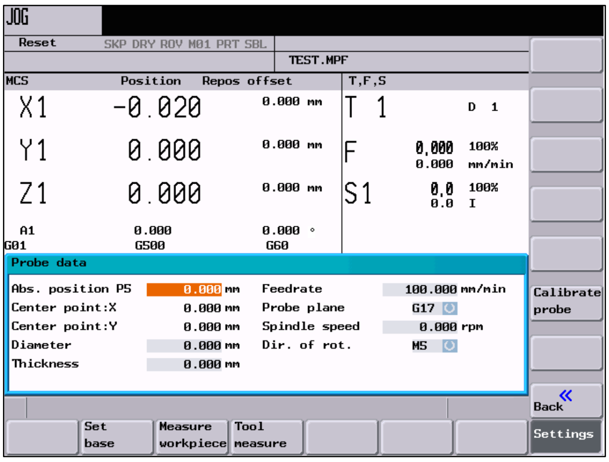


Fig. 3-9 The "Probe data" interactive screenform

Table 3-1 Meaning of the input fields

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

Calibrating the probe

Calibrate probe

The calibration of the probe can be carried out either in the **Settings menu or in the Tool measure menu.**

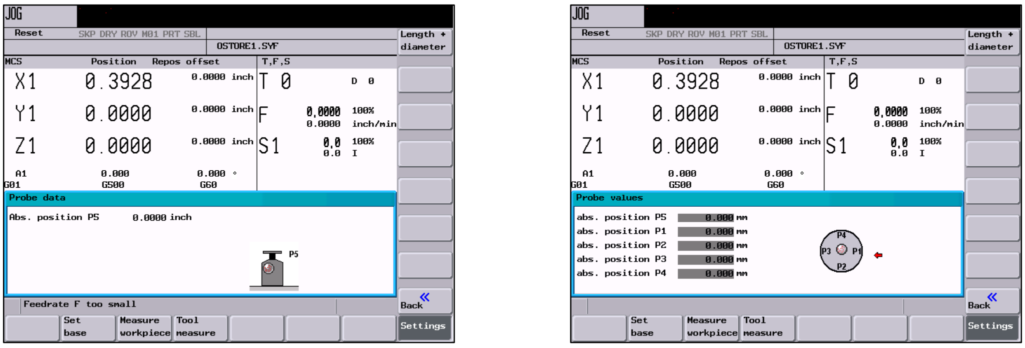
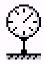


Fig. 3-10 Calibrating the probe (length) (diameter)

After the screenform has appeared, an animation signaling the step to be executed is displayed next to the current positions of the probe. This point must be approached with the appropriate axis. If the probe is triggered, the control system will take over the measuring process by switching to the AUTOMATIC mode, activating the measuring program and starting it automatically. The operator will see an axis movement in the opposite direction for a short time.

During the automatic measurement, a dial appears  symbolizing that the NC is active.

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

Note

To create the measuring program, the parameters ”Safety clearance” from the ”Settings” screenform and feedrate from the ”Probe data” screenform are used.

3.2 Tool monitoring

Tool-life

Each monitoring type is represented in 4 columns.

- Setpoint
- Prewarning limit
- Residual value
- active

Use the checkbox element in the 4th column to enable / disable the monitoring type.

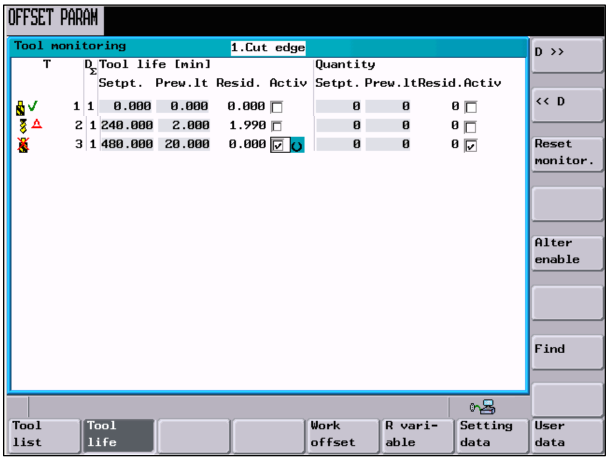


Fig. 3-11 Tool monitoring

Symbols in the T column provide information on the tool status.

 Prewarning limit reached

 Tool disabled

 Tool is monitored

Reset monitor

Use this softkey to reset the monitoring values of the selected tool.

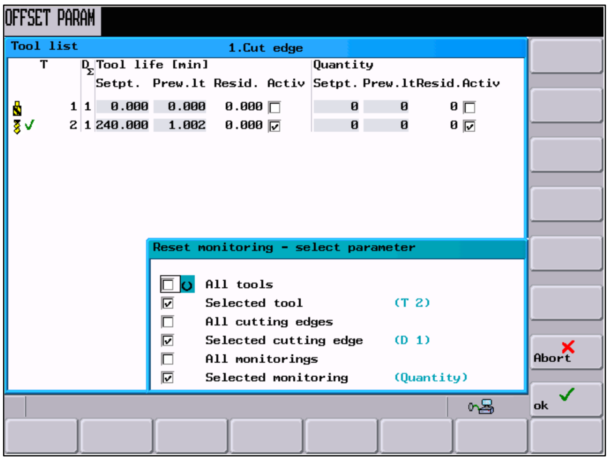


Fig. 3-12

After enable

Use this softkey to change the enable of the selected tool.

3.3 Entering/modifying a work offset

Functionality

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

Operating sequences

Use **Offset Parameter** and **Work Offset** to select the work offset.

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

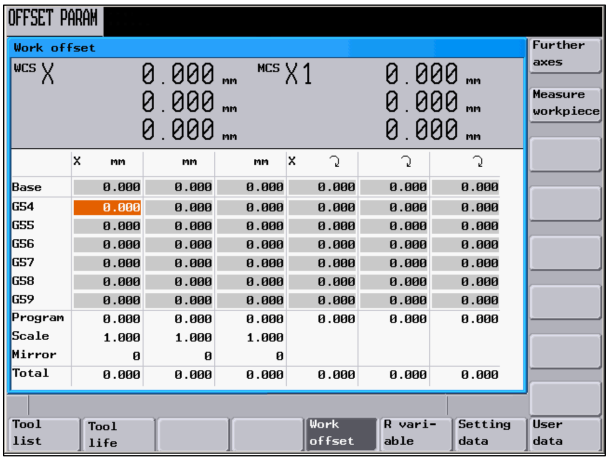
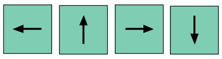
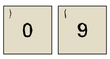


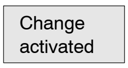
Fig. 3-13 The ”Work offset” window



Position the cursor bar on the input field to be changed



and enter the value(s). Either move the cursor a press the **Input** key to accept the values from the input fields into the work offsets.



The compensation values of the cutting edge come into effect immediately.

3.3.1 Determining the work offset

Prerequisite

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

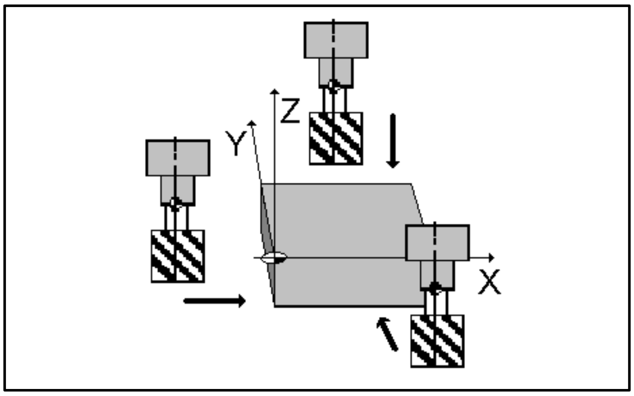


Fig. 3-14 Determining the work offset

Procedure

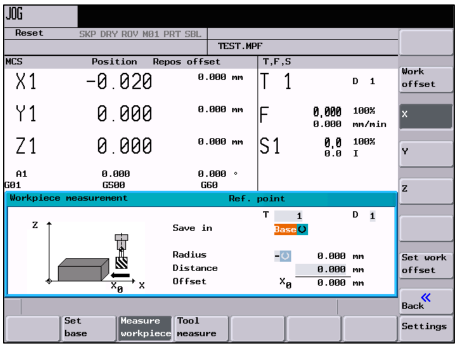
Measure workpiece

Select the ”Measure workpiece” softkey. The control system will switch to the ”Position” operating area and will open the dialog box for measuring the work offsets. The selected axis will appear as a softkey with a black background.

Then scratch the workpiece with the tool.

If scratching is not possible or if the desired point cannot be reached with the tool (for example, when using a spacer), the clearance between the tool and the workpiece surface must be entered in the **”Clearance”** field.

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



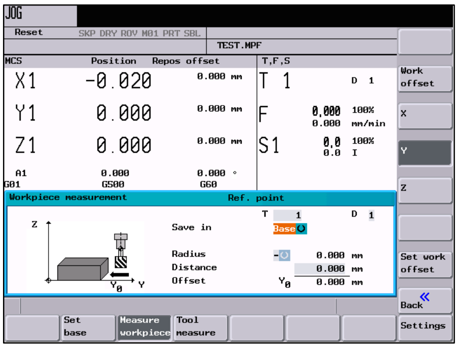


Fig. 3-15 *The* *Determine work offset* in X” screenform

The ”Determine work offset in Y” screenform

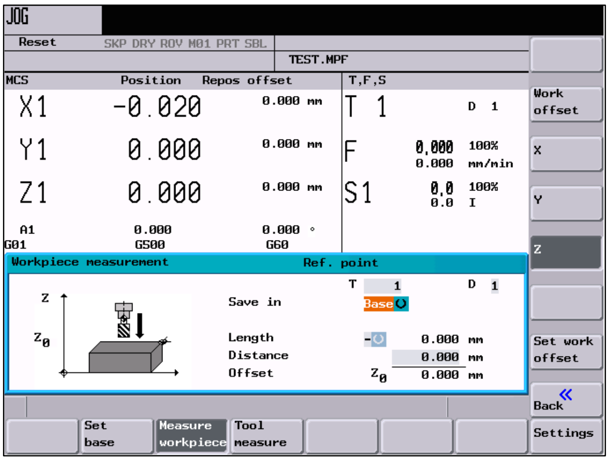


Fig. 3-16 The *Determine work offset* in *Zscreen*

Set work offset

Selecting this softkey will calculate the offset and display the result in the ”Offset” field.

3.4 Programming setting data - ”Parameter” operating area

Functionality

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

Operating sequences

Setting data

Select *Setting data* using the **Offset/Param** and **the Setting data** keys.

The **Setting data** softkey branches to another menu level where various control options can be set.

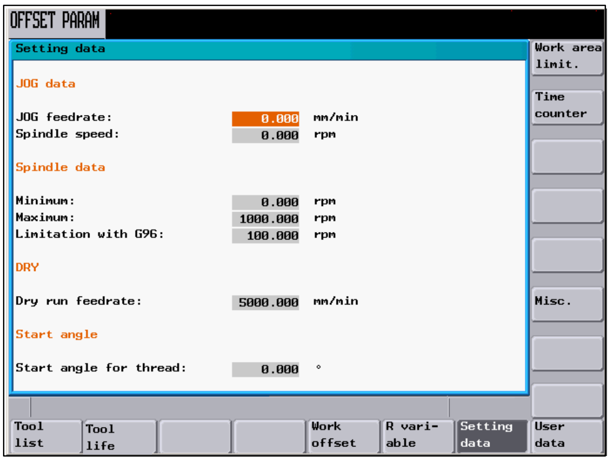


Fig. 3-17 The *Setting datastart screen*

JOG feedrate

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

Spindle

Spindle speed

Minimum / maximum

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

Programmed (limitation)

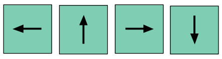
Programmable upper speed limitation (LIMS) at constant cutting rate (G96).

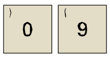
Dry run feed (DRY)

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

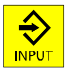
Start angle for thread cutting (SF)

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





Position the cursor bar on the input field you want to change and enter the value(s).



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

Softkeys

Work area limit.

The working area limitation is active with geometry and additional axes. Enter the values for the work area limitation. Selecting the **Set Active** softkey will activate / deactivate the values for the axis highlighted by the cursor.

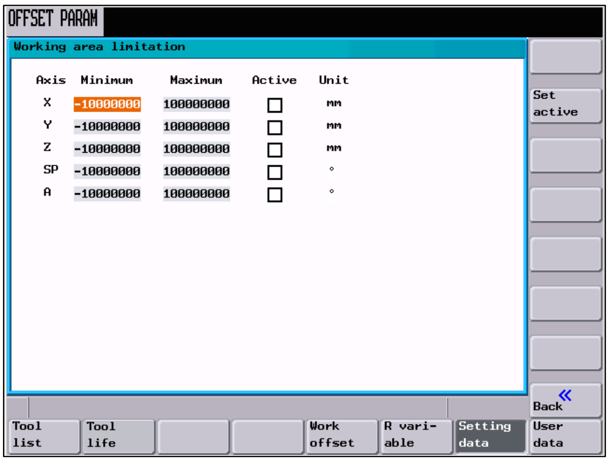


Fig. 3-18

Time counter

Timers Counters

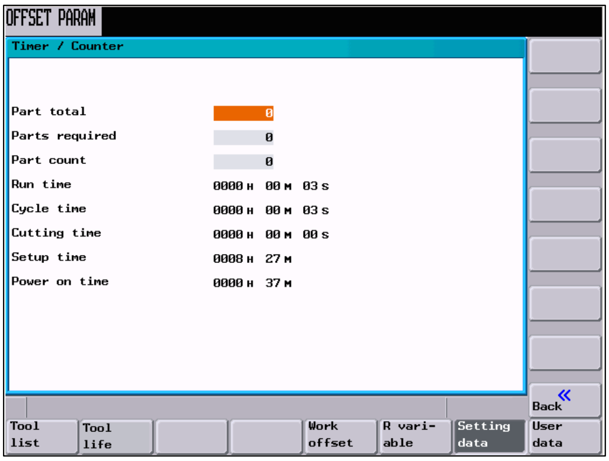


Fig. 3-19

Meaning:

- Parts required: Number of workpieces required (require number of workpieces)
- Parts total: Number of workpieces produced in total (actual total)
- Part count: This counter registers the number of all workpieces produced since the starting time.
- Run time: Total runtime of NC programs in the AUTOMATIC mode(in seconds)

  In the AUTOMATIC mode, the runtimes of all programs between NC START and end of program / RESET are summed up. The timer is zeroed with each power–up of the control system. Runtime of the selected NC program (in seconds)
- Cycle time: Tool action time (in seconds)

  The runtime between NC START and end of program / RESET is measured in the selected NC program. The timer is reset with starting a new NC program.
- Cutting time

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

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

Misc

Use this function to display all setting data for the control system in the form of a list. The data are divided into

- general
- axis–specific and
- channel setting data.

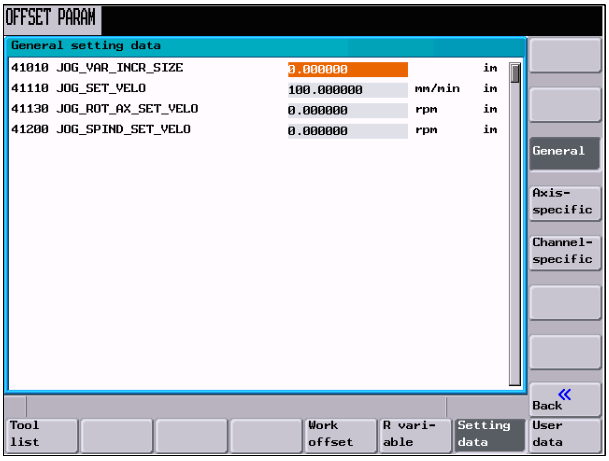


Fig. 3-20

3.5 R parameters – ”Offset/Parameter”operating area

Functionality

The **R parameters** start screen displays all R parameters existing in the control system in the form of a list (see also Section 8.9 ”R parameters”).
These can be changed as necessary.

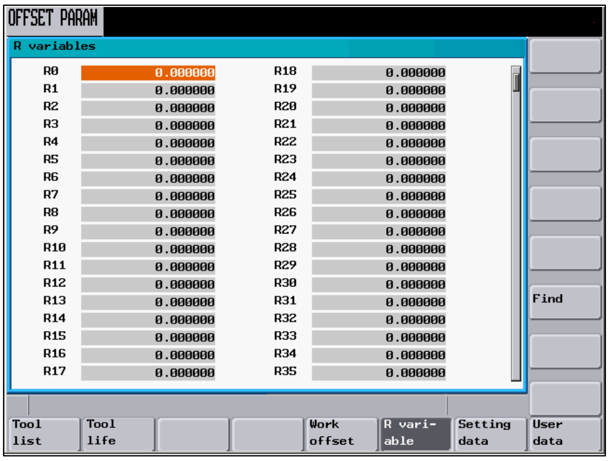


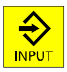
Fig. 3-21 The ”R parameters” window

Operating sequence

Use the **variable** and the **R variablesoftkeys**

R vari–
able

to position the cursor bar on the input field you want to change and enter the values.



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

4 Manually Controlled Mode

Preliminary remark

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

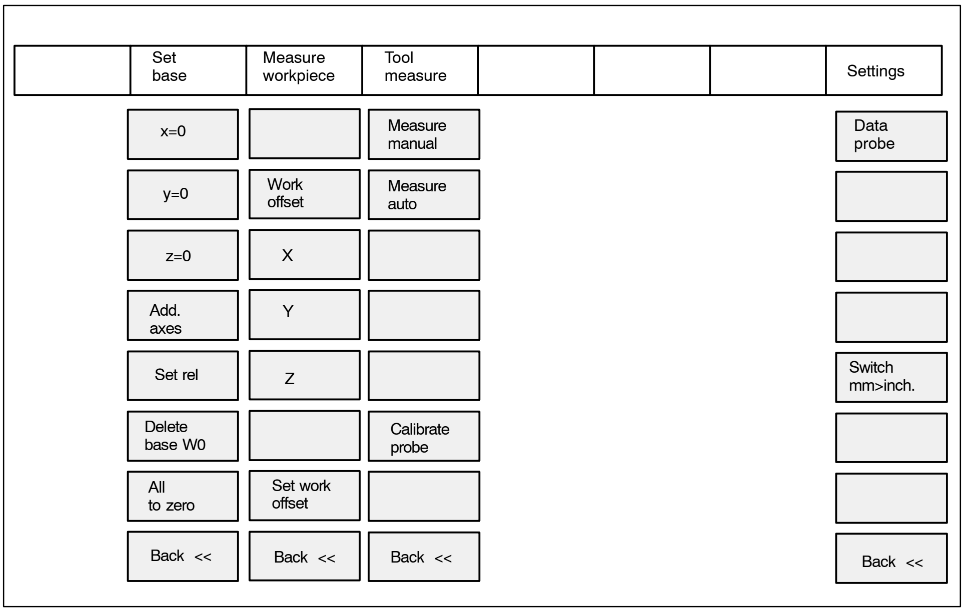


Fig. 4-1 Menu tree for the JOG mode, ”Position” operating area

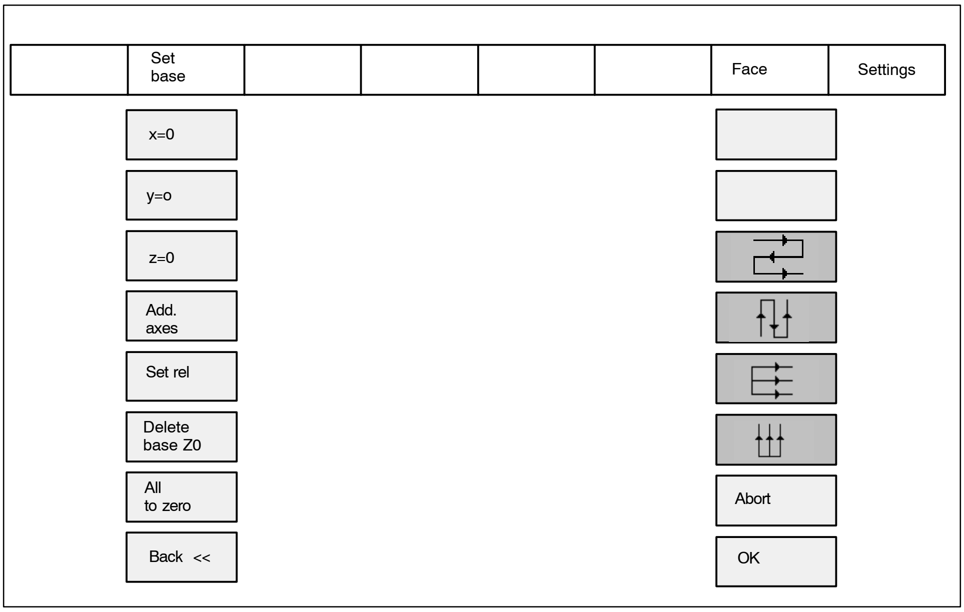
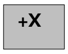


Fig. 4-2 Menu tree for the MDA mode, ”Machine” operating area

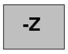
4.1 JOG mode - ”Position” operating area

Operating sequences

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

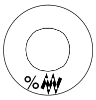


...

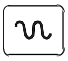


To traverse the axes, press the appropriate key of the X, Y or Z axis.

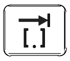
The axes will traverse continuously at the velocity stored in the setting data until the key is released. If the value of the setting data is zero, the value stored in the machine data is used.



If necessary set the velocity using the override switch.



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



In the **Jog** mode, you can traverse the axes by adjustable increments using the same operating sequence. The set number of increments is visualized in the display area. To deselect the Jog mode, press **Jog** once more.

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

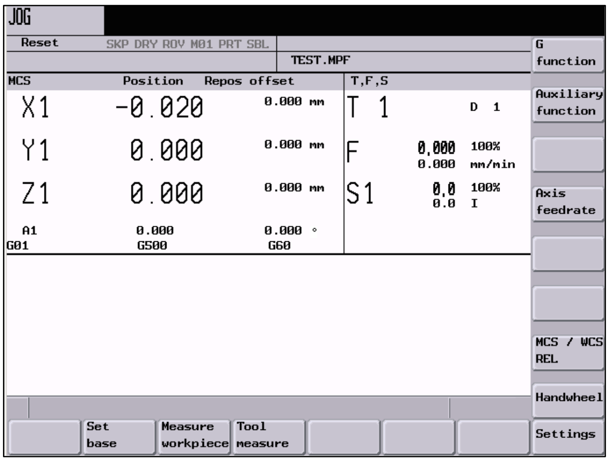


Fig. 4-3 The ”Jog” start screen

Parameters

Table 4-1 Description of the parameters in the JOG start screen

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

Note

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

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

The master spindle is displayed:

- in the idle condition;
- when starting the spindle;
- if both spindles are active.

The workspindle is displayed:

- when starting the workspindle.

The power bar applies to the spindle currently active.

Softkeys

Set base

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

The following subfunctions are provided:

- Direct input of the desired axis position
  In the input window, position the input cursor on the desired axis; thereafter, enter the new position. Then, press **Input** or move the cursor to confirm your input.
- Setting of all axes to zero
  The **X=Y=Z=0** softkey function overwrites the current position of the appropriate axis with zero.
- Setting of individual axes to zero
  Use the **X=0 Y=0** or **Z=0** softkey to overwrite the current position with zero.
  Any additional axes must only be set to zero if the X, YX and Z geometry axes required for milling have been configured.

Use the Set rel softkey to switch the display to the relative coordinate system. Any subsequent inputs will change the reference point in this coordinate system.

Note

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

Measure workpiece

Use this softkey to determine the work offset (cf. Chapter 3)

Tool measure

Use this softkey to measure the tool offsets (cf. Chapter 3)

Settings

The interactive screenform shown below is intended to set the retraction plane, the safety clearance and the direction of rotation of the spindle for automatically generated part programs in the MDA mode (see Section4.2.1). Furthermore, the values for the JOG feedrate and the variable size of increments can be set.

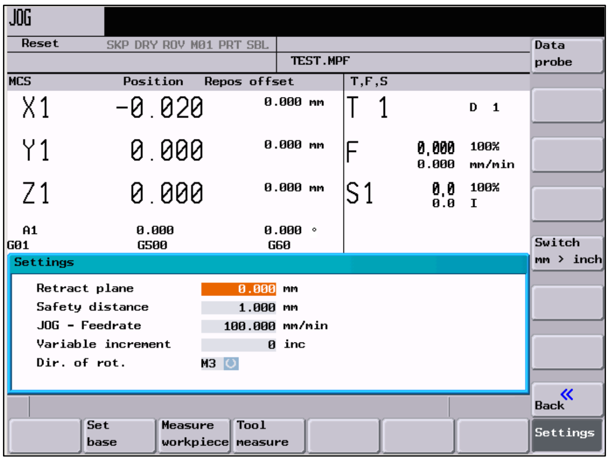


Fig. 4-4

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

Safety distance: Safety clearance to the workpiece surface
This value defines the minimum distance between the workpiece surface and the workpiece. It is used by the ”Face” and ”Automatic tool gauging” functions.

JOG feedrate: Feedrate value in the JOG mode

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

Switch mm > inch

Use this softkey to switch between the metric and the inch system.

4.1.1 Assigning handwheels

Operating sequence

Hand wheel

Use this softkey to display the **handwheel** window in the *Jog* mode.

After the window has been opened, all axis identifiers are displayed in the ”Axis” column, which simultaneously appear in the softkey bar. Depending on the number of handwheels connected, you can switch from handwheel 1 to handwheel 2 or 3.

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

The ☑.

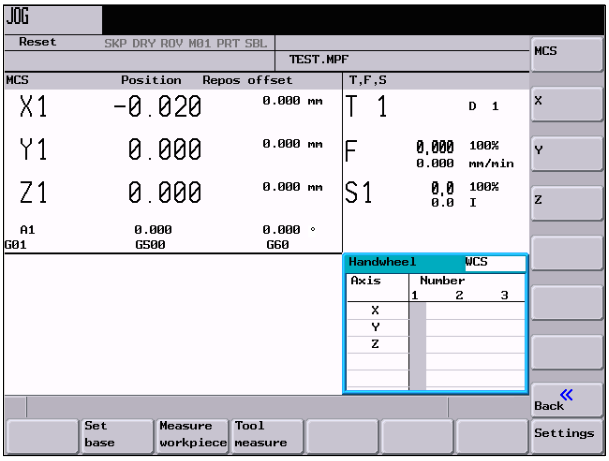


Fig. 4-5 The *Handwheel menu screen*

MCS

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

4.2 MDA mode (Manual input) - ”Machine” operating area

Functionality

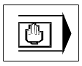
In the **MDA** mode, you can create or execute a part program.

Caution

The Manual mode is subject to the same safety interlocks as the fully automatic mode.

Furthermore, the same prerequisites are required as in the fully automatic mode.

Operating sequences



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

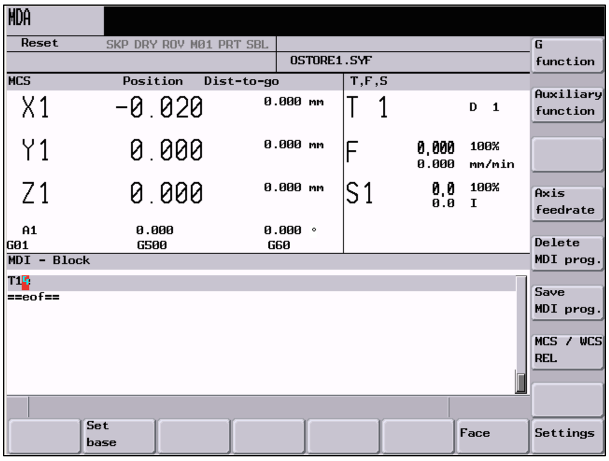
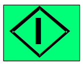


Fig. 4-6 The ”MDA” start screen

Enter one or several blocks using the keyboard.



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

After machining, the contents is preserved so that the machining can be repeated by pressing NC START once more.

Parameters

Table 4-2 Description of the parameters in the **MDAworking window**

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

Note

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

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

The master spindle is displayed:

- in the idle condition;
- when starting the spindle;
- if both spindles are active.

The workspindle is displayed:

- when starting the workspindle.

The power bar applies to the spindle currently active.

Softkeys

Set base	Use this softkey to set the base work offset (see Section 4.1).
Face	Face milling (see also Section 4.2.1)
Settings	see Section 4.1
G function	The G function window displays G functions whereby each G function is assigned to a group and has a fixed position in the window. Use the **PageDown** and **PageUp** keys to display further G functions. Selecting the softkey repeatedly will close the window.
Auxiliary function	This window displays the auxiliary and M functions currently active. Selecting the softkey repeatedly will close the window.
Axis feedrate	Use this softkey to display the *Axis feedrate* window. Selecting the softkey repeatedly will close the window.
Delete MDI prog.	Use this function to delete blocks from the program window.
Save MDI prog.	Enter a name in the input field with which you wish the MDA program to be saved in the program directory. Alternatively, you can select an existing program from the list. To switch between the input field and the program list, use the TAB key.

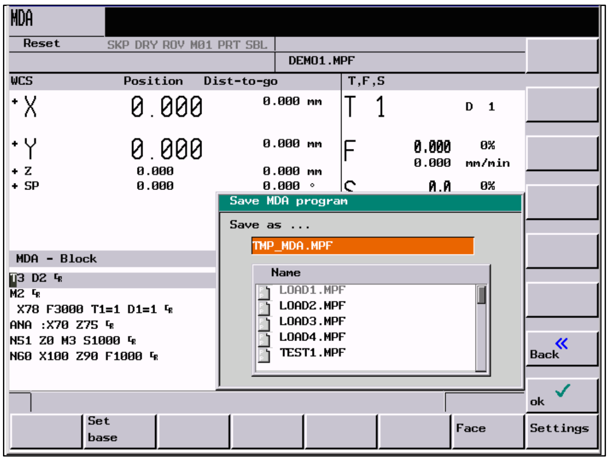


Fig. 4-7

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

4.2.1 Face milling

Functionality

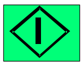
Use this function to prepare a blank for the subsequent machining without creating a special part program.

Operating sequence

Face

In the **MDA** mode, select the **Face** softkey to open the interactive screenform.

- Position the axes on the start point.
- Enter the values in the screenform.



After you have filled out the screenform completely, the function will create a part program which can be started with **NC START**. The interactive screenform will be closed, and ”Machine” start screen will appear. Here you can observe the program progress.

Important

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

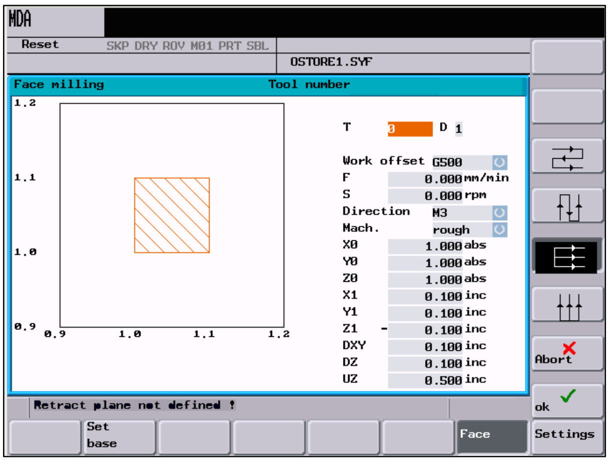
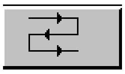


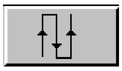
Fig. 4-8 Face milling

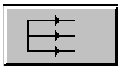
Table 4-3 Description of the parameters in the **Face milling** working window

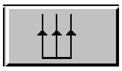
Parameter	Explanation
Tool	Input of the tool to be used The tool is loaded prior to machining. To this end, the function calls a working cycle performing all steps required. This cycle (LL6) is provided by the machine manufacturer.
Work offset	Work offset (formerly called ”zero offset” – transl.) to be selected in the program
Feed F	Input of the path feedrate, in mm/min or mm/rev.
Spindle S r.p.m.	Input of the spindle speed
Direction	Use this softkey to select the direction of rotation of the spindle.
Mach.	Definition of the surface quality You can select between roughing and finishing.
X0, Y0, Z0, X1, Y1 Blank dimensions	Use this softkey to enter the geometry of the workpiece.
Z1 Finished dimension	Finished dimension in Z
DXY Max. infeed	Input field for the amount of the infeed motion (X, Y)
DZ Max. infeed	Input field for the amount of the infeed motion (Z)
UZ	Input field for the stock allowance when roughing

Softkeys for specifying the stock removal strategy (opposite / synchronous run)


Machining parallel to the abscissa, with changing direction


Machining parallel to the ordinate, with changing direction


Machining parallel to the abscissa, in one direction


Machining parallel to the ordinate, in one direction

5 Automatic Mode

Prerequisite

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

Operating sequence

Select the **AUTOMATIC** mode using the **AUTOMATIC** key on the machine control panel.

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

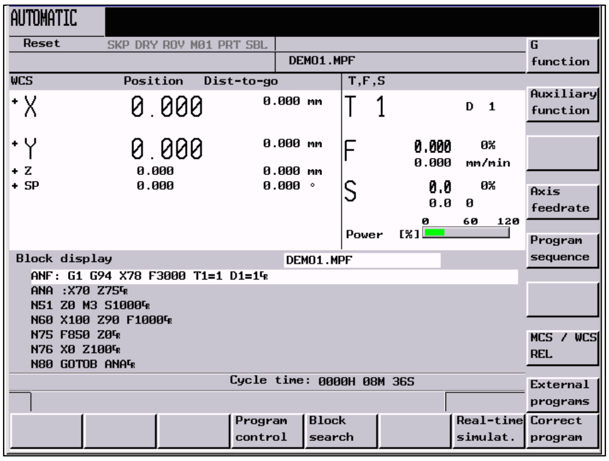


Fig. 5-1 The *AUTOMATICstart screen*

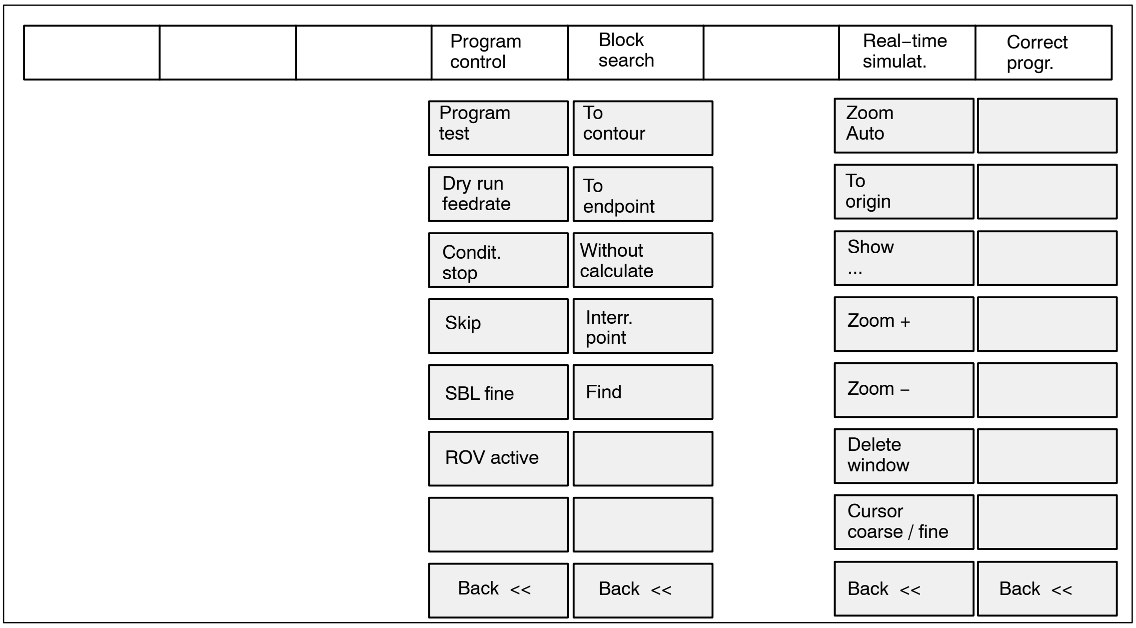


Fig. 5-2 Menu tree of the *AUTOMATIC* mode

Parameters

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

Parameter	Explanation
MCS X Z	Displays the existing axes in the MCS or WCS
+ X - Z	If you traverse an axis in the positive (+) or negative (-) direction, a plus or minus sign will appear in the relevant field. If the axis is already in the required position, no sign is displayed.
Position mm	These fields display the current position of the axes in the MCS or WCS.
Distance to go	These fields display the current position of the axes in the MCS or WCS.
G function	Displays important G functions
Spindle S r.p.m.	Displays the actual value and the setpoint of the spindle speed
Feed F mm/min or mm/rev.	Displays the path feedrate actual value and setpoint
Tool	Displays the currently active tool with the current edge number (T..., D...).
Current block	The block display displays seven subsequent blocks of the currently active part program. The display of one block is limited to the width of the window. If several blocks are executed quickly one after the other, it is recommended to switch to the ”Program progress” window. To switch back to the seven–block display, use the ”Program sequence” softkey.

Note

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

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

The master spindle is displayed:

- in the idle condition;
- when starting the spindle;
- if both spindles are active.

The workspindle is displayed:

- when starting the workspindle.

The power bar applies to the spindle currently active.

Softkeys

Program control	The program control softkeys are displayed (e.g. "Skip block", "Program test").
Program test	If "Program test" (PRT) is selected, the output of setpoints to axes and spindles is disabled. The setpoint display will "simulate" the traversing motion.
Dry run feedrate	If you select this softkey, all traversing motions will be performed with the feedrate setpoint specified via the "Dry run feed" setting data. In other words: Instead of the programmed motion commands, the dry run feedrate will act.
Condit stop	If this function is active, the program execution is stopped at the blocks in which the miscellaneous function M01 is programmed.
Skip	Program blocks marked with a slash in front of the block number are skipped during the program execution (e.g. "/N100").
SBL fine	If this function is enabled, the part program blocks are executed separately as follows: Each block is decoded separately, and a stop is performed at each block; an exception are only the thread blocks without dry run feedrate. In such blocks, a stop is only performed at the end of the current thread block. "Single Block fine" can only be selected in the RESET status.
ROV active	The feedrate override switch will also act on the rapid traverse override.
Back <<	Use this softkey to quit the screenform.
Block Search	Use the block search function to go to the desired place in the program.
To contour	Forward block search forward to the block start point with calculation During the block search, the same calculations are carried out as during normal program operation, but the axes do not move.

To endpoint

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

Without calculate

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

Interr. point

The cursor is positioned on the interruption point.

Find

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

Real–time simulat.

Broken–line graphics are displayed to trace the programmed tool path while the workpiece is being machined on the machine (see also Section 6.4)

Note: Whether or not this function is implemented is decided by the machine manufacturer and performed via parameterization.

Correct progr.

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

G funct

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

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

Use the **PageDown** and **PageUp** keys to display further G functions.

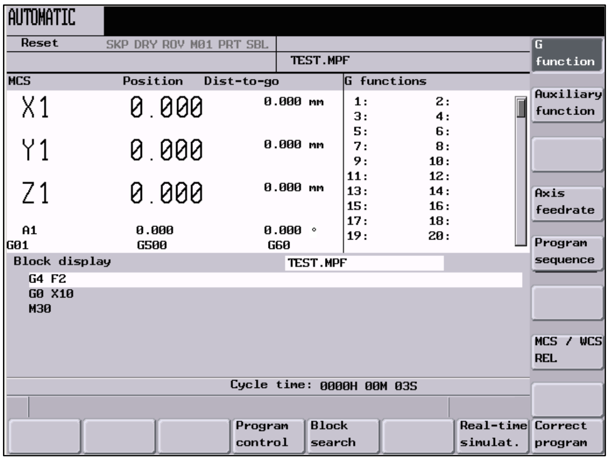


Fig. 5-3   The *Active G functionswindow*

Auxiliary function

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

Axis feedrate

Use this softkey to display the *Axis feedrate window.*
Selecting the softkey repeatedly will close the window.

Program sequence

Use this softkey to switch from the seven–block to the three–block display.

MCS/WCS REL	Use this softkey to select the machine coordinate system, the workpiece or the relative coordinate system.
External programs	Use this softkey to transmit an external program to the control system via the RS232 interface; to execute this program, press **NC START**.

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

Functionality

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

Operating sequence

Select the **AUTOMATIC** mode using the **AUTOMATIC** key on the machine control panel.

PROGRAM MANAGER

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

Position the cursor bar on the desired program.

Execute

To select the program for execution, use the **Execute** softkey. The name of the selected program will appear in the ”Program name” screen line.

Progr. control

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

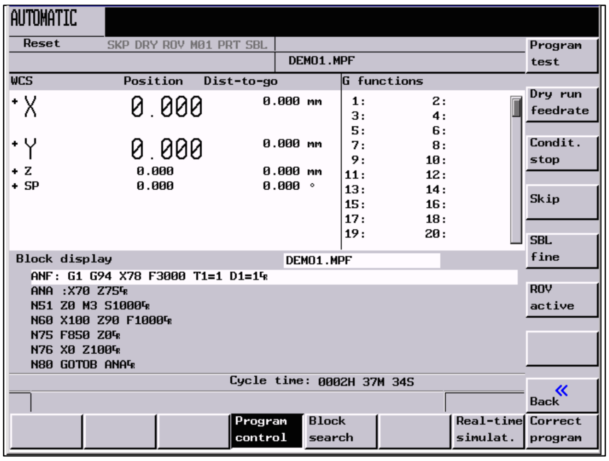


Fig. 5-4 Program control

Press **NC START** to start the part program execution.

5.2 Block search - ”Machine” operating area

Operating sequence

Prerequisite: The required program has already been selected (cf. Section 5.1) and the control system is in the RESET condition.

Block Search

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

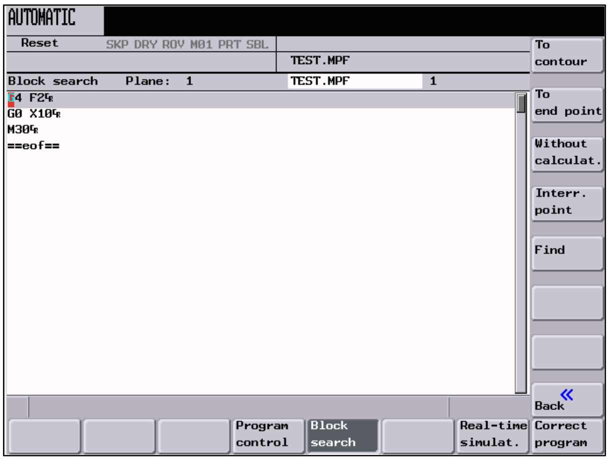


Fig. 5-5 Block search

To contour — Block search to the block start

To end point — Block search to the end of the block

Without calculate — Block search without calculation

Interr. point — The interruption point is loaded.

Find — Selecting this softkey opens a dialog box will where you can enter a line number or terms you are looking for.

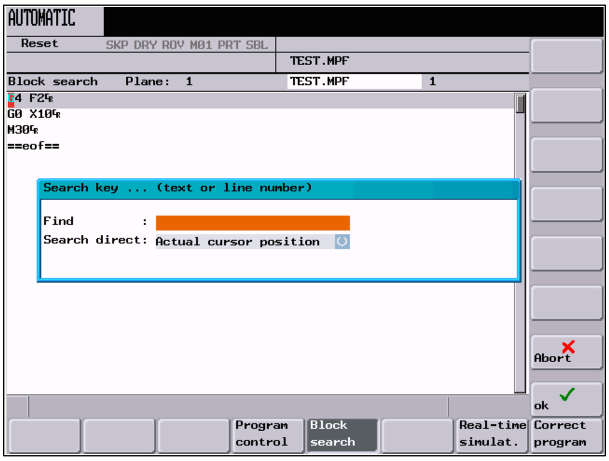


Fig. 5-6 Entering the searched term

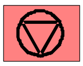
A toggle field is provided to define from which position you will search for the term.

Search result

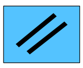
The required block is displayed in the *Current blockwindow.*

5.3 Stopping / canceling a part program

Operating sequence



Press **NC STOP** to cancel a part program.
Press **NC START** to continue the program execution.



Use **RESET** to interrupt the program currently running.
Pressing **NC START** again will restart the program you have interrupted and execute the program from the beginning.

5.4 Reapproach after cancellation

After a program cancellation (NC STOP), you can retract the tool from the contour in the Manual mode (**Jog**).

Operating sequence

	Select the **AUTOMATIC** mode.
BlockSearc h	Use this softkey to open the *Block search* window for loading the interruption point.
Interr. Point	The interruption point is loaded.
To contour	Selecting this softkey will start the block search to the interruption point. An adjustment to the start position of the interrupted block will be carried out.
	Press **NC START** to continue the program execution.

5.5 Repositioning after interruption

After a program interruption (**NC STOP**), you can retract the tool from the contour in the Manual mode (**Jog**); the coordinates of the interruption point are stored by the control system. The path differences traversed by the axes are displayed.

Operating sequence

Select the **AUTOMATIC** mode.

Press **NC START** to continue the program execution.

Caution

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

5.6 Program execution from external

Functionality

Use this softkey to transmit an external program to the control system via the RS232 interface; to execute this program, press **NC START**.

While the contents of the buffer memory are being processed, the blocks are reloaded automatically. For example, a PC with the PCIN tool installed for data transfer can be used as the external device.

Operating sequence

Prerequisite: The control system is in the RESET condition.
The RS232 interface is parameterized correctly (for the relevant text format, see also Chapter 7) and not occupied by any other application (DataIn, DatatOut, STEP7).

External programs

Select the softkey.

On the external device (PC), activate the relevant program for data output via the PCIN tool.

The program is transmitted into the buffer memory and selected and displayed in the Program Selection automatically.
Advantageous for the program execution: Wait until the buffer memory is filled.

Press **NC START** to start the program execution. The program is reloaded continuously.

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

Note

Any transmission errors are displayed in the **System** / **Data I/O** area if you select the **Error log** softkey.

Block search is not possible for programs read in from an external source.

6

Part Programming

Operating sequence

Press the **Program Manager** key to call the Program Manager.

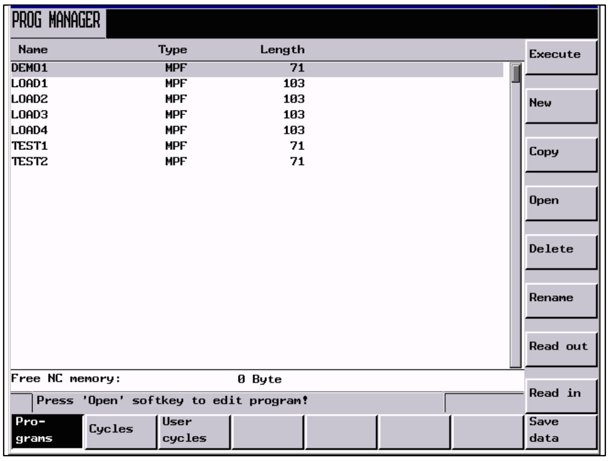


Fig. 6-1 The "Program Manager" start screen

Use the cursor keys to navigate in the program directory. To quickly find the required program, type the 1st letter of the program name you are looking for. The control system will automatically position the cursor on a program with matching characters.

Softkeys

Programs	Use this softkey to display all files contained in the part program directory.
Execute	Use this softkey to select the program on which the cursor is positioned for execution. The control system will switch to the position display. With the next **NC START**, the program is started.
New	Use the **New** softkey to create a new program.
Copy	Use the **Copy** softkey to copy the selected program into another program with a new name.
Open	Use the "Open" softkey to open the file highlighted by the cursor for processing.
Delete	Use this softkey to delete either only the program highlighted by the cursor or all part programs; first, however, a warning confirmation is displayed. Use the **OK** softkey to execute the deletion order and **Abort** to discard.
Rename	Selecting the **Rename** softkey opens a window where you can rename the program you have selected beforehand using the cursor. After you have entered the new name, either press **OK** to confirm or **Abort** to cancel.
Read out	Use this softkey to saved files via the RS232 interface.
Read in	Use this softkey to load part programs files via the RS232 interface. For the settings of the interface, please refer to the **System** operating area (Chapter 7). The part programs must be transmitted using the text format.
Cycles	Use the **User cycles** softkey to display the "Standard cycles" directory. This softkey will only appear unhidden if you have the relevant access right.
Delete	Use this softkey to delete the cycle highlighted by the cursor; first, a confirmation warning will appear.
User cycles	Use the **User cycles** softkey to display the "User cycles" directory. With the appropriate access right, the softkeys **New, Copy, Open, Delete, Rename, Read out** and **Read in** are displayed.

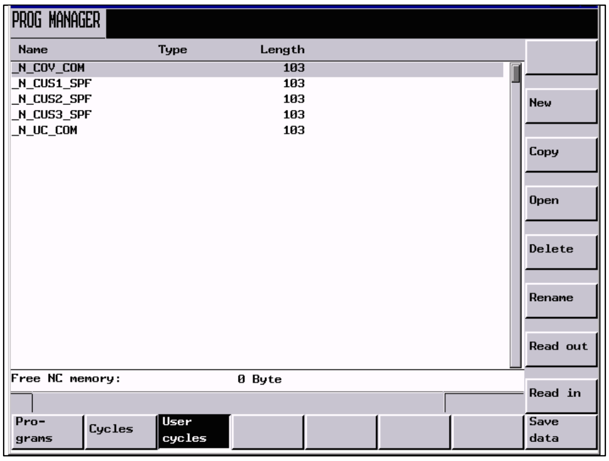


Fig. 6-2

Save data

Save data
This function is used to save the contents of the volatile memory into a non−volatile memory area.
Prerequisite: There is no program currently executed.
Do not carry out any operator actions while the data backup is running!

6.1 Entering a new program - ”Program” operating area

Operating sequences

PROGRAM MANAGER | Programs

You have selected the **Program Manager** operating area and you are in the overview of the NC programs already created.

New

Select the **New** softkey; a dialog box will appear where you can enter the name of the new main program or subroutine. The extension for main programs ”.MPF” is entered automatically; the extension for subroutines ”.SPF” must be entered together with the program name.

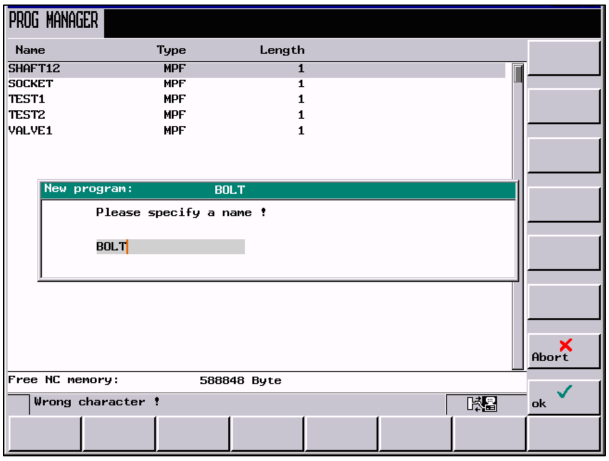


Fig. 6-3 The *New program interactive screenform*

A J | W Z

Enter the name for the new program.

ok

Use the **OK** softkey to confirm your input. The new part program file will be created, and the editor window is opened automatically.

Abort

Use **Abort** to cancel the creation of the program; the window will be closed.

6.2 Editing part programs - "Program" operating area

Functionality

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

Any modifications to the part program are stored immediately.

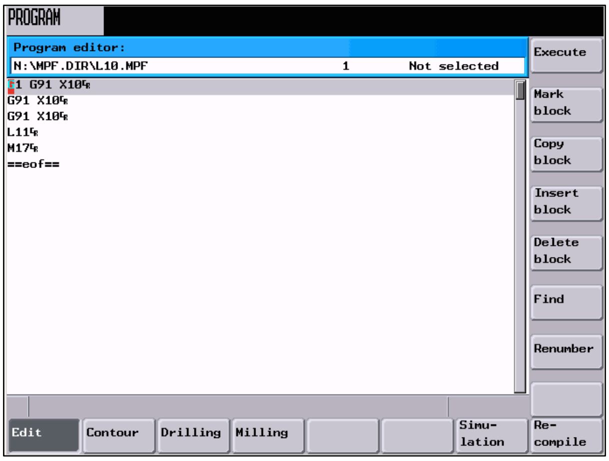


Fig. 6-4 The "Program editor" start screen

Menu tree

Edit	Contour	Drilling	Milling			Simulation	Recompile
Execute		Drilling centering				Zoom Auto	
Mark block		Center drilling	Face milling			To origin	
Copy block		Deep hole drilling	Contour milling			Show ...	
Insert block		Boring				Zoom +	
Delete block		Tapping	Standard pockets			Zoom –	
Find		Deselect modal	Grooves			Delete window	
Renumber		Hole pattern	Thread milling			Cursor crs./fine	

Fig. 6-5 The "Program" menu tree

Softkeys

Edit	Use this function to edit text segments.
Execute	Use this softkey to execute the selected file.
Mark block	Use this function to highlight a text section from the current cursor position. When doing so, use the arrow keys.
Copy block	Use this softkey to copy a selected block to the clipboard
Insert block	Use this softkey to paste a text from the clipboard at the current cursor position
Delete block	Use this softkey to delete a selected text
Find	Use the Find softkey to search for a string in the program file displayed. Type the term you are looking for in the input line and use the **OK** softkey to start the search. Use "Abort" to close the dialog box without starting the search process.
Renumber	Use this softkey to replace the block numbers from the current cursor position up to the program end.
Contour	For programming the contour ("blueprint programming"), see Section 6.3
Drilling	see Section "Cycles"
Milling	see Section "Cycles"
Simulation	The simulation is described in Section 6.4.
Recompile	For recompilation, position the cursor on the cycle calling line in the program. This function decodes the cycle name and prepares the screenform with the relevant parameters. If there are any parameters beyond the range of validity, the function will automatically use the default values. After closing the screenform, the original parameter block is replaced by the corrected block. **Please note:** Only automatically generated blocks can be recompiled.

6.3 Blueprint programming

Functionality

The control system offers various contour screenforms for the fast and reliable creation of part programs. Fill out the relevant parameters in the interactive screenforms.

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

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

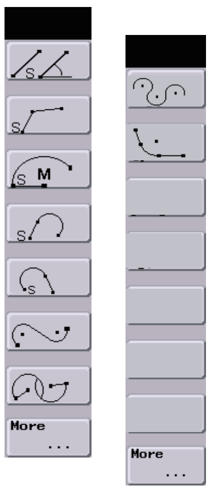


Fig. 6-6 Softkey functions

The coordinates can be input either as an absolute, incremental or polar value. Input is switched using the Toggle key.

Softkeys

Use these softkey functions to branch into the individual contour elements.

If a contour screenform is opened for the first time, the starting point of the contour section must be reported to the control system. All subsequent motions will refer to this point. If you move the input bar using the cursor, all values must be reentered.

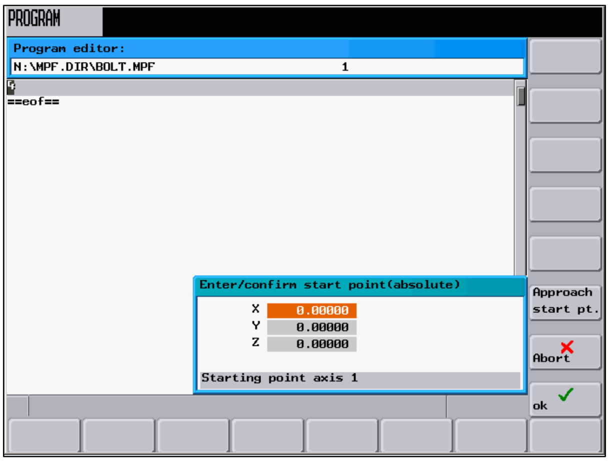
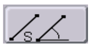


Fig. 6-7 Setting the starting point

The **Approach start point** softkey function will generate an NC block approaching the entered coordinates.



Programming aid for the programming of straight line sections

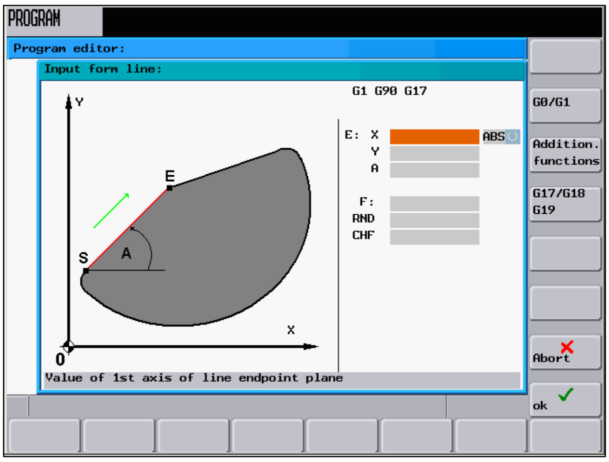


Fig. 6-8

Enter the end point of the straight line in absolute dimensions (ABS), in incremental dimensions (INC) (with reference to the starting point), or in polar coordinates (POL). The current settings are displayed in the interactive screenform.

The end point can also be defined by a coordinate and the angle between an axis and the straight line.

If you are using polar coordinates to determine the end point, it is imperative to specify the length of the vector between the pole and the end point (in field 1), as well as the angle of the vector referred to the pole (to be entered in field 2).
The prerequisite is that a pole was set beforehand. This pole will be applicable until a new pole is set.

Set Pole

A dialog box will appear where the coordinates of the pole point must be entered. The pole point will refer to the selected plane.

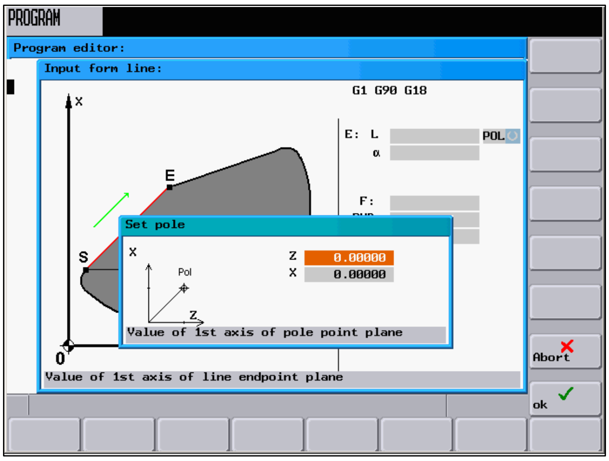


Fig. 6-9

G0/G1

If this function is selected, the selected block is traversed at rapid traverse or with the programmed path feedrate.

Add. functions

If necessary you can enter additional functions in the fields. The commands can be separated from each other by spaces, commas or semicolons.

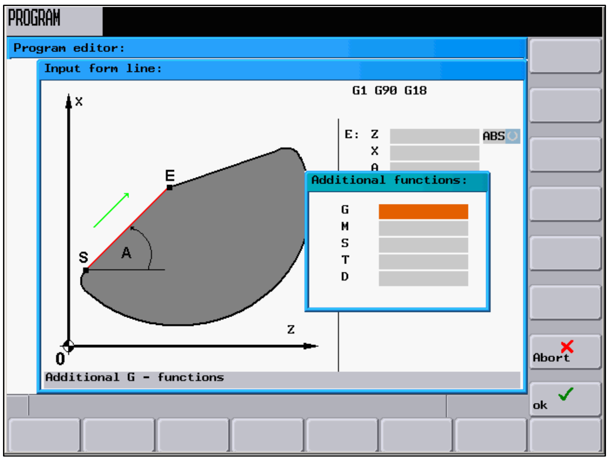


Fig. 6-10

This interactive screenform is provided for all contour elements.

G17/18/19

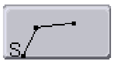
Use this softkey to select the relevant plane G17 (X–Y), G18 (Z–X) or G19 (Y–Z). The designations of the axes in the screenform will change according to the selection.

This interactive screenform is provided for all contour elements.

OK

Pressing the **OK** softkey will accept all commands into the part program.

Select **Abort** to quit the interactive screenform without saving the values.



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

Specify the coordinates of the end point of the second straight line and the angles of the straight lines.

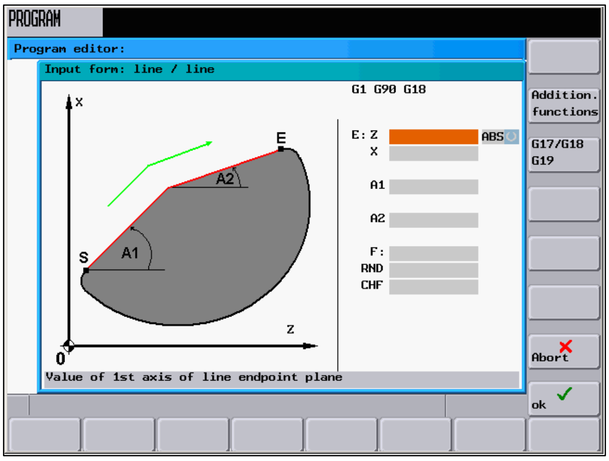
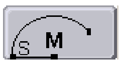


Fig. 6-11

Table 6-1 Input in the interactive screenform

End point of straight line 2	E	Enter the end point of the straight line.
Angle of straight line 1	A1	The angle is specified in the counterclockwise direction from 0 to 360 degrees.
Angle of straight line 2	A2	The angle is specified in the counterclockwise direction from 0 to 360 degrees.
Feedrate	F	Feedrate



Use this interactive screenform to create a circular block using the coordinates end point and center point.

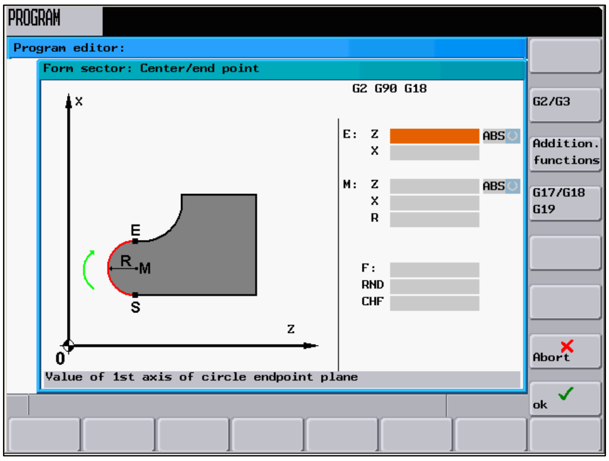


Fig. 6-12

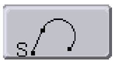
Enter the end point and center point coordinates in the input fields. Input fields no longer needed are hidden.

G2/G3

Use this softkey to switch the direction of rotation from G2 to G3. G3 will appear on the display. Pressing this softkey again will switch back the display to G2.

OK

Pressing the **OK** softkey will accept the block into the part program.



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

For calculating the points of intersection with any transition angles, the POI softkey function will display the center point coordinates.

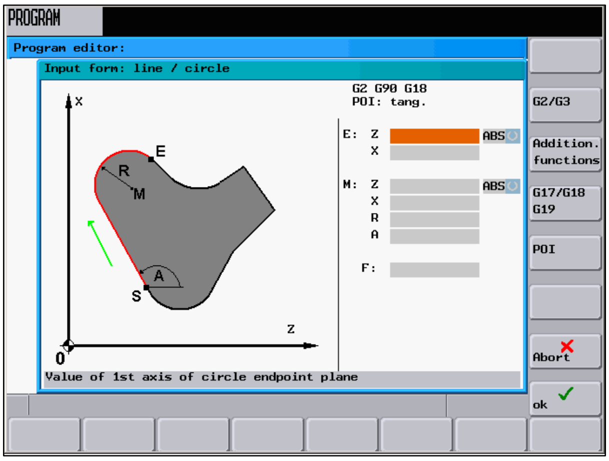


Fig. 6-13 Straight line – circle with tangential transition

Table 6-2 Input in the interactive screenform

End point of the circle	E	Enter the end point of the circle.
Angle of straight line	A	The angle is specified in the counterclockwise direction from 0 to 360 degrees.

Table 6-2 , cont'dInput in the interactive screenform

Radius of the circle	R	Input field for the circle radius
Feedrate	F	Input field for the interpolation feedrate
Center point of the circle	M	If there is no tangential transition between the straight line and the circle, the circle center point must be known. The specification is performed depending on the type of calculation (absolute, incremental or polar coordinates) selected in the previous block.

G2/G3

Use this softkey to switch the direction of rotation from G2 to G3. G3 will appear on the display. Pressing this softkey again will switch back the display to G2. The display changes to G2.

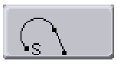
POI

You can choose between tangential or any transition.

The screenform generates a straight line and a circle block from the data you have entered.

If several points of intersection exist, the desired point of intersection must be selected from a dialog box.

If one coordinate was not entered, the program tries to calculate it from the existing specifications. If there are several possibilities, the coordinate must be selected from an appropriate dialog box.



This function will calculate the tangential transition between a contour 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.

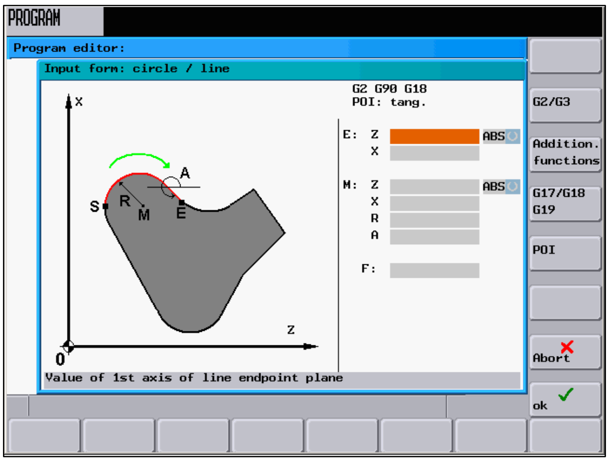


Fig. 6-14 Tangential transition

Table 6-3 Input in the interactive screenform

End point of straight line	E	Enter the end point of the straight line in absolute, incremental or polar coordinates.
Center point	M	Enter the center point of the circle in absolute, incremental or polar coordinates.
Radius of the circle	R	Input field for the circle radius

Table 6-3 , cont'dInput in the interactive screenform

Angle of straight line 1	A	The angle is specified in the counterclockwise direction from 0 to 360 degrees and with reference to the point of intersection.
Feedrate	F	Input field for the interpolation feedrate

G2/G3

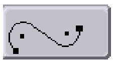
Use this softkey to switch the direction of rotation from G2 to G3. G3 will appear on the display. Pressing this softkey again will switch back the display to G2. The display changes to G2.

POI

You can choose between tangential or any transition.

The screenform generates a straight line and a circle block from the data you have entered.

If several points of intersection exist, the desired point of intersection must be selected from a dialog box.



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

Use the displayed screenform to enter the parameters center point and radius for the sector 1 and the parameters end point, center point and radius for the sector 2. Furthermore, the direction of rotation of the circles must be selected. A help screen is provided to display the current settings.

Pressing OK calculates three blocks from the entered values and inserts them into the part program.

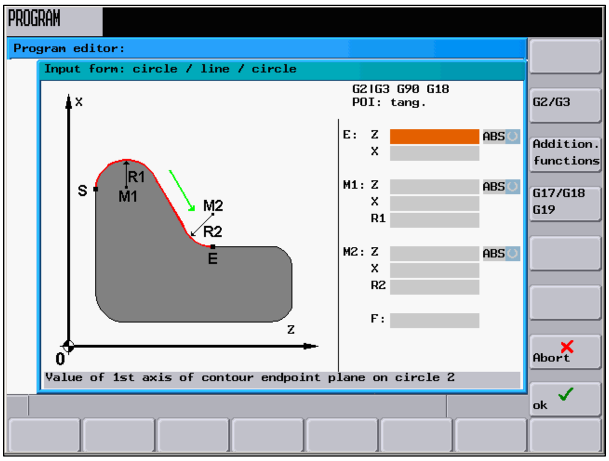


Fig. 6-15

Table 6-4 Input in the interactive screenform

End point	E	1. and 2nd geometry axes of the plane If no coordinates are entered, this function provides the point of intersection between the straight line you have inserted and sector 2.
Center point of the circle 1	M1	1st and 2nd geometry axes of the plane(absolute coordinates)
Radius of circle 1	R1	Input field for radius 1

Center point of circle 2	M2	1st and 2nd geometry axes of the plane(absolute coordinates)
Radius of circle 1	R2	Input field for radius 2
Feedrate	F	Input field for the interpolation feedrate

The screenform generates one straight line and two circle blocks from the data you have entered.

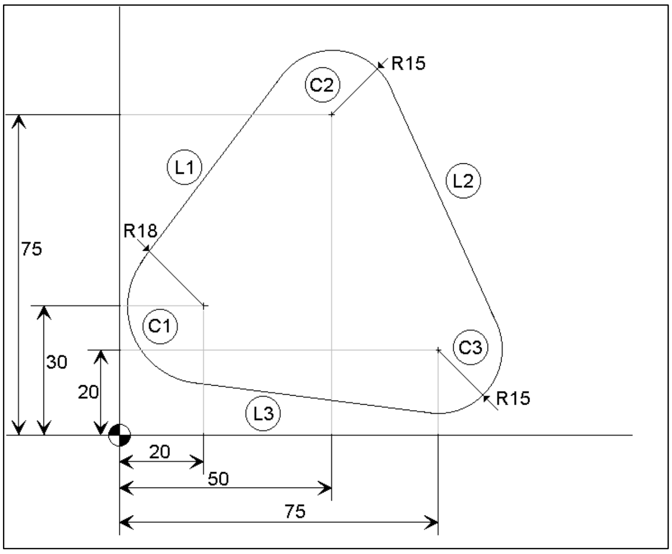
G2/G3

Use this softkey to define the direction of rotation of the two circle sectors. Possible combinations are:

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

The end point and the center point coordinates can be entered either in absolute or incremental dimensions or as polar coordinates. The current settings are displayed in the interactive screenform.

Example



Given:

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.

Procedure:

*In the **Contour** menu, select* *. A screenform will appear where you can enter the starting point.*

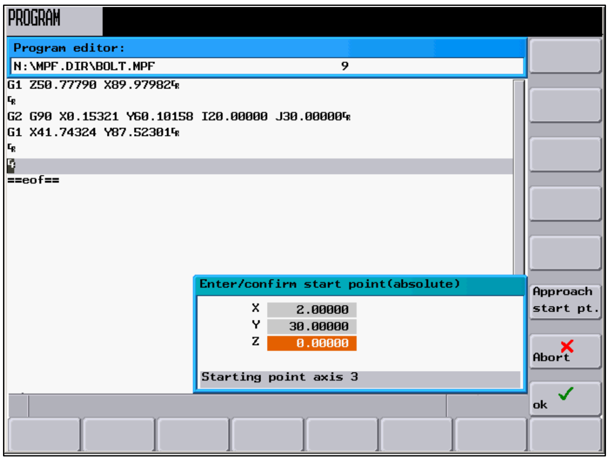


Fig. 6-16 Setting the starting point

*Press **OK** to confirm your input; the screenform for entering the values for the contour section* (C1) – (L1) – (C2) *will appear.*

*Use the **G2/G3** softkey to select the direction of rotation for the two circle sectors (G2|G3) and fill out the parameter list.*

The field for the end point can either be left open, or you can enter the points X 50 Y 90 (75 + R 15).

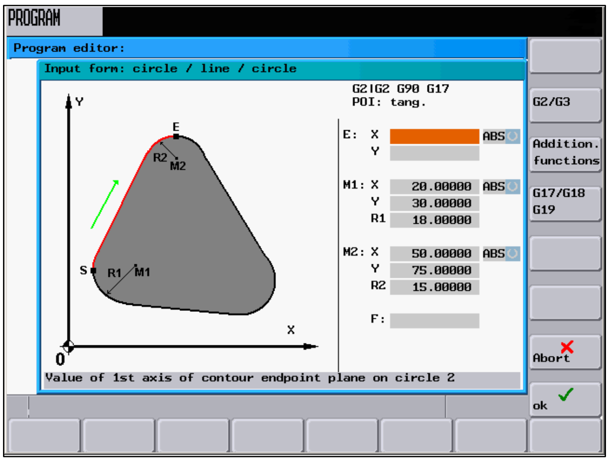


Fig. 6-17 Calling the screenform

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

Fig. 6-18 Result of step 1

Since the end point has been left open, the point of intersection of the straight line (L1) and the circle sector (C2) will be used as the starting point for the next contour definition.

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

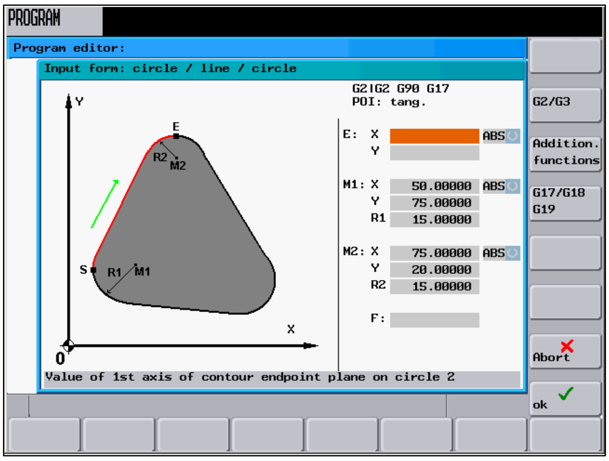


Fig. 6-19 Calling the screenform

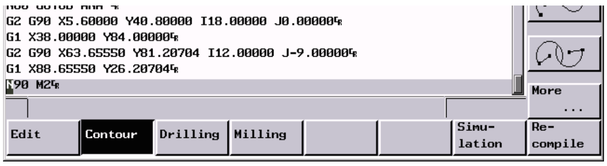


Fig. 6-20 Result of step 2

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

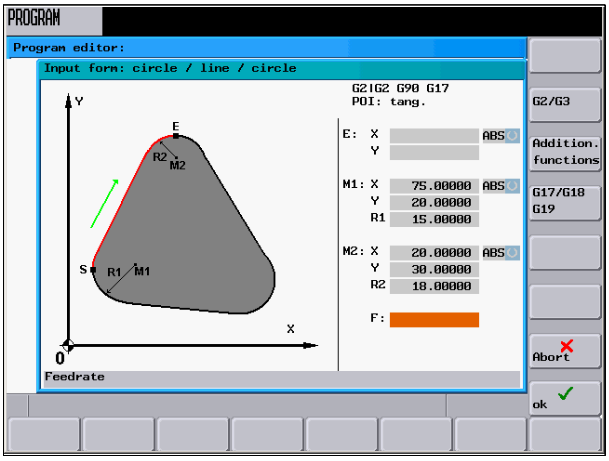


Fig. 6-21 Calling the screenform

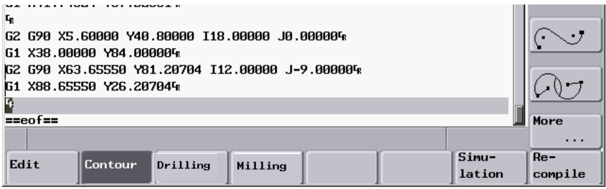


Fig. 6-22 Result of step 3

Subsequently, connect the new end point with the starting point. To do so, you can use the  *softkey.*

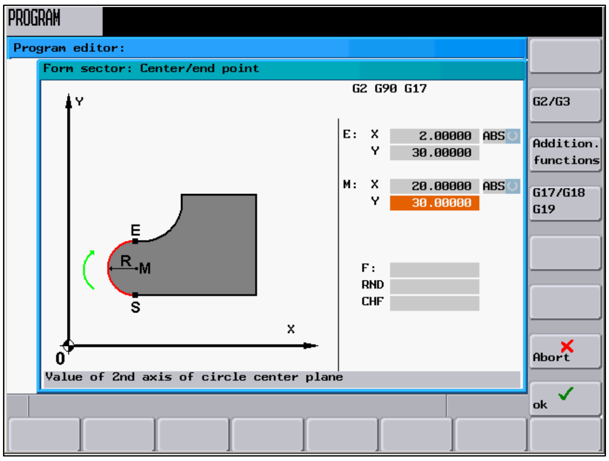
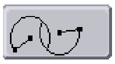


Fig. 6-23 Step 4

Fig. 6-24 Result of step 4



The function calculates the tangential transition between two circle sectors. Circle sector 1 must be described by the parameters starting point, center point and radius, and the circle sector 2 be described by the parameters end point and radius.

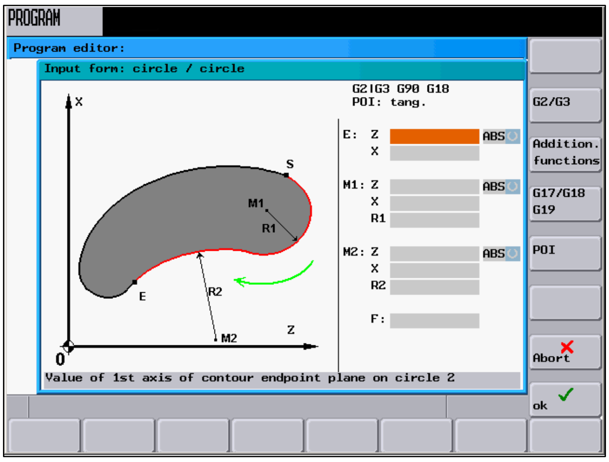


Fig. 6-25 Tangential transition

Table 6-5 Input in the interactive screenform

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

The specification of the points is performed depending on the type of calculation (absolute or incremental dimension or polar coordinates) selected beforehand. Input fields no longer needed are hidden. If any value is omitted when specifying the center point coordinates, the radius must be entered.

G2/G3

Use this softkey to switch the direction of rotation from G2 to G3. G3 will appear on the display. Pressing this softkey again will switch back the display to G2. The display changes to G2.

POI

You can choose between tangential or any transition.

The screenform generates two circle blocks from the data you have entered.

Selecting the point of intersection

If several points of intersection exist, the desired point of intersection must be selected from a dialog box.

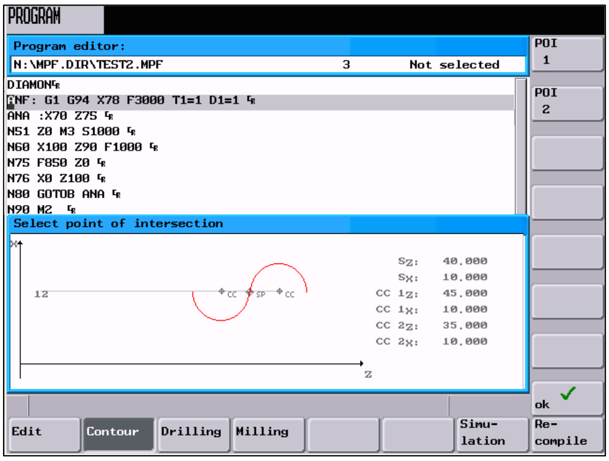


Fig. 6-26 Selecting the point of intersection

POI 1

The contour will be drawn using the point of intersection 1.

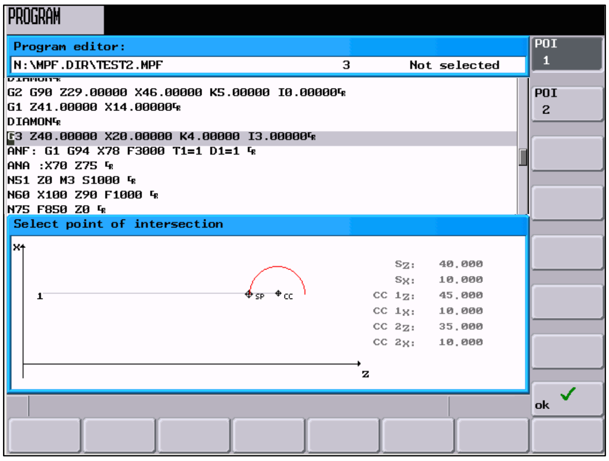


Fig. 6-27

POI 2

The contour will be drawn using the point of intersection 2.

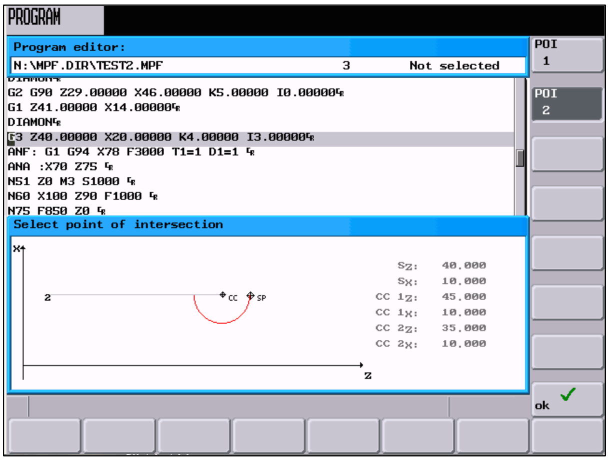


Fig. 6-28

OK

Pressing OK accepts the point of intersection of the displayed contour into the part program.

This function will insert a circle sector between two adjacent circle sectors. The circle sectors are described by their center points and circle radii, and the inserted sector is described only be its radius.

The operator is offered a screenform where he will enter the parameters center point, radius for circle sector 1 and the parameters end point, center point and radius for the circle sector 2. Furthermore, the radius for the inserted circle sector 3 must be entered and the direction of rotation be defined.

A help screen is provided to display the selected settings.

Pressing OK calculates three blocks from the entered values and inserts them into the part program.

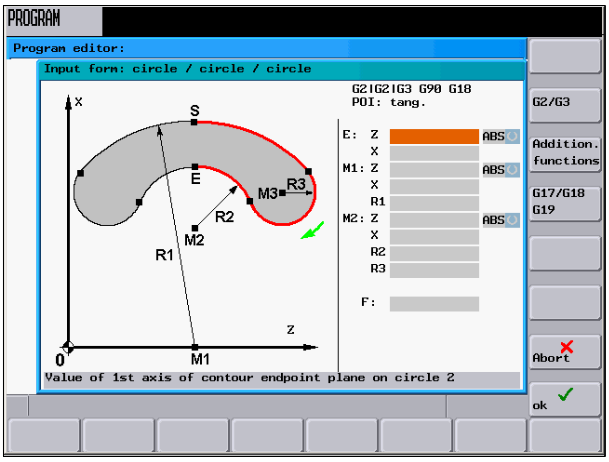


Fig. 6-29 Screenform for calculating the contour section circle–circle–circle

End point	E	1st and 2nd geometry axes of the plane If no coordinates are entered, this function provides the point of intersection between the circle sector you have inserted and sector 2.
Center point of the 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 1	R2	Input field for radius 2
Radius of circle 3	R3	Input field for radius 3
Feedrate	F	Input field for the interpolation feedrate

If it is not possible to determine the starting point from the previous blocks, use the "Starting point" screenform to enter the appropriate coordinates.

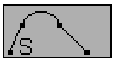
G2/G3

Use this softkey to define the direction of rotation of the two circles. You can choose 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

Center and end points can be acquired either in absolute dimensions, incremental dimensions or using polar coordinates. The current settings are displayed in the interactive screenform.



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

The screenform can be used if the following conditions are fulfilled:

Point	Given coordinates
Starting point	• Both coordinates in a Cartesian coordinate system • Starting point as a polar coordinate
Circle sector	• Both coordinates in the Cartesian coordinate system and the radius • Center point as a polar coordinate
End point	• Both coordinates in a Cartesian coordinate system • End point as a polar coordinate

Point	Given coordinates
Starting point	• Both coordinates in a Cartesian coordinate system • Starting point as a polar coordinate
Circle sector	• One coordinate in the Cartesian coordinate system and the radius • Angle A1 or A2
End point	• Both coordinates in a Cartesian coordinate system • End point as a polar coordinate

If it is not possible to determine the starting point from the previous blocks, the starting point must be set by the operator.

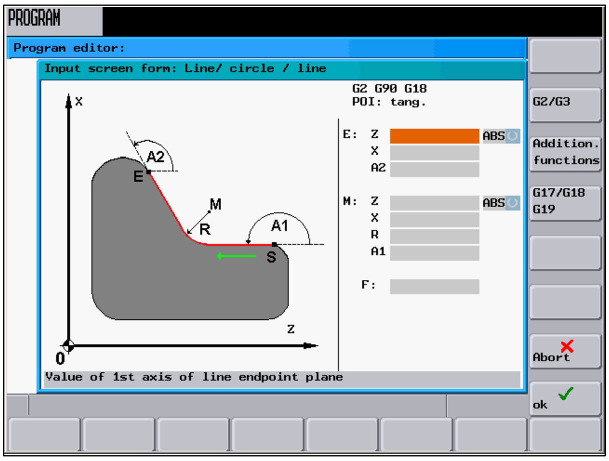


Fig. 6-30 Straight line – circle – straight line

Table 6-6 Input in the interactive screenform

End point of straight line 2	E	Enter the end point of the straight line.
Center point of the circle	M	1st and 2nd axes of the plane
Angle of straight line 1	A1	The angle is specified in the counterclockwise direction.
Angle of straight line 2	A2	The angle is specified in the counterclockwise direction.
Feedrate	F	Input field for the feedrate

End and center points can be specified either absolute, incremental or polar coordinates. The screenform generates one circle and two straight line blocks from the data you have entered.

G2/G3

Use this softkey to switch the direction of rotation from G2 to G3. G3 will appear on the display. Pressing this softkey again will switch back the display to G2. The display changes to G2.

6.4 Simulation

Functionality

By using broken–line graphics, the programmed tool path can be traced. No compensating movement is executed.

Note: Whether or not this function is implemented is decided by the machine manufacturer and performed via parameterization.

Operating sequence

You are in the AUTOMATIC mode and have selected a program for execution (cf. Section 5.1).

PROGRAM | Simulation

The start screen will appear.

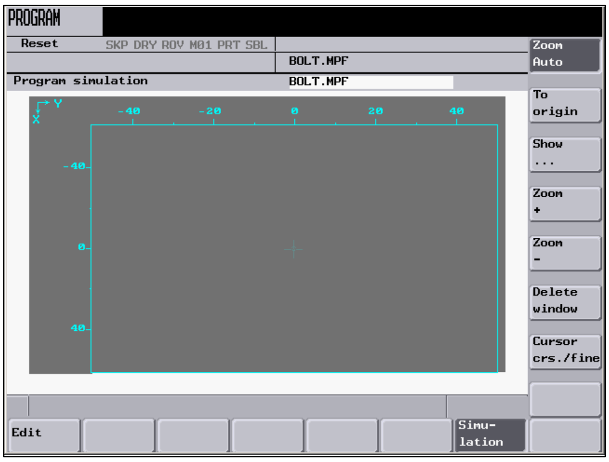
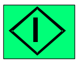


Fig. 6-31 The ”Simulation” start screen



Press **NC START** to start the simulation for the selected part program.

Softkeys

Zoom Auto — If you select this softkey, the recorded tool path is scaled automatically.

To origin — If you select this softkey, the default setting is used for the scaling.

Show ... — Various display options are offered here:

All G17 blocks — Displays the traversing motion of the specified plane.

All G18 blocks — Displays the traversing motion of the specified plane.

All G19 blocks — Displays the traversing motion of the specified plane.

Display All — Select this softkey to display the whole workpiece.

Zoom + — Use this softkey to enlarge the displayed section.

Zoom – — Use this softkey to reduce the displayed section.

Delete window — Use this softkey to delete the visible image.

Cursor crs./fine — Use this softkey to change the cursor increment.

6.5 Data transfer via the RS232 interface

Functionality

The RS232 interface of the control system can be used to output data (e.g. part programs) to an external data backup device or to read in data from there. The RS232 interface and your data backup device must be matched with each other.

File types

- **Main programs**
  - Part programs
  - Subroutines
- **Cycles**
  - Standard cycles
  - User cycles

Operating sequence

PROGRAM MANAGER — Programs

You have selected the **Program Manager** operating area and you are in the overview of the NC programs already created.

Read out — Use this softkey to saved files via the RS232 interface.

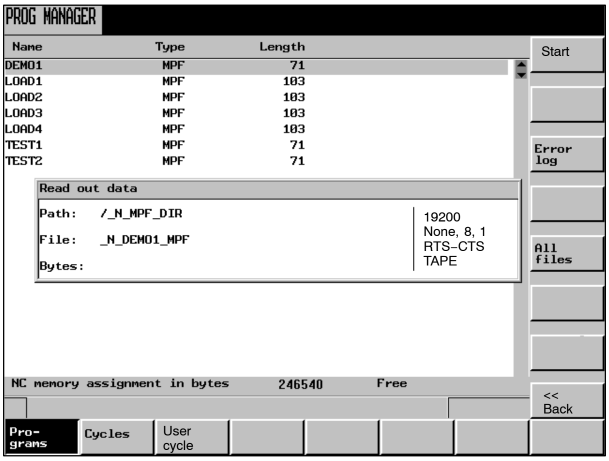


Fig. 6-32 Reading out a program

All files

Use this softkey to select all files.
Selecting this softkey selects all files from the part program directory and starts the data transfer.

Start

Use this softkey to start the output.
Selecting this softkey starts the output of one or several files from the part program directory. To cancel the transfer, use the **STOP** key.

Read in

Use this softkey to load part programs files via the RS232 interface.

Error log

Transfer log
This log contains all transmitted files with a status information:

- For files to be output:
  - the name of the file
  - an error acknowledgment
- For files to be input:
  - the name of the file and the path
  - an error acknowledgment

Transmission messages:

OK	Transmission completed successfully
ERR EOF	End–of–text character received, but archive file incomplete
Time Out	The time monitoring is reporting an interruption of the data transfer
User Abort	Data transfer aborted by the **Stop** softkey
Error Com	Error at the COM 1 port
NC / PLC Error	Error message from the NC
Error Data	Data error 1. Files read in with / without header or 2. Files transmitted without file names in the punched–tape format
Error File Name	The file name does not correspond to the name convention of the NC.

This sheet has been left empty for your notes

System

7

Functionality

The ”System” operating area provides all functions required for parameterizing and analyzing the NCK and the PLC.

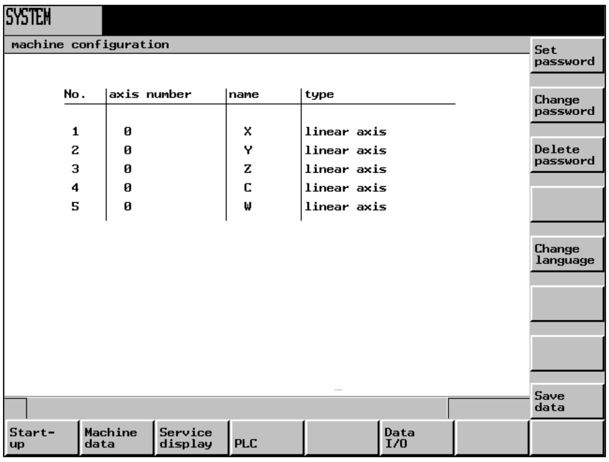


Fig. 7-1 The ”System” start screen

Depending on the function selected, the horizontal and the vertical softkey bars change. The menu tree shown below only shows the horizontal softkeys.

Start up	Machine data	Service display	PLC		Data I/O		
NC	General MD	Service Axes	STEP 7 connect		Data selection		
PLC	Axis MD	Service drives	PLC status		RS232 settings		
	Channel MD	Service profibus	Status list				
	Drive MD		PLC lprogram				
			Program list				
	Display MD						
	Servo trace	Servo trace					
		Version	Edit PLC alarm txt				

Fig. 7-2 The ”System” menu tree (only horizontal level)

Softkey

Set password

Setting the password

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

- System password
- Manufacturer password
- User password

Depending on the access levels (see also ”Technical Manual”), certain data can be changed.

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

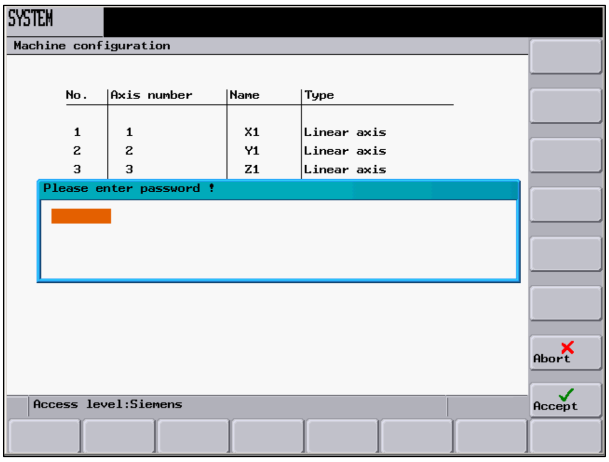


Fig. 7-3 Entering the password

After you have selected the **OK** softkey, the password is set.
Use **ABORT** to return without any action to the *System* main screen.

Change password

Changing the password

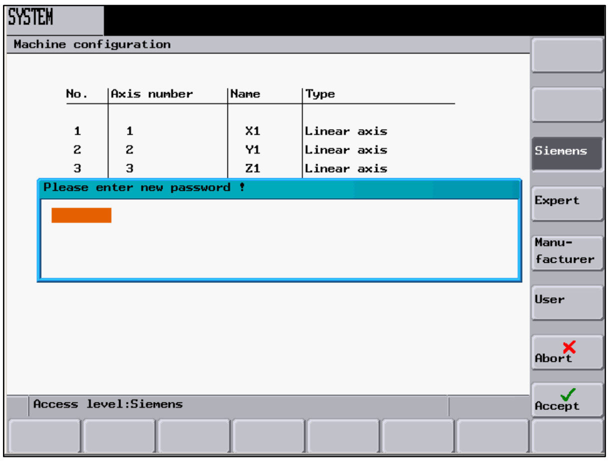


Fig. 7-4 Changing the password

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

Select the password level using the appropriate softkeys. Enter the new password and press **OK** to complete your input.

You will be prompted to enter the new password once more for confirmation.

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

Use **ABORT** to return without any action to the *Start–up* main screen.

Delete password

Resetting the access right

Change language

Change language

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

Save data

Saving data

This function will save the contents of the volatile memory into a nonvolatile memory area.
Prerequisite: There is no program currently executed.

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

Start up

Start–up

NC

Use this softkey to select the power–up mode of the NC.
Select the desired mode using the cursor.

- Normal power–up
  The system will be restarted.
- Power–up with default data
  Cold restart with the default values (will restore the default data as on delivery)
- Power–up with saved data
  Cold restart with the data saved last (see ”Data backup”)

PLC

The PLC can be started in the following modes:

- **Restart** Cold restart
- **Overall reset** Overall reset

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

OK

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

Use **RECALL** to return without any action to the System start screen.

Machine data

Machine data

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

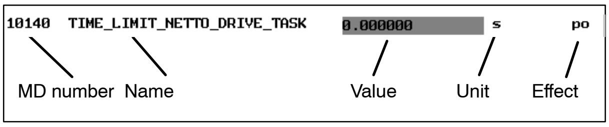


Fig. 7-5 Structure of a machine data line

Activation	so	immediately effective
	cf	with confirmation
	re	Reset
	po	Power on

Caution

Faulty parameterization may result in destruction of the machine.

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

General MD

General machine data

Open the *General machine data* window. Use the PageUp / PageDown keys to browse forward / backward.

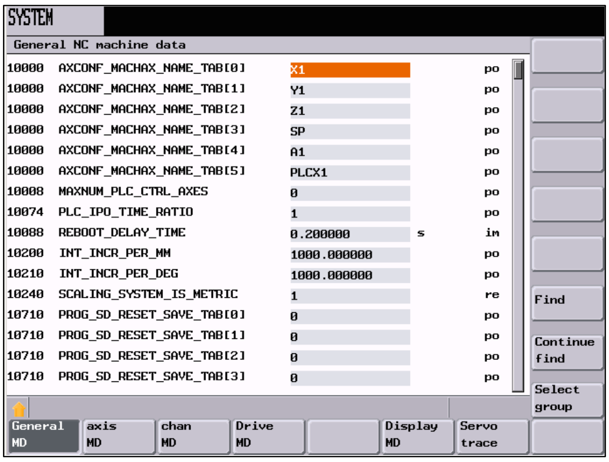


Fig. 7-6 The "Machine data" start screen

Axis MD

Axis-specific machine data

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

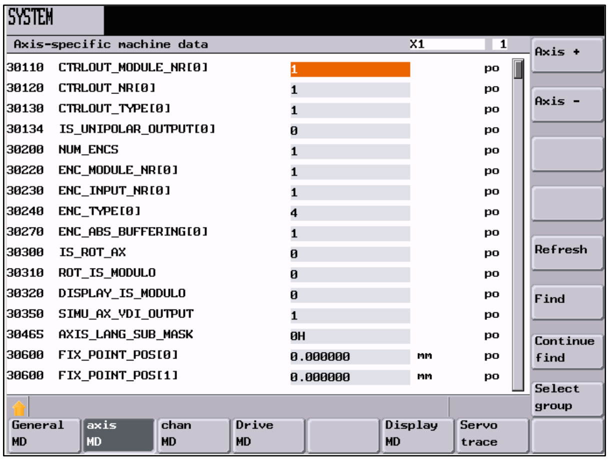


Fig. 7-7 Axis–specific machine data

The data of axis 1 are displayed.

Axis +

Axis–

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

Find

Find

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

The cursor will jump to the data searched.

Continue find

Use this softkey to continue searching for the next match.

Select group

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

Softkey **Expert**: Use this softkey to select all data groups of the Expert mode for display.

Softkey **Filter active**: Use this softkey to activate all data groups selected. After you have quit the window, you will only see the selected data on the machine data display.

Select all softkey: Use this softkey to select all data groups of the Expert mode for display.

Deselect all softkey: Selecting this softkey deselects all data groups.

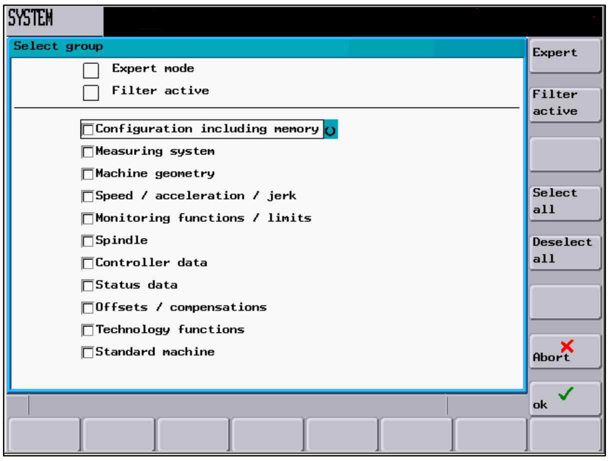


Fig. 7-8 Display filter

Channel MD

Channel–specific machine data

Open the *Axis–specific machine data* window. Use the PageUp / PageDown keys to browse forward / backward.

Drive MD

Drive machine data

Open the *Drive machine data* window. Use the PageUp / PageDown keys to browse forward / backward.

Display MD

Display machine data

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

Note for the reader

For a description of the machine data, please refer to the Manufacturer's Documentation:
"SINUMERIK 802D Instruction Manual"
"SINUMERIK 802D, Description of Functions".

Service display

Selecting this softkey displays the *Service axes* window.

Service Axes

This window displays information in respect of the axis.

Use the **Axis+** or **Axis-** softkeys to display the values for the next or previous axis.

Service drive

This window displays information in respect of the digital drive.

Service profibus

This window displays information in respect of the PROFIBUS settings.

Servo trace

To optimize the drives, an oscilloscope function is provided for graphical representation

- of the velocity setpoint
- of the contour violation
- of the following error
- of the actual position value
- of the position setpoint
- of exact stop coarse / fine

The start of tracing can be linked to various criteria allowing a synchronous tracing of internal control states. This setting must be made using the "**Select signal**" function.

To analyze the result, the following functions are provided:

- Changing and scaling of abscissa and ordinate;
- Measuring of a value using the horizontal or vertical marker;
- Measuring of abscissa and ordinate values as a difference between two marker positions;
- Storing of the result as a file in the part program directory. Thereafter, it is possible to export the file using WINPCIN and to process the data in MS Excel.

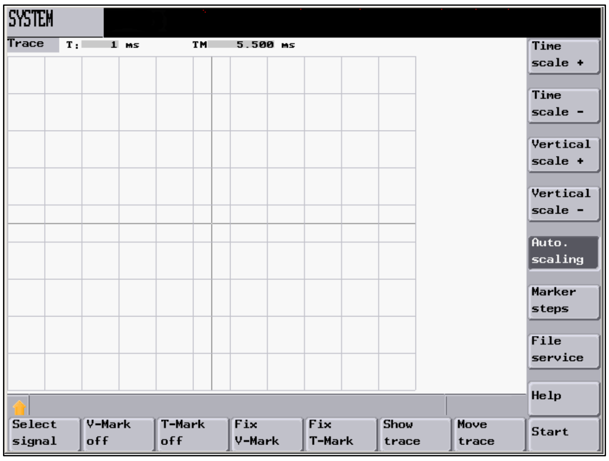


Fig. 7-9 The *Servo trace* start screen

The header of the diagram contains the current scaling of the abscissa and the difference value of the markers.

The diagram shown above can be moved within the visible screen area using the cursor keys.

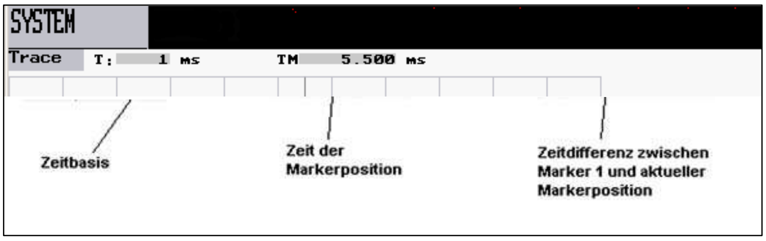


Fig. 7-10 Meaning of the fields

Select signal

Use this menu to parameterize the measuring channel.

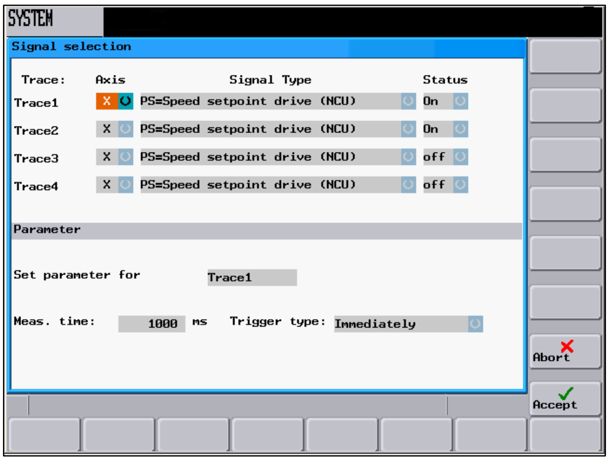


Fig. 7-11

- **Selecting the axis:** To select the axis, use the ”Axis” toggle field.
- **Signal type:** Following error
  Servo difference
  Contour violation
  Actual position value
  Velocity actual value
  Velocity setpoint
  Compensation value
  Parameter set
  Position setpoint at controller input
  Velocity setpoint at controller input
  Acceleration setpoint at controller input
  Velocity feedforward control value
  Exact stop fine signal
  exact stop coarse signal
- **Status:** On The tracing is carried out in this channel
  Off The channel is inactive.

The parameters for the measuring time and for the trigger type for channel 1 can be set in the lower screen half. The remaining channels will accept this setting.

- **Determining the measuring time:** The measuring time in ms is entered directly in the ”Measuring time” input field. It applies to all trace channels.

- **Selecting the trigger condition:** Position the cursor on the ”Trigger condition” field and select the relevant condition using the toggle key.
  - No trigger, i.e. the measurement starts directly after selecting the ”Start” softkey;
  - Positive edge;
  - Negative edge;
  - Exact stop fine reached;
  - Exact stop coarse reached

V–Mark OFF

T–Mark OFF

Use the **Marker on** / **Marker off** softkeys to hide / unhide the gridlines.

FIX V–Mark

FIX T–Mark

Use the markers to determine the differences in the horizontal or vertical direction. To this end, position the marker on the starting point and select either the ”**Fix H – Mark.**” or the ”**Fix T– Mark.**” softkey. The difference between the starting point and the current marker position is now displayed in the status bar. The softkey designations will change to ”**Free H – Mark.**” or ”**Free T – Mark.**”.

Show trace

This function opens another menu level offering softkeys for hiding / unhiding the diagrams. If a softkey is displayed on a black background, the diagrams are displayed for the selected trace channel.

Time scale +

Time scale –

Use this function to zoom in / zoom out the time basis.

Vertical scale +

vertical scale –

Use this function to increase / reduce the resolution (amplitude).

Marker steps

Use these softkeys to define the step sizes of the markers.

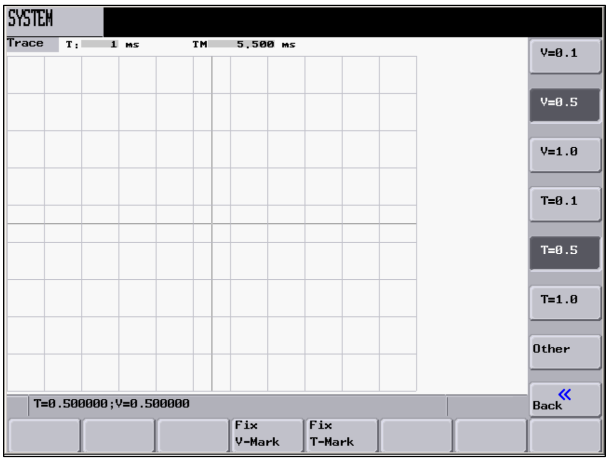


Fig. 7-12

The markers are moved using the cursor keys at a step size of one increment. larger step sizes can be set using the input fields. The value specifies how many grid units must be moved per <**SHIFT cursor movement.**If a marker reaches the margin of the diagram, the grid automatically appears in the horizontal or vertical direction.

File service

Use this softkey to save or load trace data.

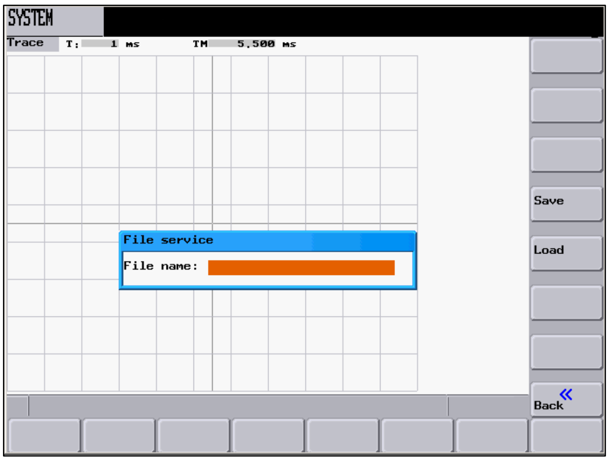


Fig. 7-13

Type the desired file name without extension in the ”File name” field.

Use the **Save** softkey to save the data with the specified name in the part program directory. Thereafter, the file can be exported, and the data can be processed in MS Excel.

Use the **Load** softkey to load the specified file and to display the data graphically.

Version

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

HMI details

The menu **HMI details** is intended for servicing and can only be accessed via the user password level. All programs provided by the operator unit are displayed with their version numbers. By reloading software components, the version numbers can be differ from each other.

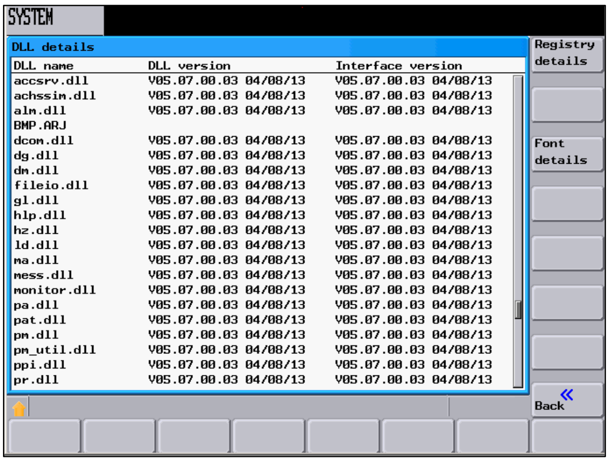


Fig. 7-14 The ”HMI version” menu area

registry details

This function displays the assignment of the hardkeys (function keys ”Machine”, ”Offset”, ”Program”, ...) for the programs to be started in the form of a list. For the meanings of the individual columns, please refer to the table below.

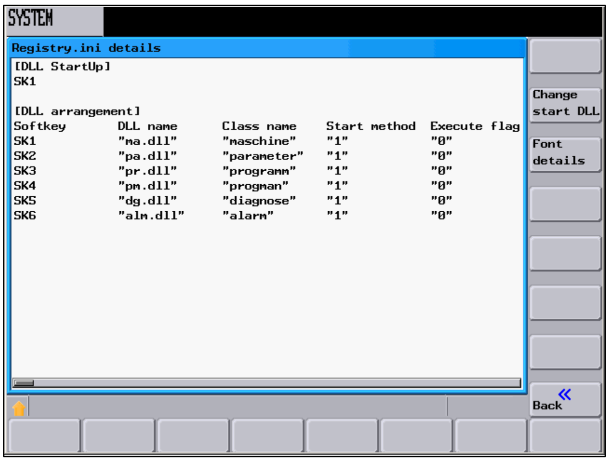


Fig. 7-15

Table 7-1 Meanings of the entries under [DLL arrangement]

Designation	Meaning
Softkey	SK1 to SK7 Hardkey assignment 1 to 7
DLL name	Name of the program to be executed
Class name	The identifier for receiving messages is defined in this column.
Start method	Number of the function executed after starting the program
Execute flag (kind of execution)	0 - The program is managed via the basic system. 1 - The basic system starts the program and transfers the control to the loaded program.
Text file name	Name of the text file (without extension)
Softkey text ID (SK ID)	Reserved
Password level	The execution of the program depends on the password level.
Class SK	Reserved
SK file	Reserved

Font details

This function displays the data of the loaded character sets in the form of a list.

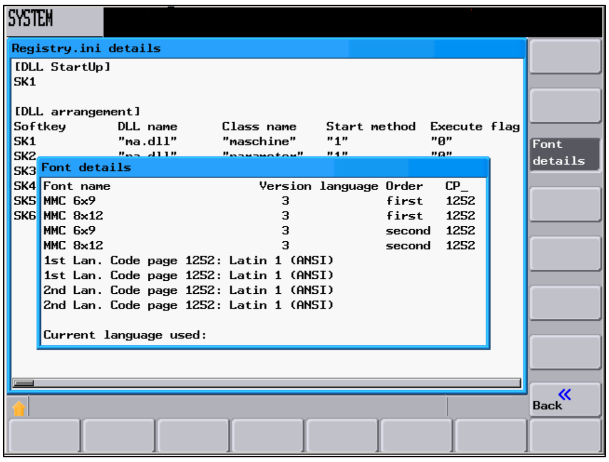


Fig. 7-16

Change Start DLL

Defining the start program

After the system has booted, the control system automatically starts the "Machine" operating area (SK 1). If a different starting behavior is desired, you can use this function to define a different starting behavior.

Type the number of the program ("Softkey" column) to be started after the system has booted here.

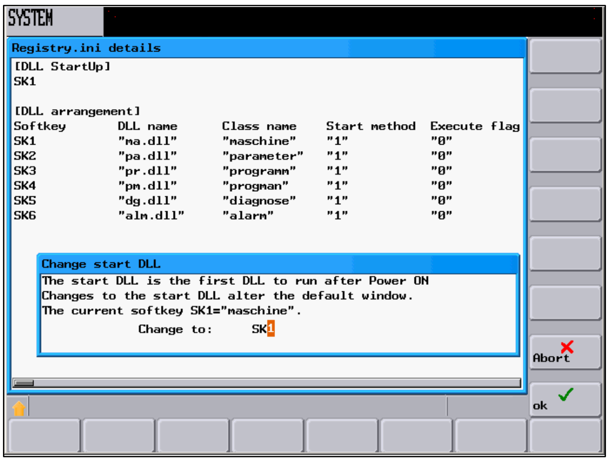


Fig. 7-17 Modifying the start-up DLL

PLC

This softkey offers further functions for diagnostics and start–up of the PLC.

STEP 7 connect

This softkey opens the configuration dialog for the interface parameters for the connection to STEP 7 (see also description of the Programming Tool, Section ”Communications”).

If the RS232 interface is already occupied by the data transfer, you can connect the control system to the Programming Tool only if the transmission is completed.

The RS232 interface is initialized with activation of the connection.

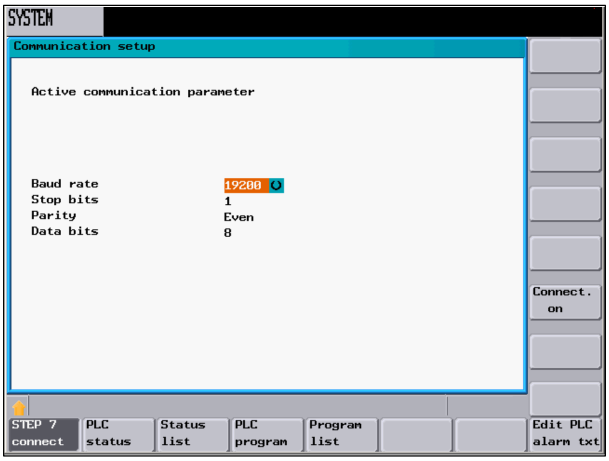


Fig. 7-18 Activating/deactivating RS232 for the Programming Tool

The baud rate is set using the toggle field. The following values are possible: 9600 / 19200 / 38400 / 57600 / 115200.

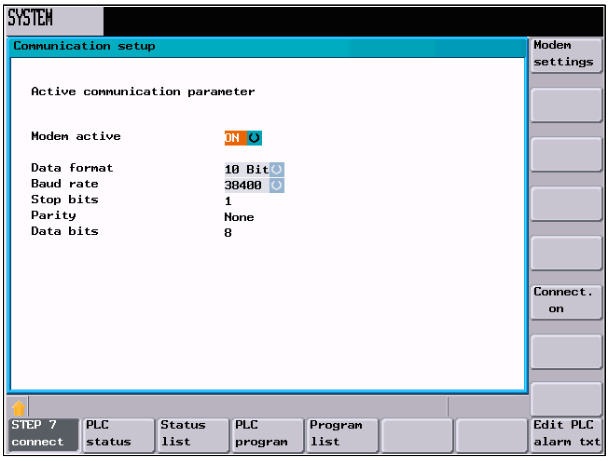


Fig. 7-19 Settings with the modem active

With the modem active (”ON”), you can additionally choose between the data formats 10 or 11–bit.

- Parity: ”None” with 10–bit format
  ”Even” with 11–bit format
- Stop bits: 1 (set by default; active with initialization of the control system)
- Data bits: 8 (set by default; active with initialization of the control system)

Connect on
Connect off

Use this softkey to activate the connection between the control system and the PC/PG. It is waited for the call of the Programming Tool. No modifications to the settings are possible in this state.
The softkey designations will change to **Connect off**.
You can cancel the transmission from the control system at any time by pressing **Connect off**. Now it is possible again to make changes in the settings.

The active or inactive state is kept even after Power On (except power–up with the default data). An active connection is displayed by a symbol in the status bar (cf. Table 1–2).

Press **Back** to quit the menu.

Modem settings

In this area, the modem settings are made.

Possible modem types are: Analog Modem
ISDN Box
Mobile Phone.

The types of both communication partners must match with each other.

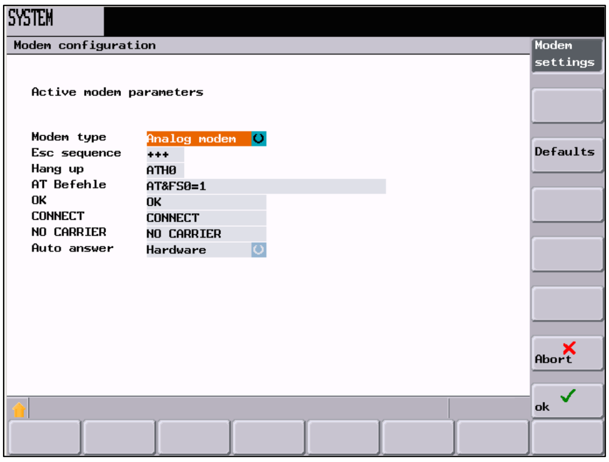


Fig. 7-20 Settings for an analog modem

When specifying several AT strings, AT must merely be written once; the remaining commands can simply be attached, e.g. AT&FS0=1E1X0&W. For the individual commands and their parameters, please refer to the manuals of the appropriate manufacturers. The default values of the control system are therefore only a real minimum and should be verified very exactly in any case before they are used for the first time. To be on the safe side, it is recommended to connect the devices first to a PC/PG and then to test and optimize the establishment of the connection.

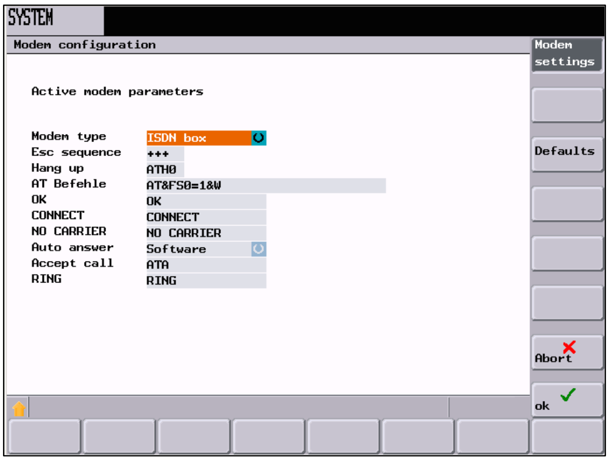


Fig. 7-21 Settings for an ISDN box

PLC status

Use this function to display and change the current states of the memory areas listed in Table 7-2.

It is possible to display 16 operands at the same time.

Table 7-2 Memory areas

Inputs	I	Input byte (IBx), input word (Iwx), input double–word (IDx)
Outputs	Q	Output byte (Qbx), output word (Qwx), output double–word (QDx)
Flags	M	Flag byte (Mx), flag word (Mw), flag double–word (MDx)
Timers	T	Timer (Tx)
Counter	C	Counter (Zx)
Data	V	Data byte (Vbx), data word (Vwx), data double–word (VDx)
Format	B H D	binary hexadecimal decimal The binary representation is not possible with double words. Counters and timers are represented decimally.

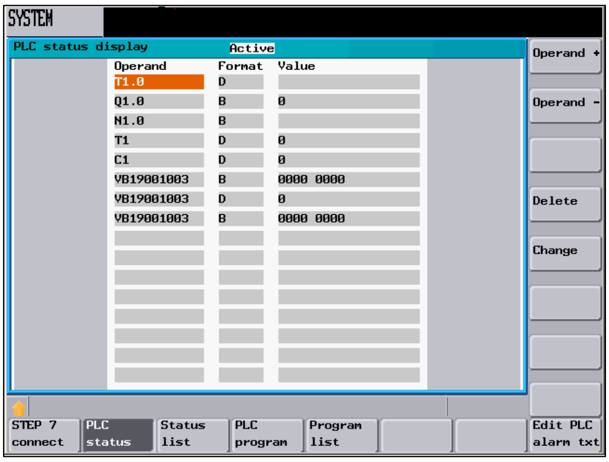


Fig. 7-22 PLC status display

Operand +
Each time if you select this softkey the operand address is incremented by 1.

Operand –
The operand address displays the value decremented by 1.

Delete
Use this softkey to delete all operands.

Change
This softkey will cancel the cyclic update of the values. Then you can change the values of the operands.

Status list
Use the **PLC status list** function to display and modify PLC signals.

There are 3 lists to choose from:

- Inputs (default setting) left list
- Flags (default setting) central list
- Outputs (default setting) right list
- Variable

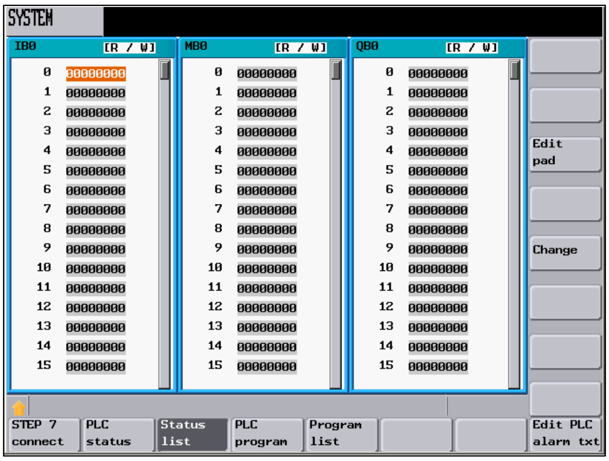


Fig. 7-23 The *PLC status liststart screen*

Use the **Edit pad function to change the settings.**

Change
Use this softkey to change the value of the selected variable. Select the **Accept softkey to confirm your changes.**

Edit pad
Use this softkey to assign the active column a new area. To this end, the interactive screenform offers four areas to choose from. For each column, a start address can be assigned which must be entered in the relevant input field. When you quit the interactive screenform, the control system will save your settings.

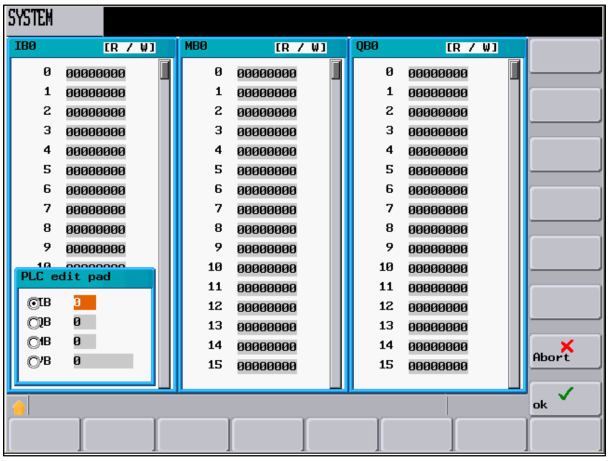


Fig. 7-24 The ”Data type” selection screen

Use the cursor keys and the PageUp / PageDownkeys to navigate in and between the columns.

PLC program

PLC diagnosis using a ladder diagram (see Section 7.1)

Program list

Using the PLC, you may select part programs and run them via the PLC. To this end, the PLC user program writes a program number to the PLC interface, which is then converted to a program name using a reference list. It is possible to manage max. 255 programs.

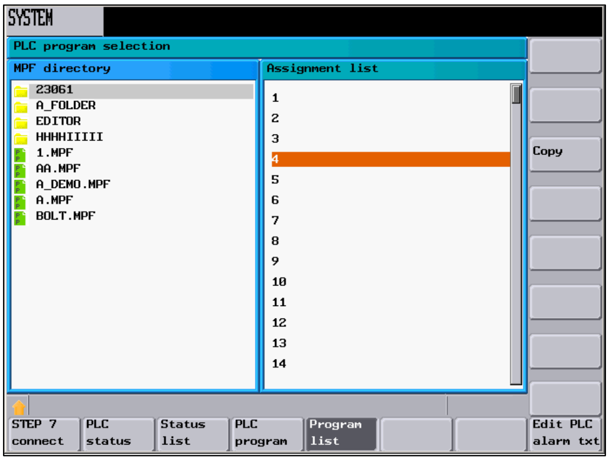


Fig. 7-25

This dialog displays all files of the CUS directory and their assignment in the reference list (PLCPROG.LST) in the form of a list. You can use the TAB key to switch between the two columns. The softkey functions **Copy**, **Insert** and **Delete** are displayed with reference to a specific context. If the cursor is positioned on the left–hand side, only the **Copy** function is available. On the right–hand side, the functions **Insert** and **Delete** are offered to modify the reference list.

Copy

... writes the selected file name to the clipboard

Insert

... pastes the file name at the current cursor position

Delete

... deletes the selected file name from the assignment list

Structure of the reference list (file PLCPROG.LST)

It is divided into 3 areas:

Number	Area	Protection level
1 ... 100	User area	User
101 ... 200	Machine manufacturer	Machine manufacturer
201 ... 255	Siemens	Siemens

The notation is carried out for each program by lines. Two columns are intended per line, which must be separated from each other by TAB, space or the "|" character. In the first column, the PLC reference number must be specified, and in the second column, the file name.

Example: 1 | shaft.mpf
2 | taper.mpf

Edit PLC alarm txt

This function can be used to insert or modify PLC user alarm texts. Select the desired alarm number using the cursor. At the same time, the text currently valid is displayed in the input line.

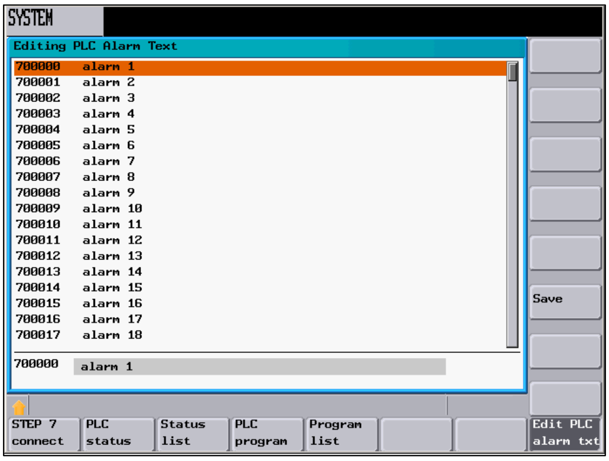


Fig. 7-26 Editing the PLC alarm text

Enter the new text in the input line. Press the **Input** key to complete your input and select **Save** to save it.

For the notation of the texts, please refer to the Start–Up Guide.

Data I/O

The window is divided into two columns. The left column is used to select the data group, and the right–hand column is used to select individual data for transfer. If the cursor is positioned in the left–hand column, the whole data group is output when **Read out** is selected. If it is positioned in the right–hand column, only the selected file is transferred. You can use the TAB key to switch between the two columns.

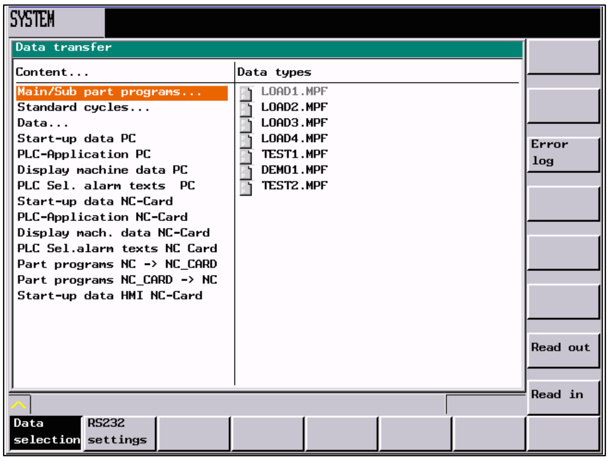


Fig. 7-27

In the **NC Card** selection area, the set interface parameters are ineffective. When reading in data from **NC Card**, first the desired area must be selected.

If when reading in one of the areas

- **PLC Sel.** or
- **Alarm texts PC** is selected when reading in
- **Start–up data PC, PLC–Application PC or Display machine data PC**
- **Alarm texts PC**

the settings of the column **special functions** are internally switched to **Binary format**.

Note

The menu item ”Part programs to NC –> NC_Card” or . “Part programs from NC_Card –> NC” will overwrite the existing files without confirmation warning.

Data selection

Select the data to be transferred. To start the transfer of the data to an external device, use the **Read out** softkey function.

To read in data from an external device, use the **Read in** function. For reading in, it is not necessary to select the data group, since the target is determined by the data flow.

RS232 settings

Use this function to display and change the interface parameters. The type of data to be transferred can be selected using the **Text Format** and **Binary Format** softkey functions.

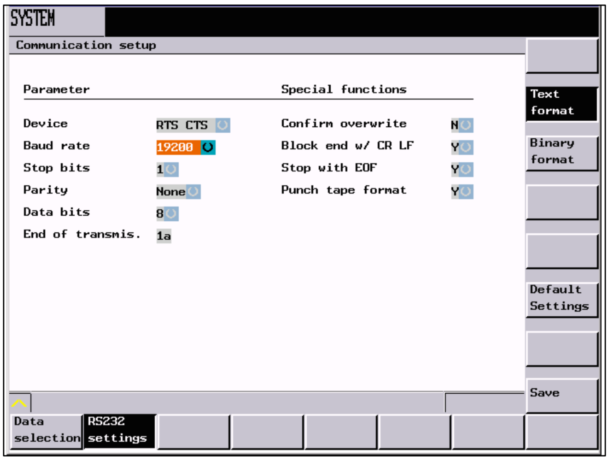


Fig. 7-28

Any changes in the settings come into effect immediately.

Pressing the **Save** softkey will save the selected settings even beyond switching off.

The **Default Settings** softkey will reset all settings to their default settings.

Interface parameters

Table 7-3 Interfac e parameters

Parameter	Description
Device type	**RTS/CTS** The signal RTS (Request to Send) controls the Send mode of the data transfer device. Active: Data are to be sent. Passive: The Send mode is only quitted after all data have been transmitted. The CTS signal indicates the readiness to transmit data as the acknowledgment signal for RTS.
Baud rate	... used to set the interface transmission rate. 300 Baud 600 Baud 1200 Baud 2400 Baud 4800 Baud 9600 Baud 19200 Baud 38400 Baud 57600 Baud 115200 Baud
Stop bits	Number of stop bits with asynchronous transmission Input: 1 stop bit (default setting) 2 stop bits

Table 7-3 Interfac e parameters, cont'd

Parameter	Description
Parity	Parity bits are used for error detection. These are added to the coded character to convert the number of digits set to ”1” into an odd or even number. Input: No parity (default setting) Parity even Parity odd
Data bits	Number of data bits with asynchronous transmission Input: 7 data bits 8 data bits (default setting)
Overwriting with con-firmation	Y: When reading in, it is checked whether the file already exists in the NC. **N:** The files are overwritten without confirmation warning.

7.1 PLC diagnosis represented as a ladder diagram

Functionality

A PLC user program consists to a large degree of logical operations to realize safety functions and to support process sequences. These logical operations include the linking of various contacts and relays. As a rule, the failure of a single contact or relay results in a failure of the whole system/installation.

To locate causes of faults/failures or of a program error, various diagnostic functions are offered in the ”System” operating area.

Note

It is not possible here to edit the program.

Operating sequence

PLC

Select the **PLC** softkey which is to be found in the ”System” operating area.

PLC program

The project stored in the permanent memory is opened.

7.1.1 Screen layout

The screen layout with its division into the main areas corresponds to the layout already described in Section 1.1. Any deviations and amendments pertaining to the PLC diagnosis are shown below.

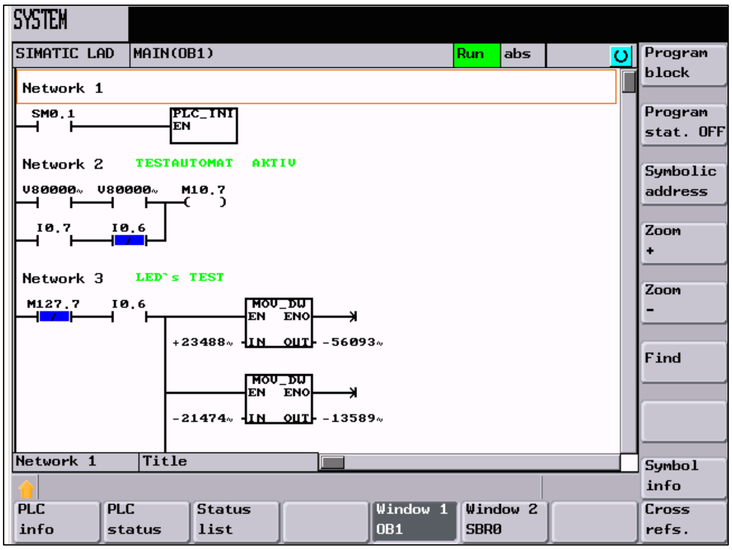


Fig. 7-29 Screen layout

Screen Control	Display	Meaning
①	**Applicationarea**	
②	**Supported PLC program language**	
③	**Name of the active program block** Representation: Symbolic name (absolute name)	
④	**Program status**	
	RUN	Program running
	STOP	Program stopped
	Status of the applicationarea	
	Sym	Symbolic representation
	abs	Absolute representation
⑤		**Display of the active keys**
⑥	**Focus** performs the tasks of the cursor	
⑦	**Tip line** contains notes for searching	

7.1.2 Operating options

In addition to the softkeys and the navigation keys, this area provides still further key combinations.

Key combinations

The cursor keys move the focus over the PLC user program. When reaching the window borders, it is scrolled automatically.

Table 7-4 Key combinations

Key combination	Action
NEXT WINDOW or CTRL ←	to the first line of the row
END or CTRL →	to the last line of the row
PAGE UP	up a screen
PAGE DOWN	down a screen
←	one field to the left
→	one field to the right

Table 7-4 Key combinations, cont'd

Key combination	Action
↑	up a field
↓	down a field
CTRL NEXT WINDOW or CTRL ↑	to the first field of the first network
CTRL END or CTRL ↓	to the last field of the first network
CTRL PAGE UP	opens the next program block in the same window
CTRL PAGE DOWN	opens the previous program block in the same window
SELECT	The function of the Select key depends on the position of the input focus. • Table line: Displays the complete text line • Network title: Displays the network comment • Command: Displays the complete operands
INPUT	If the input focus is positioned on a command, all operands including the comments are displayed.

Softkeys

PLC info

The ”PLC Info” menu (normally called ”About ... – transl.) displays the PLC model, the PLC system version, cycle time and PLC user program runtime.

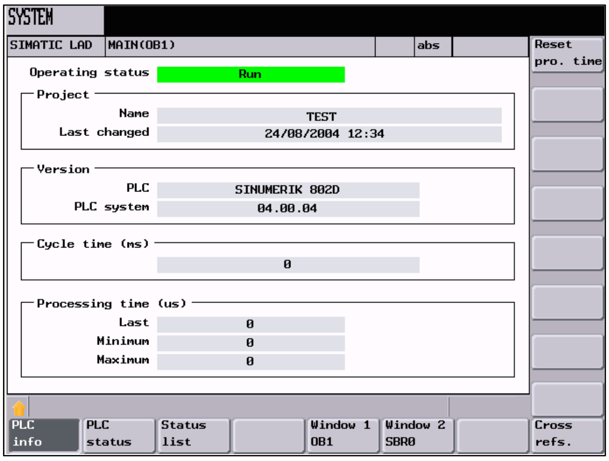


Fig. 7-30 PLCinfo

Reset pro. time

Use this softkey to refresh the data in the window.

PLC status

Use ”PLC status” for monitoring and changing during the program execution.

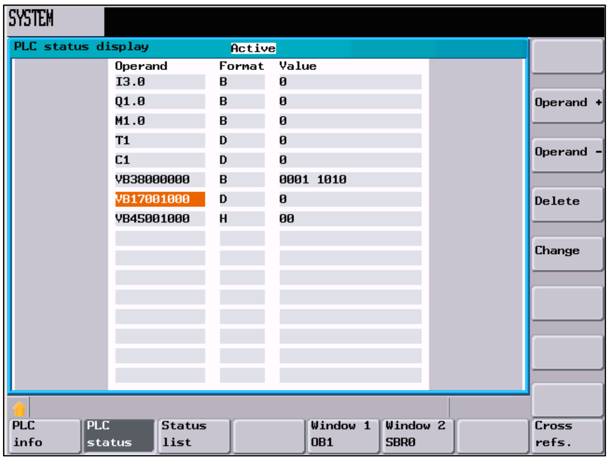


Fig. 7-31 PLC status display

Status list

Use the **PLC status list** function to display and modify PLC signals.

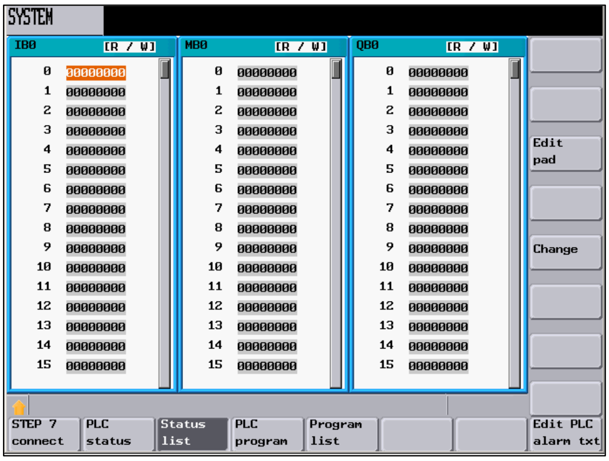


Fig. 7-32 Status list

Window 1 xxxx

Window 2 xxxx

This window displays all logical and graphical information of the PLC program running in the appropriate program block. The logic in the LAD (ladder diagram) is divided into clearly structured program parts and current paths, called networks. Generally, programs written in LADs represent the electrical current flow using various logical operations.

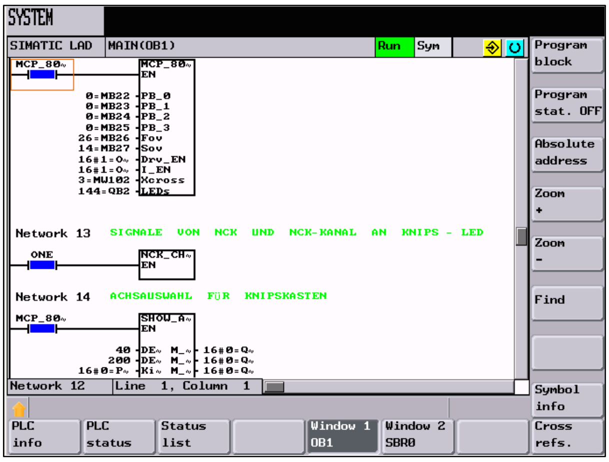


Fig. 7-33 Window 1

In this menu, you can switch between symbolic and absolute representation of the operand. Program sections can be displayed using various zoom factors; a search function is provided to find operands quickly.

Program block

This softkey can be used to display the list of the PLC program blocks. Use the **Cursor Up**/**Cursor Down** and **Page Up**/**Page Down** keys to select the PLC program block to be opened. The current program block is displayed in the Info line of the list box.

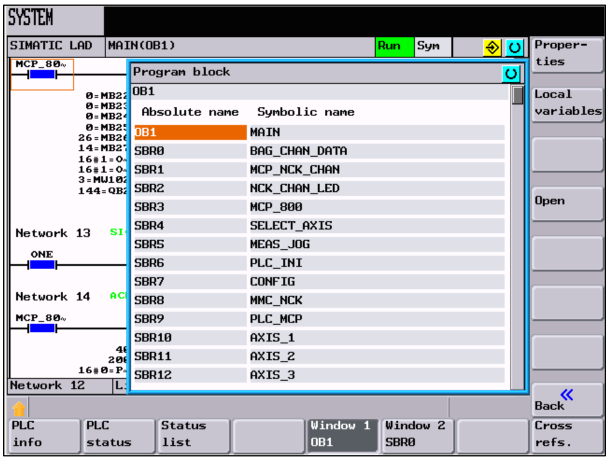


Fig. 7-34 PLC block selection

Proper–
ties

Selecting this softkey displays the description of the selected program block which was stored when the PLC project was created.

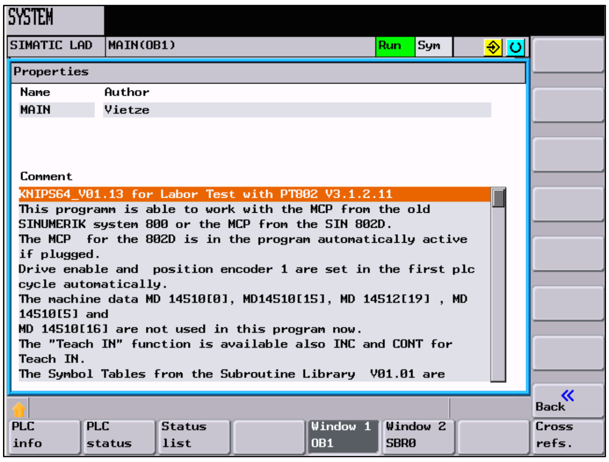


Fig. 7-35 Properties of the selected PLC program block

Local
variables

Pressing this softkey displays the table of local variables of the selected program block.

There are two types of program blocks.

- OB1 only temporary local variable
- SBRxx temporary local variable

A table of variables exists for each program block.

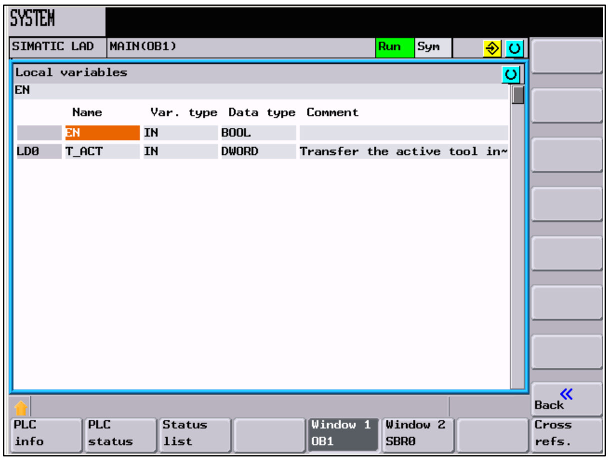


Fig. 7-36 Table of local variables for the selected program block

Texts which are longer than the column width are cut in all tables and the ”~” character is attached. For such a case, a higher–level text field exists in such tables in which the text of the current cursor position is displayed. If the text is cut with a ”~”, it is displayed in the same color as that of the cursor in the higher–level text field. With longer texts, it is possible to display the whole text by pressing the SELECT key.

Open

Selecting this softkey opens the selected program block; its name (absolute) is displayed on the ”Window 1/2” softkey.

Program stat. ON

Program stat. OFF

Use this softkey to activate/deactivate the display of the program status. It is possible here to observe the current network states beginning from the end of the PLC cycle. The states of all operands are displayed in the ”Program status” ladder diagram. This LAD acquires the values for the status display in several PLC cycles and then refreshes the status display.

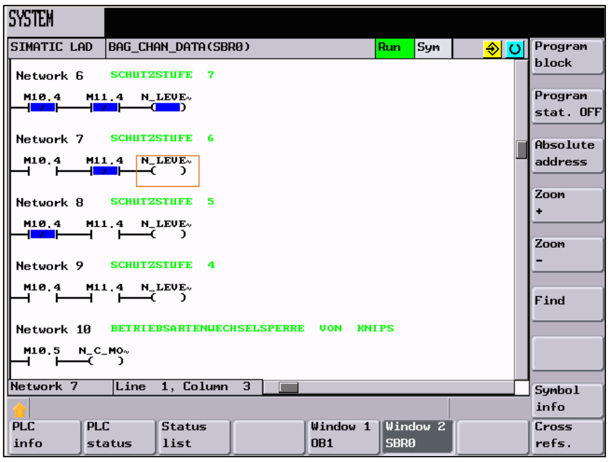


Fig. 7-37 ”Program status” ON – symbolic representation

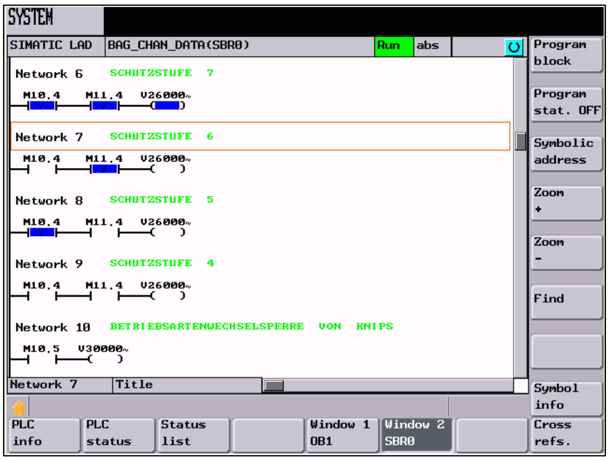


Fig. 7-38 "Program status" ON – absolute representation

Symbolic address / Absolute address

Use this softkey to switch between the absolute and symbolic representation of the operands. Depending on the selected type of representation, the operands are displayed either with absolute or symbolic identifiers.

If no symbol exists for a variable, this is automatically displayed absolutely.

Zoom + / Zoom –

The representation in the application area can be zoomed in or zoomed out step by step. The following zoom stages are provided:

20% (default), 60%, 100% and 300%

Find

can be used to search for operands in the symbolic or absolute representation

A dialog box is displayed from which various search criteria can be selected. Using the "**Absolute/Symbolic address**" softkey, you can search for a certain operand matching this criterion in both PLC windows. When searching, uppercase and lowercase letters are ignored.

Selection in the upper toggle field:

- Search for absolute and symbolic operands
- Go to network number
- Find SBR command

Further search criteria:

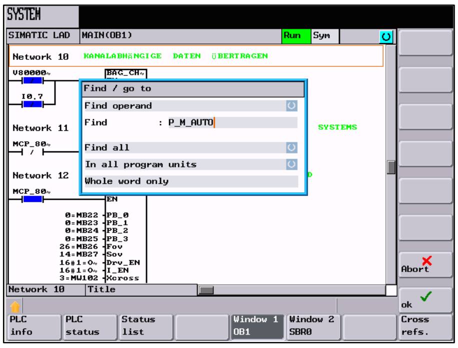
- Search direction down (from the current cursor position)
- Whole program block (from the beginning)
- In one program block
- Over all program blocks

You can search for the operands and constants as whole words (identifiers).

Depending on the display settings, you can search for symbolic or absolute operands.

Press the **OK** softkey to start the search. The found search element is highlighted by the focus. If nothing is found, an appropriate error message will appear in the notes line.

Use the **Abort** softkey to quit the dialog box; no search is carried out.



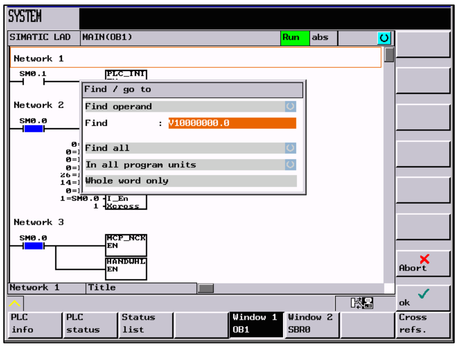


Fig. 7-39 Searching for symbolic operands

Searching for absolute operands

If the search object is found, use the **Continue search** softkey to continue the search.

Symbol info

Selecting this softkey displays all symbolic identifiers used in the highlighted network.

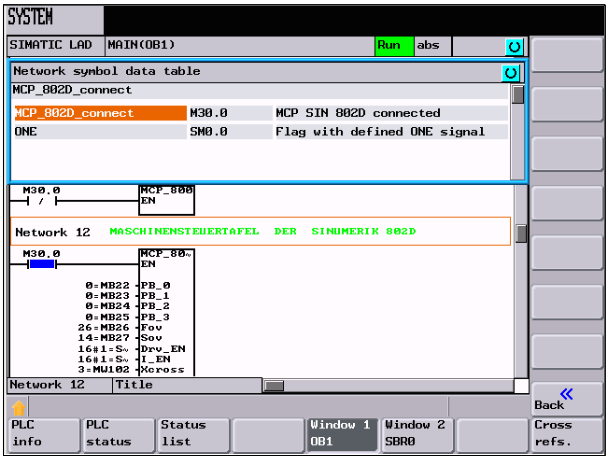
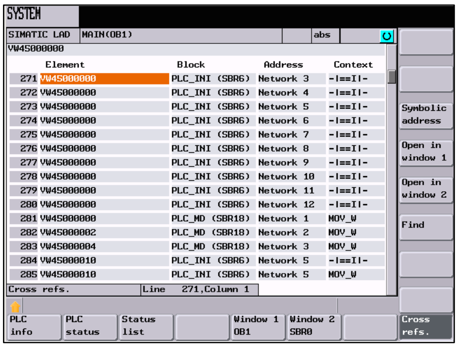


Fig. 7-40 Network symbolic

Cross refs.

Use this softkey to display the list of cross references. All operands used in the PLC project are displayed.

This list indicates in which networks an input, output, flag etc. is used.



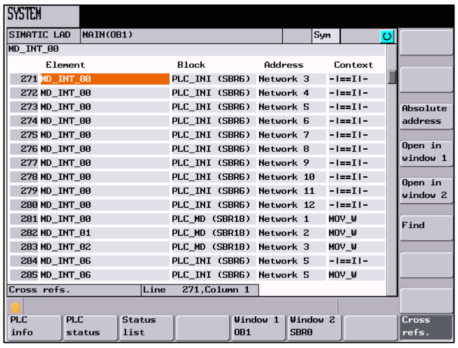


Fig. 7-41 The "Cross references" main menu (absolute)
(symbolic)

You can open the appropriate program segment directly in the 1/2 window using the **Open in Window 1/2** function.

Symbolic address

Absolute address

Depending on the active type of representation, the elements are displayed either with absolute or symbolic identifiers.

If no symbol exists for an identifier, the description is automatically absolute.

The type of representation of identifiers is displayed in the status bar. The absolute representation of identifiers is set by default.

Open in window 1

Open in window 2

The operand selected from the list of cross references is opened in the appropriate window.

Example:

You want to view the logic interrelation of the absolute operand M251.0 in network 1 in program block OB1.

After the operand has been selected from the cross–reference list and the **Open in Window 1** softkey has been actuated, the appropriate program section is displayed in window 1.

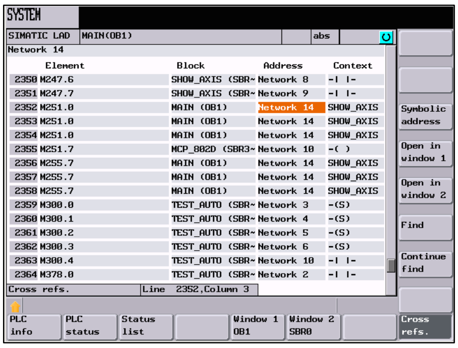
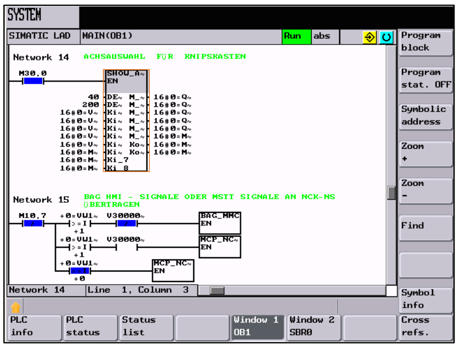


Fig. 7-42 Cursor "M251.0 in OB1 network 2)



M251.0 in OB1 network 2 in window 1

Find

... is used to search for operands in the list of cross references

You can search for the operands as whole words (identifiers). When searching, uppercase and lowercase letters are ignored.

Search options:

- Search for absolute and symbolic operands
- Go to line

Search criteria:

- Down (from the current cursor position)
- Whole program block (from the beginning)

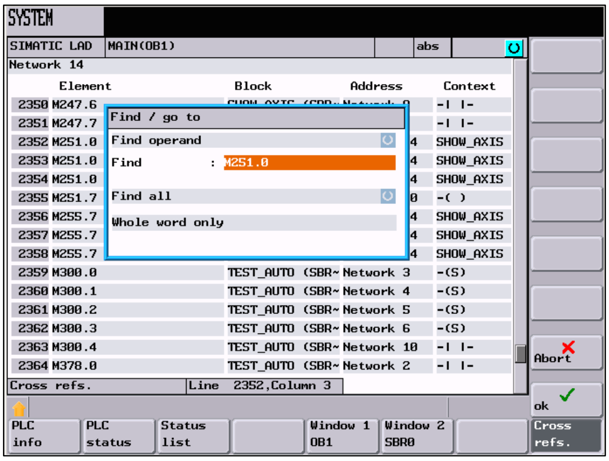


Fig. 7-43 Searching for operands in cross references

The text you are looking for is displayed in the notes line. If the text is not found, an appropriate error message is displayed which must be confirmed with OK.

If the search object is found, use the ”Continue search” softkey to continue the search.

7.2 Alarm display

Operating sequence

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

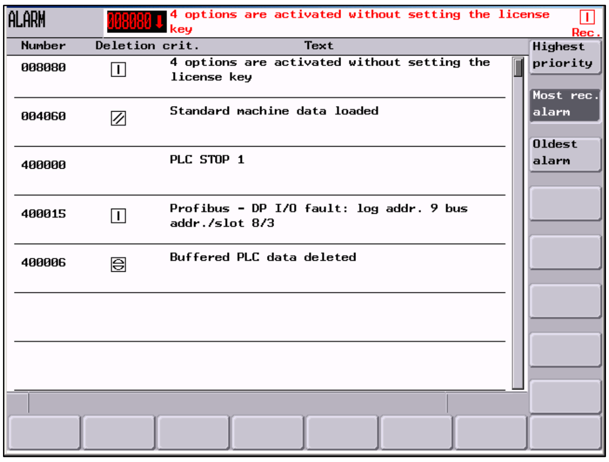


Fig. 7-44 Alarm window

Softkeys

Highest priority	Use this softkey to display all alarms sorted by their priority. The alarm with the highest priority stands at the beginning of the list.
Most rec. alarm	Use this softkey to display the alarms sorted by the time of their occurrence. The most recent alarm stands at the beginning of the list.
Oldest alarm	Use this softkey to display the alarms sorted by the time of their occurrence. The oldest alarm stands at the beginning of the list.

Programming

8.2 Positional data

8.2 Positional data

8.2.1 Plane selection: G17 to G19

Fu...lity

To assign, for example, **tool radius and tool length co**
is selected from the three axes X, Y and Z. In this plane
pensation.
For drill and cutter, the length compensation (length1) is

This sheet has been left empty for your notes

Programming 8

8.1 Fundamentals of NC programming

8.1.1 Program names

Each program has its own program name. When creating a program, the program name can be freely selected, observing the following rules:

- The first two characters must be letters;
- Only use letters, digits or underscore.
- Do not use delimiters (see Section ”Character set”)
- The decimal point must only be used for separation of the file extension.
- Do not use more than 16 characters.

Example: **FRAME52**

8.1.2 Program structure

Structure and contents

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

Each block represents a machining step.

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

The last block in the order of execution of the blocks contains a special word for the **program end: 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

8.1.3 Word structure and address

Functionality/structure

A word is a block element and mainly constitutes a control command. The word consists of

- **address character**: generally a letter
- and a **numerical value**: a sequence of digits which with certain addresses can be added by a sign put in front of the address, and a decimal point.

  A positive sign (+) can be omitted.

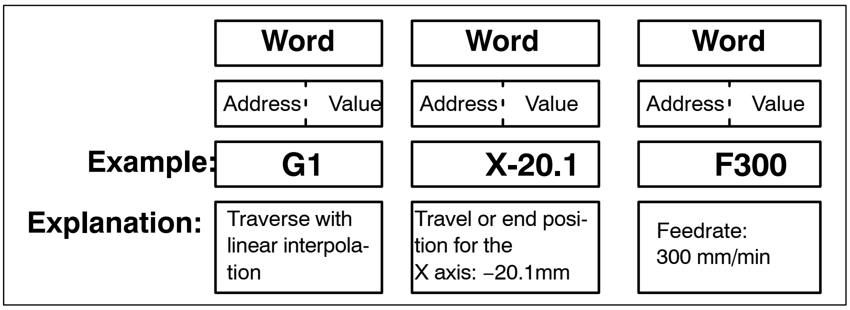


Fig. 8-1 Word structure (example)

Several address characters

A word can also contain several address letters. In this case, however, the numerical value must be assigned via the intermediate character ”=” .
Example: **CR=5.23**

Additionally, it is also possible to call G functions using a symbolic name (see also Section ”List of instructions”).
Example: **SCALE** ; Enable scaling factor

Extended address

For the addresses

R	Arithmetic parameter
H	H function
I, J, K	Interpolation parameter/intermediate point,

the addresses are extended by 1 to 4 digits to obtain more addresses. In this case, the value must be assigned using an equality sign ”=” (see also Section ”List of instructions”).
Example: **R10=6.234 H5=12.1 I1=32.67**

8.1.4 Block structure

Functionality

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

Generally, a block consists of several **words** and is always completed with the **end–of–block character** " L_F " (LineFeed). This character is automatically generated when pressing the line feed key or the **Input** key.

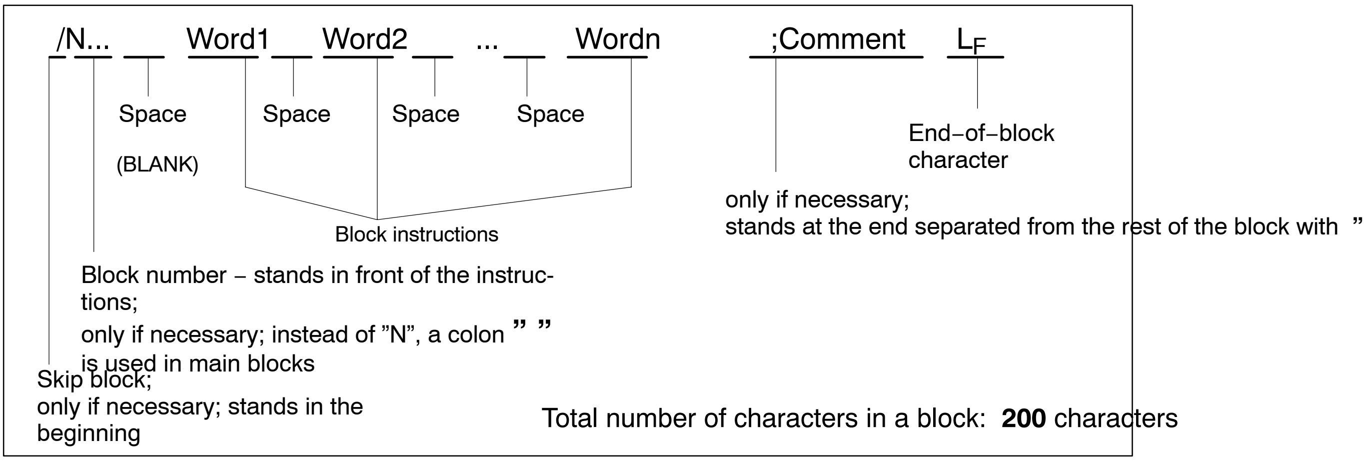


Fig. 8-2 Block structure diagram

Word order

If a block contains several instructions, the following order is recommended:
N... G... X... Y... Z... F... S... T... D... M... H...

Note regarding block numbers

First select the block numbers in steps of 5 or 10. Thus, you can later insert blocks and nevertheless observe the ascending order of block numbers.

Block skip

Blocks of a program, which are to be executed not with each program run, can be **marked** by a slash "\" in front of the block number. The block skip operation itself is activated either via **operation** (Program control: "SKP") or via the PLC (signal). It is also possible to skip a whole program section by skipping several blocks using the " / ".
If block skip is active during the program execution, all blocks marked with " / " are skipped. All instructions contained in the blocks concerned will not be considered. The program is continued with the next block without marking.

Comment, remark

The instructions in the blocks of a program can be explained using comments (remarks). A comment starts with the character " ; " and ends with block end.
Comments are displayed in the current block display, together with the remaining contents of the block.

Messages

Messages are programmed in a separate block. A message is displayed in a special field and remains active until a block with a new message is executed or until the end of the program is reached. Max. **65** characters of a text message can be displayed.
A message without message text will delete any previous message.
MSG ("THIS IS THE MESSAGE TEXT")

Programming example

```
N10                                   ; Company G&S, order no. 12A71
N20                                   ; Pump part 17, drawing no.: 123 677
N30                                   ; Program created by H. Adam, Dept. TV 4
N40 MSG("ROUGHING BLANK")
:50  G17 G54 G94 F470 S20 D2 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 G0 G90 X200
N120 M2                              ;End of program
```

8.1.5 Character set

The following characters are used for programming; they are interpreted in accordance with the relevant definitions.

Letters, digits

A, B, C, D, E, F, G, H, I, J, K, L, M, N,O, P, Q, R, S, T, U, V, W X, Y, Z
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
No distinction is made between lowercase and uppercase letters.

Printable special characters

(	Left round bracket	"	Inverted commas
)	Right round bracket	_	Underscore (belonging to a letter)
[	Left square bracket	.	Decimal point
]	Right square bracket	,	comma, delimiter
<	Less than	;	Start of a comment
>	Greater than	%	Reserved; do not use

:	Main block, completion of label
=	Assignment; part of equality
/	Division; block skip
*	Multiplication
+	Addition; positive sign
−	Subtraction; negative sign
&	reserved; do not use
’	Reserved; do not use
$	System−internal variable identifier
?	Reserved; do not use
!	Reserved; do not use

Non−printable special characters

L_F	End−of−block character
Blank	Delimiter between the words; blank
Tabulator	Reserved; do not use

8.1.6 Overview of the instructions

Valid as of software version 2.0.

Address	Meaning	Value Assignment	Information	Programming
D	Tool offset number	0 ... 9, only integer, no sign	Contains offset data for a certain tool T... ; D0–> offset values= 0, max. 9 D numbers per tool	D...
F	Feedrate	0.001 ... 99 999.999	Path velocity of a tool/workpiece; unit: mm/min or mm/revolution depending on G94 or G95	F...
F	Dwell time in block with G4	0.001 ... 99 999.999	Dwell time in seconds	G4 F... ;separate block
G	G function (preparatory function)	Only integer, specified values	The G functions are divided into G groups. Only one G function of a group can be programmed in a block. A G function can be either modal (until it is canceled by another function of the same group) or only effective for the block in which it is programmed non–modal. **G group:**	G... or symbolic name, e.g.: CIP
G0	Linear interpolation at rapid traverse rate		1: Motion commands	G0 X... Y... Z... ; Cartesian using polar coordinates: G0 AP=... RP=... or with additional axis: G0 AP=... RP=... Z... ; e.g.: with G17, Z axis
G1 *	Linear interpolation at feedrate		(type of interpolation)	G1 X... Y... Z... F... With polar coordinates: G1 AP=... RP=... F... or with additional axis: G1 AP=... RP=... Z... F... ; e.g.: with G17, Z axis
G2	Circular interpolation CW (in conjunction with a 3rd axis and TURN=... also helix interpolation –>see also TURN)		modally effective	G2 X... Y... I... J... F... ; Center and end points G2 X... Y... CR=... F... ; Radius and end point G2 AR=... I... J... F... ; Aperture angle and center point G2 AR=... X... Y... F... ; Aperture angle and end point in polar coordinates: G2 AP=... RP=... F... or with additional axis: G2 AP=... RP=... Z... F... ; e.g.: with G17, Z axis

G3	Circular interpolation CCW (in conjunction with a 3rd axis and TURN=... also helix interpolation –>see also TURN)		G3 .... ;otherwise, as with G2
CIP	Circular interpolation via intermediate point		CIP X... Y... Z... I1=... J1=... K1=... F...
CT	Circular interpolation; tangential transition		N10 ... N20 CT X... Y... F... ;Circle, tangential transition to the previous path segment
G33	Thread cutting, tapping with constant lead		S... M... ;spindle speed, direction G33 Z... K... ;Tapping **with** compensation chuck, e.g. in the Z axis
G331	Thread interpolation		N10 SPOS=... ;Position–controlled spindle N20 G331 Z... K... S... ;Tapping **without** compensation chuck, e.g. in the Z axis ;RH or LH thread is specified via the arithmetic sign of the lead (e.g. K+): + : as with M3 – : as with M4
G332	Thread interpolation – retraction		G332 Z... K... ;rigid tapping (**without** compen–sation chuck, e.g. along the Z axis, **retraction motion** ; Sign of the lead as with G331
G4	Dwell time	2: Special motions, non–modal	G4 F... ;separate block, F: Time in seconds or G4 S.... ;separate block, S: in spindle re-volutions
G63	Tapping with compensation chuck		G63 Z... F... S... M...
G74	Reference point approach		G74 X1=0 Y1=0 Z1=0 ;separate block (machine axis identifier!)
G75	Fixed–point approach		G75 X1=0 Y1=0 Z1=0 ;separate block (machine axis identifier!)
G147	SAR –approach along a straight line		G147 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G148	SAR –retraction along a straight line		G148 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
G247	SAR –approach along a quadrant		G247 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G248	SAR –retraction along a quadrant		G248 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
G347	SAR –approach along a semicircle		G347 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G348	SAR –retraction along a semicircle		G348 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...

TRANS	Programmable offset	3: Write memory	TRANS X... Y... Z... ;separate block
ROT	programmable rotation	non−modal	ROT RPL=... ;rotation in the current plane G17 ... G19, separate block
SCALE	Programmable scaling factor		SCALE X... Y... Z... ;scaling factor in the direction of the specified axis, separate block
MIRROR	programmable mirroring		MIRROR X0 ; coordinate axis whose direction is changed; separate block
ATRANS	Additive programmable offset		ATRANS X... Y... Z... ;separate block
AROT	additive programmable rotation		AROT RPL=... ; Add. rotation in the current plane G17 ... G19, separate block
ASCALE	Additive programmable scaling factor		ASCALE X... Y... Z... ; scaling factor in the direction of the specified axis, separate block
AMIRROR	additive programmable mirroring		AMIRROR X0 ; coordinate axis whose direction is changed; separate block
G25	Lower spindle speed limitation or lower working area limitation		G25 S... ;separate block G25 X... Y ... Z... ;separate block
G26	Upper spindle speed limitation or upper working area limitation		G26 S... ;separate block G26 X... Y ... Z... ;separate block
G110	Pole specification, relative to the last programmed set position		G110 X... Y... ; Pole specification, Cartesian, e.g.: With G17 G110 RP=... AP=... ;pole specification, polar separate block
G111	Pole specification, relative to the origin of the current workpiece coordi-nate system		G111 X... Y... ; Pole specification, Cartesian, e.g.: With G17 G111 RP=... AP=... ;pole specification, polar separate block
G112	Pole specification, relative to the POLElast valid		G112 X... Y... ; Pole specification, Cartesian, e.g.: With G17 G112 RP=... AP=... ;pole specification, polar separate block

G17 *	X/Y plane	6: Plane selection	G17 .... ;vertical axis on this
G18	Z/X plane	modally effective	Plane is tool length
G19	Y/Z plane		offset axis
G40 *	Tool radius compensation OFF	7: Tool radius compensation	
G41	Tool radius compensation left of the contour	modally effective	
G42	Tool radius compensation right of the contour		
G500 *	Settable work offset OFF	8: Settable work offset	
G54	1st settable work offset	modally effective	
G55	2nd settable work offset		
G56	3rd settable work offset		
G57	4th settable work offset		
G58	5th settable work offset		
G59	6th settable work offset		
G53	Non−modal skipping of the settable work offset	9: Skipping of the settable work offset non−modal	
G153	Non−modal skipping of the settable work offset including base frame		
G60 *	Exact stop	10: Approach behavior	
G64	Continuous−path control mode	modally effective	
G9	Non−modal exact stop	11: Non−modal exact stop non−modal	
G601 *	Exact stop window, fine, with G60, G9	12: Exact stop window	
G602	Exact stop window, coarse, with G60, G9	modally effective	
G70	Inch dimension input	13: Inch / metr.dimension input	
G71 *	Metric dimension data input	modally effective	
G700	Inch dimension data input; also for feedrate F		
G710	Metric dimension data input; also for feedrate F		

G90 *	Absolute dimension data input	14: Absolute / incremental dimension modally effective	
G91	Incremental dimension data input		
G94 *	Feed F in mm/min	15: Feedrate / spindle modally effective	
G95	Feedrate F in mm/spindle revolutions		
CFC *	Feedrate with circle ON	16: Feedrate override modally effective	
CFTCP	Feedrate override OFF		
G450 *	Transition circle	18: Behavior at corners when working with tool radius compensation modally effective	
G451	Point of intersection		
BRISK *	Jerking path acceleration	21: Acceleration profile modally effective	
SOFT	Jerk–limited path acceleration		
FFWOF *	Feedforward control OFF	24: Feedforward control modally effective	
FFWON	Feedforward control ON		
WALIMON *	Working area limitation ON	28: Working area limitation modally effective	; applies to all axes activated via setting data; values set via G25, G26
WALIMOF	Working area limitation OFF		
G340 *	Approach and retraction in the space (SAR = smooth approach and retraction)	44: Path distribution with smooth approach and retraction modally effective	
G341	Approach and retraction in the plane (SAR)		
G290 *	SIEMENS mode	47: External NC languages modally effective	
G291	External mode		
The functions marked with an asterisk (*) act when starting the program (in the default condition of the control system, unless otherwise programmed and if the machine manufacturer has preserved the default settings for the milling technology).			

Address	Meaning	Value Assignment	Information	Programming
H H0= through H9999=	H function	± 0.0000001 ... 9999 9999 (8 decimals) or with specification of an exponent: ± (10^{-300} ... 10^{+300})	Value transfer to the PLC; meaning defined by the machine manufacturer	H0=... H9999=... e. g.: H7=23.456
I	Interpolation parameters	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	Belongs to the X axis; meaning dependent on G2,G3 –> circle center or G33, G331, G332 –> thread lead	See G2, G3, G33, G331 and G332
J	Interpolation parameters	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	Belongs to the Y axis; otherwise, as with I	See G2, G3, G33, G331 and G332
K	Interpolation parameters	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	Belongs to the Z axis; otherwise, as with I	See G2, G3, G33, G331 and G332
I1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the X axis; specification for circular interpolation with CIP	See CIP
J1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the Y axis; specification for circular interpolation with CIP	See CIP
K1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the Z axis; specification for circular interpolation with CIP	See CIP
L	Subroutine; name and call	7 decimals; integer only, no sign	It is also possible to use L1 ...L9999999, Instead of a free name; thus, the subroutine will be called in a separate block. Please observe: L0001 is not always equal to L1. The name ”LL6” is reserved for the tool change subroutine.	L781 ;separate block
M	Miscellaneous function	0 ... 99 integer only, no sign	For example, for initiating switching actions, such as ”Coolant ON”; max. 5 M functions per block	M...
M0	Programmed stop		The machining is stopped at the end of a block containing M0; to continue, press NC START.	
M1	Optional stop		As with M0, but the stop is only performed if a special signal (Program control: ”M01”) is present.	
M2	End of program		Can be found in the last block of the processing sequence	
M30	–		Reserved; do not use	

Address	Meaning	Value Assignment	Information	Programming
M17	–		Reserved; do not use	
M3	Spindle CW rotation			
M4	Spindle CCW rotation			
M5	Spindle stop			
M6	Tool change		Only if activated with M6 via the machine control panel; otherwise, change directly using the T command	
M40	Automatic gear stage switching			
M41 to M45	Gear stages 1 to 5			
M70, M19	–		Reserved; do not use	
M...	Remaining M functions		Functionality is not defined by the control system and can therefore be used freely by the machine manufacturer	
N	Block number of an auxiliary block	0 ... 9999 9999 integer only, no sign	Can be used to identify blocks with a number; is written in the beginning of a block	N20 ...
:	Block number of a main block	0 ... 9999 9999 integer only, no sign	Special block identification, used instead of N... ; such a block should contain all instructions for a complete subsequent machining step.	**:**20 ...
P	Number of subroutine passes	1 ... 9999 integer only, no sign	Is used if the subroutine is run several times and is contained in the same block as the call	N10 L781 P... ; separate block N10 L871 P3 ; passed three times
R0 through R299	Arithmetic parameters	± 0.0000001 ... 9999 9999 (8 decimals) or with specification of an exponent: ± (10^{-300} ... 10^{+300})		R1=7.9431 R2=4 With specification of an exponent: R1=–1.9876EX9 ; R1=–1 987 600 000
Arithmetic functions			In addition to the 4 basic arithmetic functions using the operands + – * /, there are the following arithmetic functions:	
SIN( )	Sine	Degrees		R1=SIN(17.35)
COS( )	Cosine	Degrees		R2=COS(R3)
TAN( )	Tangent	Degrees		R4=TAN(R5)

Address	Meaning	Value Assignment	Information	Programming
ASIN( )	Arc sine			R10=ASIN(0.35) ; R10: 20.487 degrees
ACOS( )	Arc cosine			R20=ACOS(R2) ; R20: ... degrees
ATAN2(,)	Arc tangent2		The angle of the sum vector is calculated from 2 vectors standing vertically one on another. The 2nd vector specified is always used for angle reference. Result in the range: −180 to +180 degrees	R40=ATAN2(30.5,80.1) ; R40: 20.8455 degrees
SQRT( )	Square root			R6=SQRT(R7)
POT( )	Square			R12=POT(R13)
ABS( )	Amount			R8=ABS(R9)
TRUNC( )	Integer portion			R10=TRUNC(R11)
LN( )	Natural logarithm			R12=LN(R9)
EXP( )	Exponential function			R13=EXP(R1)
RET	End of subroutine		Used instead of M2 – to maintain the continuous–path control mode	RET ;separate block
S	Spindle speed	0.001 ... 99 999.999	Unit of measurement of the spindle r.p.m.	S...
S	Dwell time in block with G4	0.001 ... 99 999.999	Dwell time in spindle revolutions	G4 S... ;separate block
T	Tool number	1 ... 32 000 integer only, no sign	The tool change can be performed either directly using the T command or only with M6. This can be set in the machine data.	T...
X	Axis	±0.001 ... 99 999.999	G command	X...
Y	Axis	±0.001 ... 99 999.999	G command	Y...
Z	Axis	±0.001 ... 99 999.999	G command	Z...
AC	Absolute coordinate	–	The dimension can be specified for the end or center point of a certain axis, irrespective of G91.	N10 G91 X10 Z=AC(20) ; X – incremental dimension, Z – absolute
ACC[*axis*]	Percentage path acceleration override	1 ... 200, integer	Acceleration override for an axis or spindle; specified as a percentage	N10 ACC[X]=80 ;for the X axis: 80% N20 ACC[S]=50 ;for the spindle: 50%
ACP	Absolute coordinate; approach position in the positive direction (for rotary axis, spindle)	–	It is also possible to specify the dimensions for the end point of a rotary axis with ACP(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=ACP(45.3) ;Approach absolute position of the A axis in the positive direction N20 SPOS=ACP(33.1) ;Position spindle

Address	Meaning	Value Assignment	Information	Programming
ACN	Absolute coordinate; approach position in the negative direction (for rotary axis, spindle)	–	It is also possible to specify the dimensions for the end point of a rotary axis with ACN(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=ACN(45.3) ;Approach absolute position of the A axis in the negative direction N20 SPOS=ACP(33.1) ;Position spindle
ANG	Angle for the specification of a straight line for the contour definition	±0.00001 ... 359.99999	Specified in degrees; one possibility of specifying a straight line when using G0 or G1 if only one end–point coordinate of the plane is known or if the complete end point is known with contour ranging over several blocks	N10 G1 G17 X... Y.... N11 X... **ANG=...** or contour over several blocks: N10 G1 G17 X... Y.... N11 **ANG=...** N12 X... Y... **ANG=...**
AP	Polar angle	0 ... ±359.99999	Specified in degrees, traversing in polar coordinates, definition of the pole; in addition: RP – polar radius	see G0, G1, G2, G3 G110, G111, G112
AR	Aperture angle for circular interpolation	0.00001 ... 359.99999	Specified in degrees; one possibility of defining the circle when using G2/G3	See G2, G3
CALL	Indirect cycle call	–	Special form of the cycle call; no parameter transfer; the name of the cycle is stored in a variable; only intended for cycle–internal use	N10 CALL VARNAME ; variable name
CHF	Chamfer; general use	0.001 ... 99 999.999	Inserts a chamfer of the specified **chamfer length** between two contour blocks	N10 X... Y.... **CHF=...** N11 X... Y...
CHR	Chamfer; in the contour definition	0.001 ... 99 999.999	Inserts a chamfer of the specified **leg length** between two contour blocks	N10 X... Y.... **CHR=...** N11 X... Y...
CR	Radius for circular interpolation	0.010 ... 99 999.999 Negative sign – for selecting the circle: greater than semicircle	One possibility of defining a circle when using G2/G3	See G2, G3
CYCLE... HOLES... POCKET.. SLOT...	Machining cycle	Only specified values	The call of the machining cycles requires a separate block; the appropriate transfer parameters must be loaded with values. Special cycle calls are also possible with an additional MCALL or CALL.	
CYCLE81	Drilling, centering			N5 RTP=110 RFP=100 .... ;assign values N10 CYCLE81(RTP, RFP, ...) ;separate block
CYCLE82	Drilling, counterboring			N5 RTP=110 RFP=100 .... ;assign values N10 CYCLE82(RTP, RFP, ...) ;separate block
CYCLE83	Deep–hole drilling			N10 CYCLE83(110, 100, ...) ;or transfer values directly; separate block

Address	Meaning	Value Assignment	Information	Programming
CYCLE84	Rigid tapping			N10 CYCLE84(...) ;separate block
CYCLE840	Tapping **with** compensation chuck			N10 CYCLE840(...) ;separate block
CYCLE85	Reaming			N10 CYCLE85(...) ;separate block
CYCLE86	Boring			N10 CYCLE86(...) ;separate block
CYCLE87	Boring 3			N10 CYCLE87(...) ;separate block
CYCLE88	Boring with stop			N10 CYCLE88(...) ;separate block
CYCLE89	Boring 5			N10 CYCLE89(...) ;separate block
CYCLE90	Thread milling			N10 CYCLE90(...) ;separate block
HOLES1	Row of holes			N10 HOLES1(...) ;separate block
HOLES2	Circle of holes			N10 HOLES2(...) ;separate block
SLOT1	Milling a slot			N10 SLOT1(...) ;separate block
SLOT2	Milling a circumferential slot			N10 SLOT2(...) ;separate block
POCKET3	Square pocket			N10 POCKET3(...) separate block
POCKET4	Circular pocket			N10 POCKET4(...) separate block
CYCLE71	Face milling			N10 CYCLE71(...) separate block
CYCLE72	Contour milling			N10 CYCLE72(...) separate block
LONG-HOLE	Elongated hole			N10 LONGHOLE(...) separate block
DC	Absolute coordinate; approach position directly (for rotary axis, spindle)	–	It is also possible to specify the dimensions for the end point of a rotary axis with DC(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=DC(45.3) ;Approach absolute position of the A axis directly N20 SPOS=DC(33.1) ;Position spindle
DEF	Definition instruction		Defining a local user variable of the type BOOL, CHAR, INT, REAL, directly at the beginning of the program	DEF INT VARI1=24, VARI2 ; 2 variables of the type INT ; the name is defined by the user
DISCL	Approach / retraction distance of the infeed movement to the machining plane (SAR)	–	Safety clearance for switching the velocity during the infeed motion; Observe: G340, G341	See with G147, G148 , G247, G248 , G347, G348

Address	Meaning	Value Assignment	Information	Programming
DISR	Approach/retraction distance or approach/retraction radius (SAR)	–	G147/G148: Distance of the cutter edge from the starting or end point of the contour G247, G347/G248, G348: Radius of the tool center point path	See with G147, G148 , G247, G248 , G347, G348
FAD	Infeed speed (SAR)	–	The velocity acts after reaching the safety clearance when infeeding; Observe: G340, G341	See with G147, G148 , G247, G248 , G347, G348
FXS [*axis*]	Travel to fixed stop	=1: Selection =0: Deselection	Axis: Use the machine identifier	N20 G1 X10 Z25 FXS[Z1]=1 FXST[Z1]=12.3 FXSW[Z1]=2 F...
FXST [*axis*]	Clamping torque, travel to fixed stop	> 0.0 ... 100.0	in %, max. 100% from the max. torque of the drive, *axis:* Use the machine identifier	N30 FXST[Z1]=12.3
FXSW [*axis*]	Monitoring window, travel to fixed stop	> 0.0	Unit of measurement mm or degrees, axis–specific, *axis:* Use the machine identifier	N40 FXSW[Z1]=2.4
GOTOB	GoBack instruction	–	A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the program start.	N10 LABEL1: ... ... N100 GOTOB LABEL1
GOTOF	GoForward instruction	–	A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the end of the program.	N10 GOTOF LABEL2 ... N130 LABEL2: ...
IC	Coordinate specified using incremental dimensions	–	The dimension can be specified for the end or center point of a certain axis irrespective of G90.	N10 G90 X10 Z=IC(20) ;Z – incremental dimension, X – absolute dimension
IF	Jump condition	–	If the jump condition is fulfilled, the jump to the next block marked with the *label:* is executed; otherwise, the next instruction/block is executed. Several IF instructions in a block are possible. **Comparison operands:** = = equal to, <> not equal > greater than, < less than >= greater than or equal to <= less than or equal to	N10 IF R1>5 GOTOF LABEL3 ... N80 LABEL3: ...
MEAS	Measuring **with** deletion of the distance to go	+1 –1	=+1: Measuring input 1, rising edge =–1: Measuring input1, falling edge	N10 **MEAS=–1** G1 X... Y... Z... F...
MEAW	Measuring **without** deletion of the distance to go	+1 –1	=+1: Measuring input 1, rising edge =–1: Measuring input1, falling edge	N10 **MEAW=–1** G1 X... Y... Z... F...

Address	Meaning	Value Assignment	Information	Programming
$A_DBB[n] $A_DBW[n] $A_DBD[n] $A_DBR[n]	Data byte Data word Data double word Real data		Reading and writing PLC variables	N10 $A_DBR[5]=16.3 ; write real variables ; with offset position 5 ; (position, type and meaning are agreed between NC and PLC)
$A_MONI-FACT	Factor for tool life monitoring	> 0.0	Initialization value: 1.0	N10 $A_MONIFACT=5.0 ; tool life elapsed 5 times faster
$AA_FXS [*axis*]	Status, travel to fixed stop	–	Values: 0 ... 5 *axis*: machine axis identifier	N10 IF $AA_FXS[X1]==1 GOTOF ....
$AA_MM[*axis]*	Measurement result for an axis in the **machine coordinate system**	–	*Axis*: Identifier of an axis (X, Y, Z, ...) traversing when measuring	N10 R1=$AA_MM[X]
$AA_MW[*axis]*	Measurement result for an axis in the **workpiece coordinate system**	–	*Axis*: Identifier of an axis (X, Y, Z, ...) traversing when measuring	N10 R2=$AA_MW[X]
$A..._..._ TIME	Timer for runtime: $AN_SETUP_TIME $AN_POWERON_TIME $AC_OPERATING_TIME $AC_CYCLE_TIME $AC_CUTTING_TIME	0.0 ... 10^{+300} min (read–only value) min (read–only value) s s s	System variable: Time since the control has last booted Time since the control has last booted normally Total runtime of all NC programs Runtime of NC program (only of selected program) Tool action time	N10 IF $AC_CYCLE_TIME==50.5 ....
$AC_..._ PARTS	Workpiece counter: $AC_TOTAL_PARTS $AC_REQUIRED _PARTS $AC_ACTUAL_PARTS $AC_SPECIAL_PARTS	0 ... 999 999 999, integer	System variable: Total actual count Set number of workpiece Current actual count Count of workpieces – specified by the user	N10 IF $AC_ACTUAL_PARTS==15 ....
$AC_MEA [1]	Measurement task status	–	Default condition: 0: Default condition, probe did not switch 1: Probe switched	N10 IF $AC_MEAS[1]==1 GOTOF .... ; if the probe has switched ; continue the program ...
$P_ TOOLNO	Number of the active tool T	–	read–only	N10 IF $P_TOOLNO==12 GOTOF ....
$P_TOOL	Active D number of the active tool	–	read–only	N10 IF $P_TOOL==1 GOTOF ....
$TC_MOP 1[t,d]	Tool life prewarning limit	0.0 ...	in minutes, writing or reading values for tool t, D number d	N10 IF $TC_MOP1[13,1]<15.8 GOTOF ....
$TC_MOP 2[t,d]	Residual tool life	0.0 ...	in minutes, writing or reading values for tool t, D number d	N10 IF $TC_MOP2[13,1]<15.8 GOTOF ....

Address	Meaning	Value Assignment	Information	Programming
$TC_MOP 3[t,d]	Count prewarning limit	0 ... 999 999 999, integer	writing or reading values for tool t, D number d	N10 IF $TC_MOP3[13,1]<15 GOTOF ....
$TC_MOP 4[t,d]	Residual count	0 ... 999 999 999, integer	writing or reading values for tool t, D number d	N10 IF $TC_MOP4[13,1]<8 GOTOF ....
$TC_MOP 11[t,d]	Set tool life	0.0 ...	in minutes, writing or reading values for tool t, D number d	N10 $TC_MOP11[13,1]=247.5
$TC_MOP 13[t,d]	Required count	0 ... 999 999 999, integer	writing or reading values for tool t, D number d	N10 $TC_MOP13[13,1]=715
$TC_TP8[t]	Status of the tool	–	default status – coding by bits for tool t, (bit 0 to bit 4)	N10 IF $TC_TP8[1]==1 GOTOF ....
$TC_TP9[t]	Type of monitoring of the tool	0 ... 2	Monitoring type for tool t, writing or reading 0: No monitoring, 1: Tool life, 2: Count	N10 $TC_TP9[1]=2 ; Select count monitoring
MCALL	Modal subroutine call	–	The subroutine in the block containing MCALL is called automatically after each successive block containing a path motion. The call acts until the next MCALL is called. Application example: Drilling a hole pattern	N10 MCALL CYCLE82(...) ;Separate block, drilling cycle N20 HOLES1(...) ;Row of holes N30 MCALL ;separate block, modal call of CYCLE82(...) completed
MSG( )	Message	max. 65 characters	Message text in inverted commas	N10 MSG("MESSAGE TEXT") ; separate block ... N150 MSG() ; cancels the previous message
OFFN	Groove width with TRACYL, otherwise specification of stock allowance	–	Only effective with the tool radius compensation G41, G42 active	N10 OFFN=12.4
RND	Rounding	0.010 ... 99 999.999	Inserts a rounding with the specified radius value tangentially between two contour blocks	N10 X... Y.... **RND=...** N11 X... Y...
RP	Polar radius	0.001 ... 99 999.999	Traversing in polar coordinates, definition of the pole; in addition: AP – polar angle	see G0, G1, G2; G3 G110, G111, G112
RPL	Angle of rotation with ROT, AROT	±0.00001 ... 359.9999	Specification in degrees; angle for a programmable rotation in the current plane G17 to G19	see ROT, AROT
SET(, , ,) REP()	Set values for the variable fields		SET: Various values, from the specified element up to: according to the number of values REP: the same value, from the specified element up to the end of the field	DEF REAL VAR2[12]=REP(4.5) ; all elements value 4.5 N10 R10=SET(1.1,2.3,4.4) ; R10=1.1, R11=2.3, R4=4.4

Address	Meaning	Value Assignment	Information	Programming
SF	Thread starting point when using G33	0.001 ... 359.999	Specified in degrees; the thread commencement point with G33 is offset by the specified value (not relevant for tapping)	See G33
SPI(n)	converts the spindle number n into the axis identifier		n= 1 or n= 2 axis identifier: e.g. "SP1" or "C"	
SPOS	Spindle position	0.0000 ... 359.9999 If specified incrementally (IC): ±0.001 ... 99 999.999	specified in degrees; the spindle stops at the specified position (to achieve this, the spindle must provide the appropriate technical prerequisites: position control)	N10 SPOS=.... N10 SPOS=ACP(...) N10 SPOS=ACN(...) N10 SPOS=IC(...) N10 SPOS=DC(...)
STOPRE	Preprocessing stop	–	Special function; the next block is only decoded if the block before STOPRE is completed.	STOPRE ;separate block
TRACYL(d)	Milling of the peripheral	d: 1.000 ... 99 999.999	kinematic transformation (only available if the relevant option exists; to be configured)	TRACYL(20.4) ; separate block ; Cylinder diameter: 20.4 mm TRACYL(20.4,1) ; also possible
TRAFOOF	Deactivate TRACYL	–	Disables all kinematic transformations	TRAFOOF ; separate block
TURN	Number of additional circle passes with helix interpolation	0 ... 999	in conjunction with circular interpolation G2/G3 in a plane G17 to G19 and infeed motion of the axis standing vertically on the plane	N10 G0 G17 X20 Y5 Z3 N20 G1 Z–5 F50 N30 G3 X20 Y5 Z–20 I0 J7.5 **TURN=2** **;** in total, 3 full circles

8.2 Positional data

8.2.1 Plane selection: G17 to G19

Functionality

To assign, for example, **tool radius and tool length compensations,** a plane with two axes is selected from the three axes X, Y and Z. In this plane, you can activate a tool radius compensation.
For drill and cutter, the length compensation (length1) is assigned to the axis standing vertically on the selected plane (see Section 8.6 ”Tool and tool offsets”). It is also possible to use a 3–dimensional length compensation for special cases.

Another influence of plane selection is described with the appropriate functions (e.g. Section 8.5 ”Rounding, chamfer”).

The individual planes are also used to define the **direction of rotation of the circle for the circular interpolation** CW or CCW. In the plane in which the circle is traversed, the abscissa and the ordinate are designed and thus also the direction of rotation of the circle. Circles can also be traversed in a plane other than that of the currently active G17 to G19 plane (see Section 8.3 ”Axis movements”).

The following plane and axis assignments are possible:

Table 8-2 Plane and axis assignments

G function	**Plane** (abscissa/ordinate)	**Axis standing vertically on the plane** (length compensation axis when drilling / milling)
G17	X/Y	Z
G18	Z/X	Y
G19	Y/Z	X

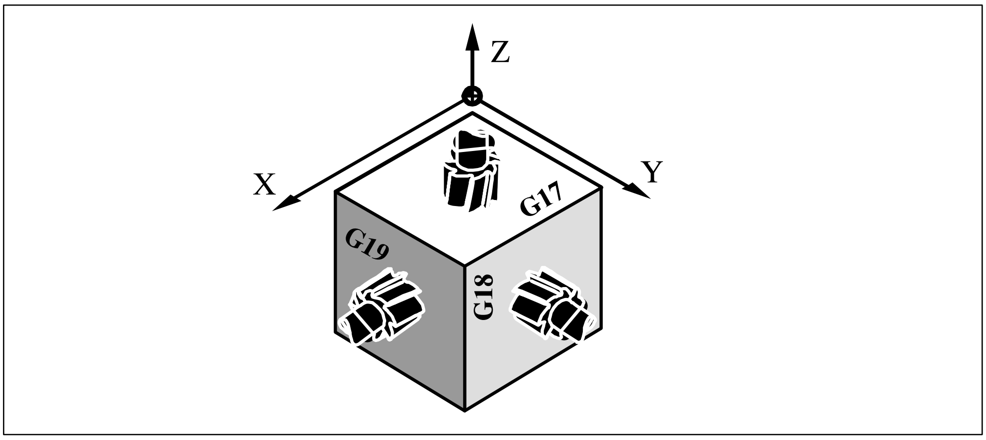


Fig. 8-3 Planes and axes when drilling/milling

Programming example

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

8.2.2 Absolute / incremental dimensioning: G90, G91, AC, IC

Functionality

By using the instructions G90/G91, the written positional data for X,Y, Z,.. are interpreted either as a coordinate target point (G90) or as an axis path to be traversed (G91). G90/G91 applies to all axes.
Irrespective of G90/G91, certain positional data can be specified for certain blocks in absolute/incremental dimensions using AC/IC.

These instructions **do not determine the path** by which the end points are reached; this is provided by a G group (G0,G1,G2, and G3... see Section 8.3 "Axis movements").

Programming

G90	; Absolute dimensioning
G91	; Incremental dimensioning
X=AC(..)	; Absolute dimensioning for a certain axis (here: X axis), non–modal
X=IC(..)	; Incremental dimensioning for a certain axis (here: X axis), non–modal

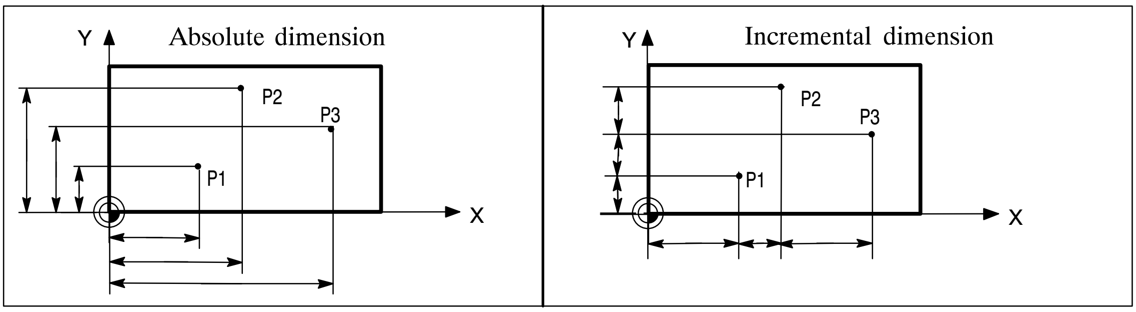


Fig. 8-4 Different dimensioning types in the drawing

Absolute dimensioning G90

With absolute dimensioning, the dimensioning data refers to the **zero point of the currently active coordinate system** (workpiece or current workpiece coordinate system or machine coordinate system). This is dependent on which offsets are currently active: programmable, settable, or no offsets.

Upon program start, G90 is active for **all axes** and remains active until it is deselected in a subsequent block by G91 (incremental dimensioning data) (modally active).

Incremental dimensioning G91

With incremental dimensioning, the numerical value of the path information corresponds to the **axis path to be traversed**. The leading sign indicates the **traversing direction**.

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

Specification with =AC(...), =IC(...)

After the end point coordinate, write an equality sign. The value must be specified in parentheses (round brackets).
Absolute dimensions are also possible for circle center points using =AC(...). Otherwise, the reference point for the circle center is the circle starting point.

Programming example

```
N10 G90 X20 Z90          ;Absolute dimensioning
N20 X75 Z=IC(–32)        ;X dimensioning still absolute, Z incremental dimensioning
...
N180 G91 X40 Z20         ;Switching to incremental dimensioning
N190 X–12 Z=AC(17)       ;X still incremental dimensioning, Z absolute
```

8.2.3 Dimensions in metric units and inches: G71, G70, G710, G700

Functionality

If workpiece dimensions that deviate from the base system settings of the control are present (inch or mm), the dimensions can be entered directly in the program. The required conversion into the base system will then be performed by the control system.

Programming

```
G70      ;Inch dimensional notation
G71      ;Metric dimensional notation

G700     ;Inch dimensional notation, also for feed F
G710     ;Metric dimensional notation, also for feed F
```

Programming example

```
N10 G70 X10 Z30        ;Inch dimensional notation
N20 X40 Z50            ;G70 continues to be active
...
N80 G71 X19 Z17.3      ;Metric dimensional notation from here ...
```

Information

Depending on the **default setting** you choose, the control interprets all geometric values as either metric **or** inch dimensions. Tool offsets and settable zero offsets including their displays are also to be understood as geometrical values; this also applies to the feedrate F in mm/min or inch/min. The default setting can be changed via machine data.
All examples listed in this Manual are based on a **metric default setting**.

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

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

All remaining geometric parameters that are not direct workpiece parameters, such as feedrates, tool offsets, and **settable** work point offsets, are not affected by **G70**/**G71**.
G700/**G710** however, also affects the feedrate F (inch/min, inch/rotation or mm/min, mm/rotation).

8.2.4 Polar coordinates, pole definition: G110, G111, G112

Functionality

In addition to the common specification in Cartesian coordinates (X, Y, Z), the points of a workpiece can also be specified using polar coordinates.
Polar coordinates are also helpful if a workpiece or a part of it is dimensioned from a central point (pole) with specification of the radius and the angle.

Plane

The polar coordinates refer to the plane activated with G17 to G19.
The 3rd axis standing vertically on this plane can additionally be specified. When doing so, spatial specifications can be programmed as cylinder coordinates.

Polar radius RP=...

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

Polar angle AP=...

The angle is always referred to the horizontal axis (abscissa) of the plane (for example, with G17: X axis). Positive or negative angle specifications are possible.
The polar angle remains stored and must only be written in blocks in which it changes, after changing the pole or when switching the plane.

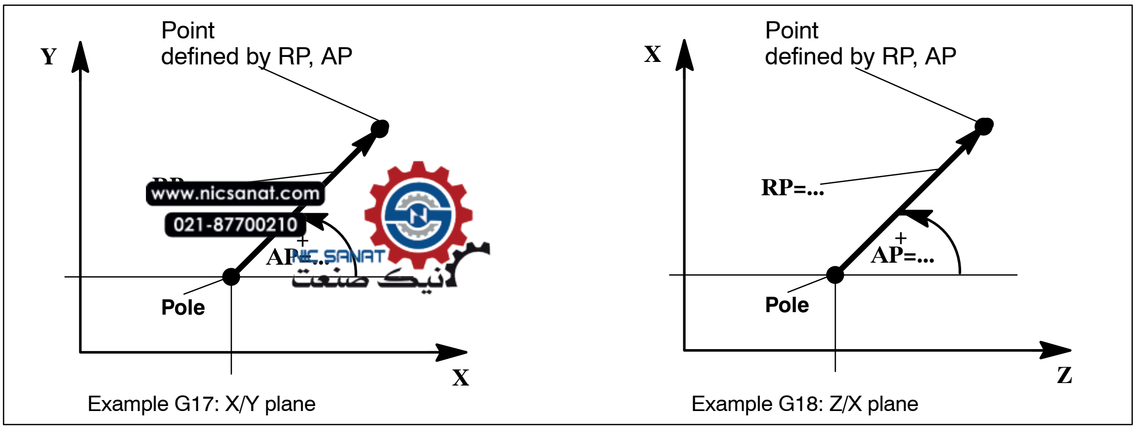


Fig. 8-5 Polar radius and polar angle with definition of the positive direction in different planes

Pole definition, programming

G110	;Pole specification, relative to the last programmed set position (in the plane, for example, with G17: X/Y)
G111	;Pole specification, relative to the zero point of the current workpiece coordinate system (in the plane, for example, with G17: X/Y)
G112	;Pole specification, relative to the last valid pole; preserve plane

Notes

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

Programming example

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

Traversing with polar coordinates

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

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

(see also section 8.3 ”Axis movements”)

8.2.5 Programmable work offset: TRANS, ATRANS

Functionality

The programmable work offset can be used for recurring forms/arrangements in various positions on a workpiece or simply for the selection of a new reference point for the dimensional information or as an allowance for roughing. This results in the **current workpiece coordinate system**. The rewritten dimensions use this as a reference.
The offset is possible in all axes.

Programming

TRANS X... Y... Z... ;Programmable offset; clears old instructions for offset, rotation, scaling factor, mirroring
ATRANS X... Y... Z... ;Programmable offset additively to existing instructions
TRANS ;No values: clears old instructions for offset, rotation, scale factor, mirroring
The instructions with TRANS/ATRANS always requires a separate block.

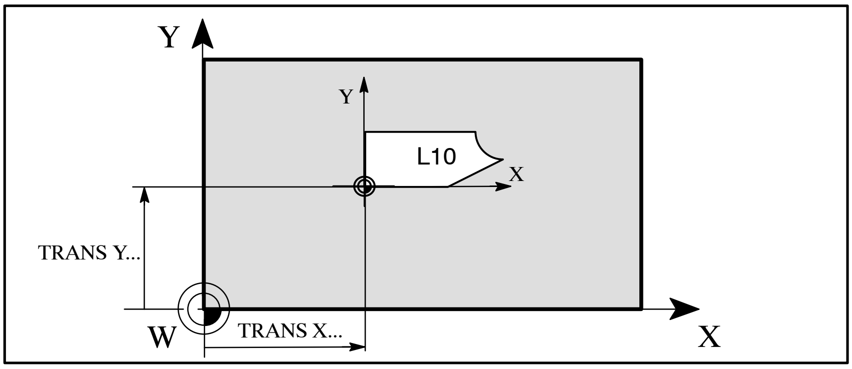


Fig. 8-6 Programmable offset (example)

Programming example

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

Subroutine call – see Section 8.11 ”Subroutine technique ”

8.2.6 Programmable rotation: ROT, AROT

Functionality

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

Programming

ROT RPL=... ;Programmable rotation, clears old instructions for offset, rotation, scale factor, mirroring

AROT RPL=... ;Programmable offset, cumulative with existing ROT instructions

;without values: clears old instructions for offset, rotation, scale factor, mirroring

The instruction with ROT, AROT always requires a separate block.

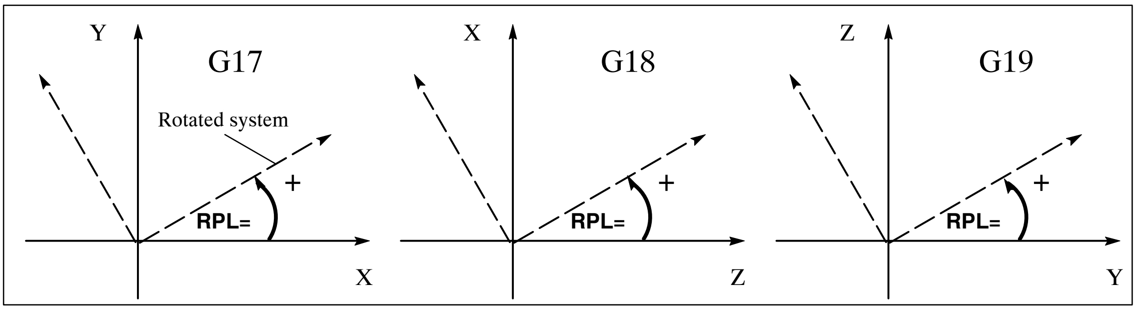


Fig. 8-7 Definition of the positive direction of the angle of rotation in the individual planes

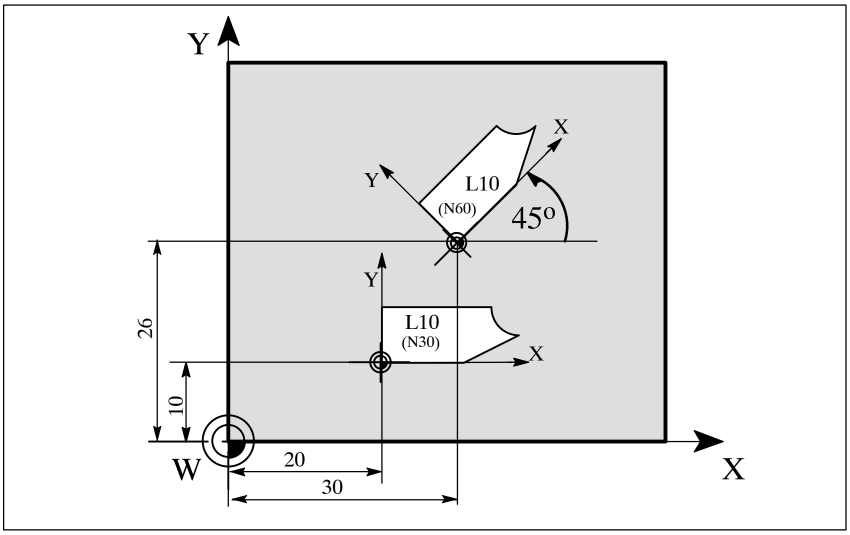


Fig. 8-8 Programming example for programmable offset and programmable rotation

Programming example

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

Subroutine call – see Section 8.11 ”Subroutine technique ”

8.2.7 Programmable scaling factor: SCALE, ASCALE

Functionality

With SCALE, ASCALE, a scaling factor used to zoom in or zoom out the relevant axis can be used for all axes.
The currently set coordinate system is used as the reference for the scale change.

Programming

SCALE X... Y... Z... ;Programmable scaling factor; clears old instructions for offset, rotation, scaling factor, mirroring
ASCALE X... Y... Z... ; Programmable scaling factor; additively to existing instructions

SCALE ; No values: clears old instructions for offset, rotation, scaling factor, mirroring
The instruction with SCALE, ASCALE always requires a separate block.

Notes

- For circles, the same factor should be used in both axes.
- If ATRANS is programmed with SCALE/ASCALE active, these offset values are also scaled.

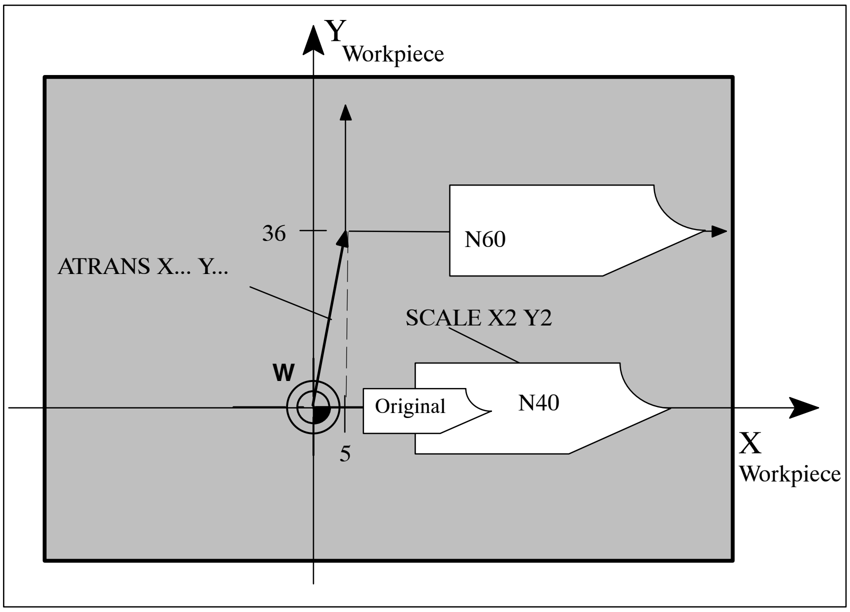


Fig. 8-9 Example for scaling and offset

Programming example

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

For calling a subroutine, see section 8.11 ”Subroutine technique”

8.2.8 Programmable mirroring: MIRROR, AMIRROR

Functionality

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

Programming

MIRROR X0 Y0 Z0 ;Programmable mirroring, clears old instructions for offset, rotation, scaling factor, mirroring
AMIRROR X0 Y0 Z0 ;Programmable mirroring, additive to existing instructions
MIRROR ;No values: clears old instructions for offset, rotation, scaling factor, mirroring

The instruction with MIRROR, AMIRROR always requires a separate block. The axis value has no influence. A value, however, must be specified.

Notes

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

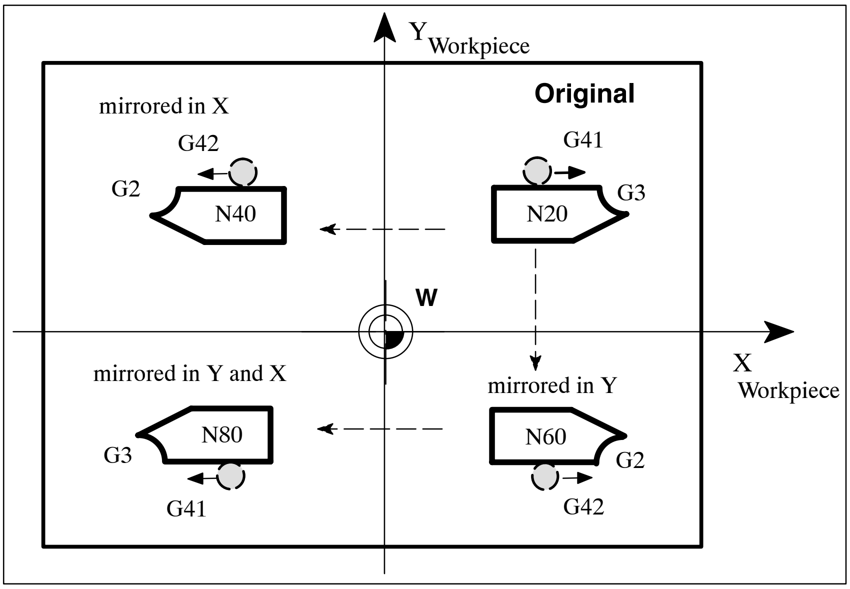


Fig. 8-10 Example for mirroring with the tool position shown

Programming example

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

```
...
N10 G17              ; X/Y plane with Z standing vertically on it
N20 L10              ; Programmed contour with G41
N30 MIRROR X0        ; The direction is changed in X
N40 L10              ; Mirrored contour
N50 MIRROR Y0        ; The direction is changed in Y
N60 L10
N70 AMIRROR X0       ; Repeated mirroring, now in X
N80 L10              ; Contour mirrored twice
N90 MIRROR           ; Mirroring OFF
...
```

Subroutine call – see Section 8.11 ”Subroutine technique ”

8.2.9 Workpiece clamping – settable work offset: G54 to G59, G500, G53, G153

Functionality

The settable work offset specifies the position of the workpiece zero point on the machine (offset of the workpiece zero point with respect to the machine zero point). This offset is determined upon clamping of the workpiece into the machine and must be entered in the corresponding data field by the operator. The value is activated by the program by selection from six possible groupings: G54 to G59.

Note: An inclined workpiece clamping is possible by entering the angles of rotation around the machine axes. These rotation portions are activated with the offset G54 to G59.

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

Programming

```
G54          ;1st settable work offset
G55          ;2nd settable work offset
G56          ;3rd settable work offset
G57          ;4th settable work offset
G58          ;5th settable work offset
G59          ;6th settable work offset
G500         ;Settable work offset OFF – modal

G53          ;Settable work offset OFF – non–modal;
              also suppresses the programmable offset
G153         ;As with G53, but additionally suppresses base frame
```

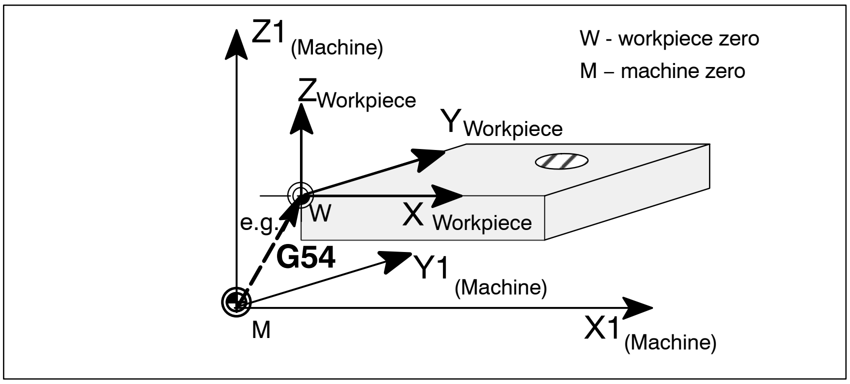


Fig. 8-11 Settable work offset

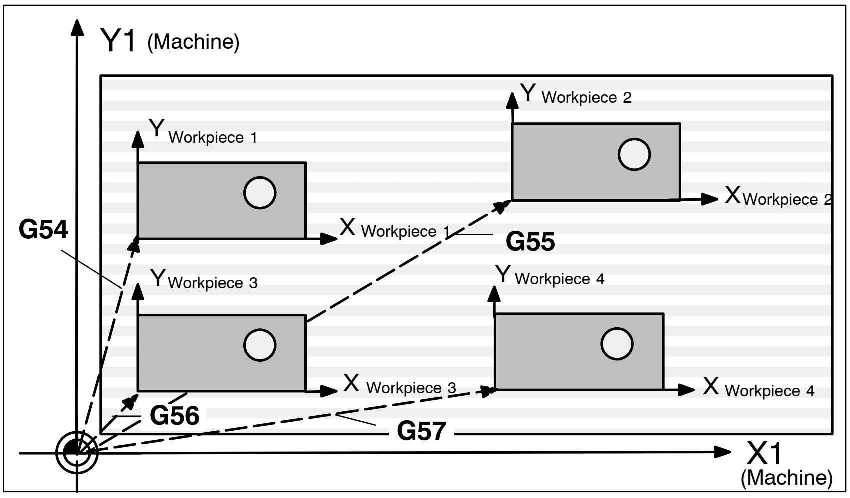


Fig. 8-12 Various workpiece clamping positions when drilling/milling

Programming example

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

Subroutine call – see Section 8.11 ”Subroutine technique ”

8.2.10 Programmable working area limitation: G25, G26, WALIMON, WALIMOF

Functionality

With G25, G26, a working area can be defined for all axes in which it is possible to traverse, with no traversing allowed outside this area. With the tool length compensation active, the tool tip is decisive; otherwise, the toolholder reference point. The coordinate parameters are machine–based.

In order to use the working area limitation, it must be activated in the setting data (under Offset/Setting data/Work area limit) for the respective axis. In this dialog, the values for the working area limitation can be also be preset. This makes them effective in the JOG mode. In the part program, the values for the individual axes can be changed with G25/G26, whereby the values of the working area limitation in the setting data are overwritten. The working area limitation is enabled/disabled in the program by WALIMON/WALIMOF.

Programming

```
G25  X... Y... Z...        ; Lower working area limitation
G26  X... Y... Z...        ; Upper working area limitation

WALIMON                    ; Working area limitation ON
WALIMOF                    ; Working area limitation OFF
```

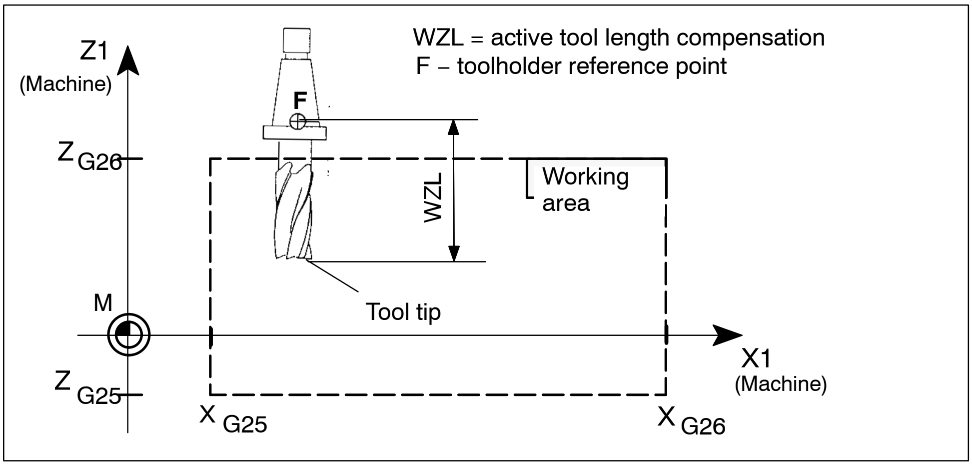


Fig. 8-13 Programmable working area limitation (example: 2–dimensional)

Notes

- For G25, G26, the channel axis identifier consisting of machine data 20080: AXCONF_CHANAX_NAME_TAB is to be used. These can be other than the geometry axis identifiers in MD 20060: AXCONF_GEOAX_NAME_TAB.
- G25 / G26 is also used in connection with the address S for the spindle speed limitation (see also Section ”Spindle speed limitation”).
- A working area limitation can only be activated if the reference point for the relevant axes has been approached.

Programming example

```
N10 G25 X10 Y−20 Z30          ; Lower working area limitation values
N20 G26 X100 Y110 Z300        ; Upper working area limitation values
N30 T1 M6
N40 G0 X90 Y100 Z180
N50 WALIMON                   ; Working area limitation ON
...                           ; Work only within the limitation
N90 WALIMOF                   ; Working area limitation OFF
```

8.3 Axis movements

8.3.1 Linear interpolation with rapid traverse: G0

Functionality

The rapid traverse movement G0 is used for rapid positioning of the tool, but **not for direct workpiece machining**.
All axes can be traversed simultaneously. This results in a straight path.

For each axis, the maximum speed (rapid traverse) is defined in machine data. If only one axis traverses, it uses its rapid traverse. If two or three axes are traversed simultaneously, the path velocity (e.g. the resulting velocity at the tool tip) must be selected such that the **maximum possible path velocity** with consideration of all axes involved results.

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

Programming

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

Note: Another option for linear programming is available with the angle specification ANG=... (see Section 8.5.2 ”Blueprint programming”).

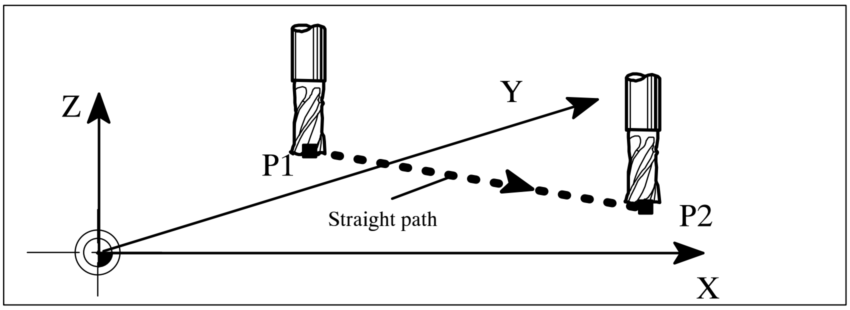


Fig. 8-14 Linear interpolation with rapid traverse from point P1 to P2

Programming example

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

Information

Another group of G functions exists for movement to the position (see Section 8.3.15 ”Exact stop / continuous–path control mode: G60, G64”).
For G60 exact stop, a window with various precision values can be selected with another G group. For exact stop, an alternative instruction with non–modal effectiveness exists: G9. You should consider these options for adaptation to your positioning tasks.

8.3.2 Linear interpolation with feedrate: G1

Functionality

The tool moves from the starting point to the end point along a straight path. The **programmed F word** is decisive for the **path velocity**.
All axes can be traversed simultaneously.
G1 remains active until canceled by another instruction from this G group (G0, G2, G3, ...).

Programming

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

Note: Another option for linear programming is available with the angle specification ANG=... (see Section 8.5.2 ”Blueprint programming”).

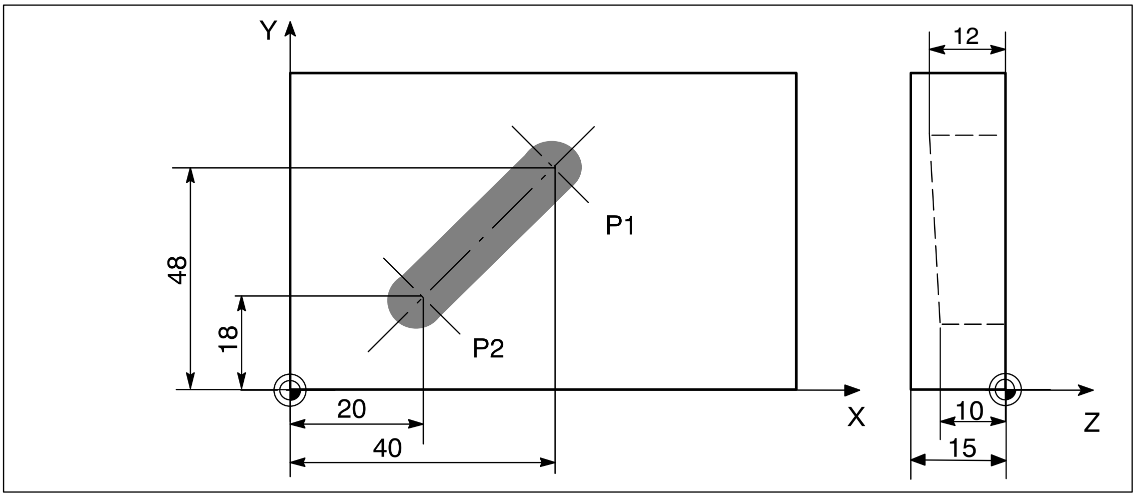


Fig. 8-15 Linear interpolation in three axes using the example of a slot

Programming example

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

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

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 ; CW
G3 ; CCW

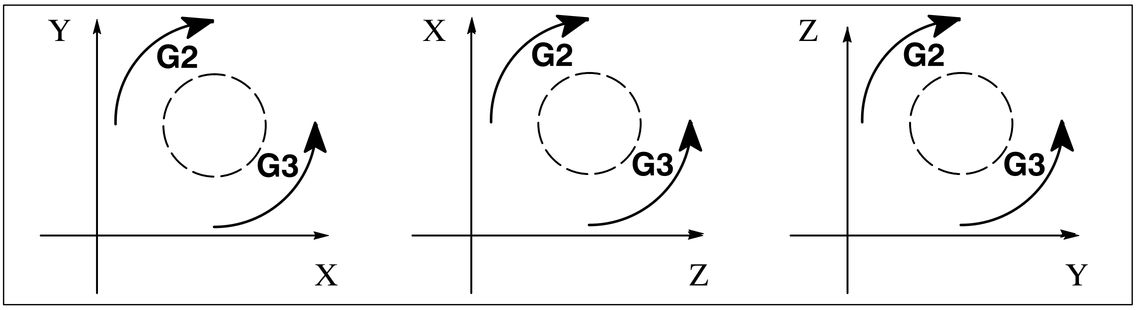


Fig. 8-16 Definition of the direction of rotation of the circle G2/G3 in the 3 possible planes

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

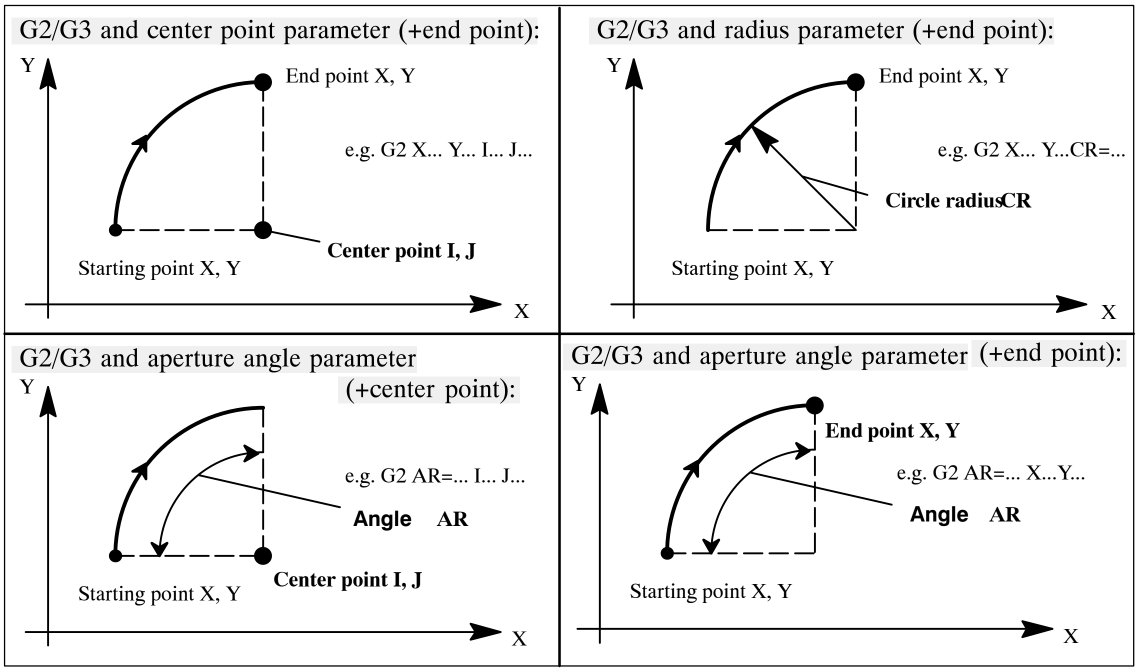


Fig. 8-17 Possibilities of circle programming with G2/G3 using the example of the axes X/Y

G2/G3 acts until it is canceled by another instruction from this G group (G0, G1, ...).
For the **path velocity**, the programmed **F word** is decisive.

Programming

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

Note

Additional options for circular path programming are available with
CT – circle with tangential connection and
CIP – circle via intermediate point (see next sections).

Input tolerances for the circle

Circles are only accepted by the control system with a certain dimensional tolerance. The circle radius at the starting and end points are compared here. If the difference is within the tolerance, the center point is exactly set internally. Otherwise, an alarm message is issued.

The tolerance value can be set via machine data.

Information

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

For circles with radius specification, the arithmetic sign of CR=... is used to select the correct circle. It is possible to program 2 circles with the same starting and end points, as well as with the same radius and the same direction. The negative sign with CR=–... defines the circle whose circle segment is larger than a semicircle; otherwise, the circle with the circle segment smaller or larger than the semicircle is defined as follows:

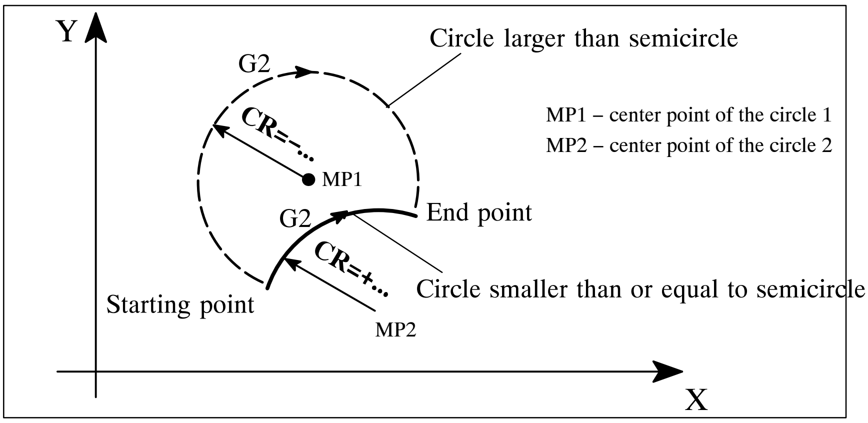


Fig. 8-18 Selection of the circle from two possible circles with radius specification via the sign of CR=

Programming example for center point and end point specification:

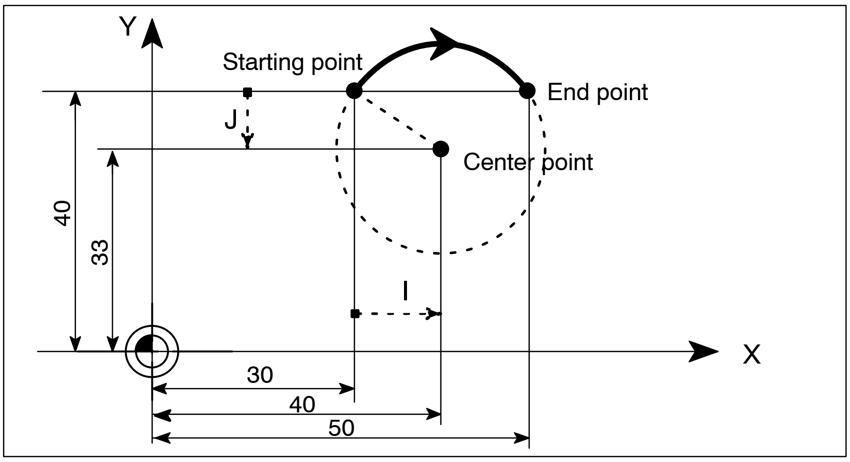


Fig. 8-19 Example for center point and end point specification

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

Note: Center point values refer to the circle starting point!

Programming example for end point and radius specification:

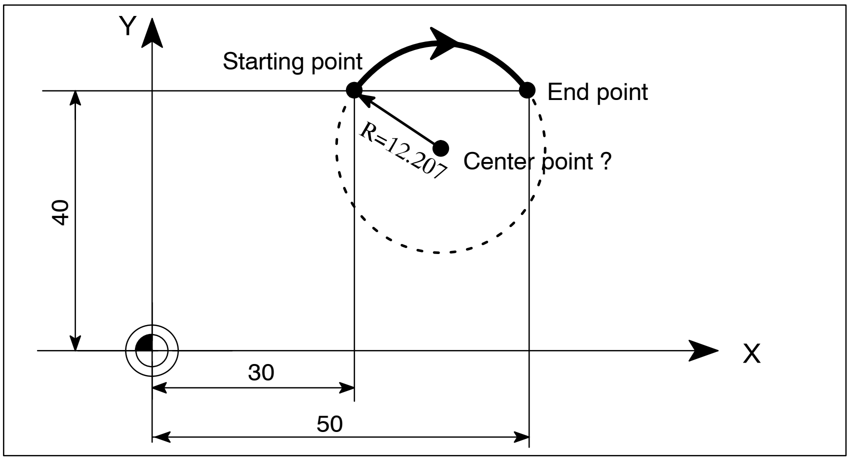


Fig. 8-20 Example for end point and radius specification

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

Note: With a negative leading sign for the value with CR=–..., a circular segment larger than a semicircle is selected.

Programming example for end point and aperture angle:

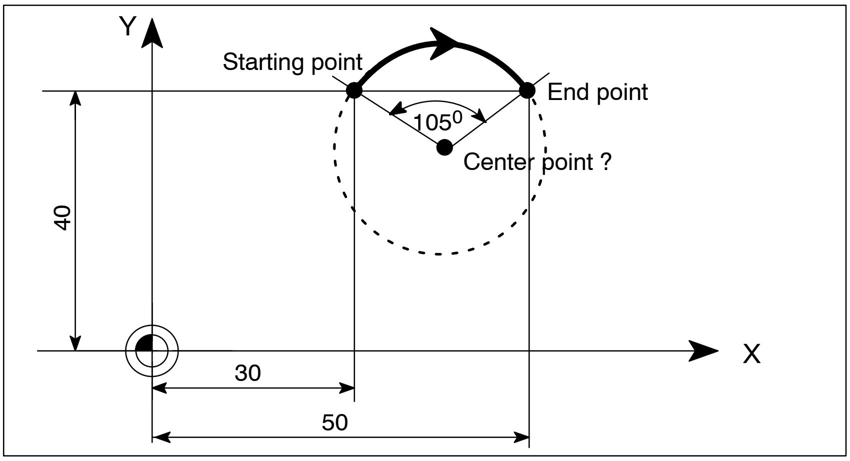


Fig. 8-21 Example for end point and aperture angle specification

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

Programming example for center point and aperture angle:

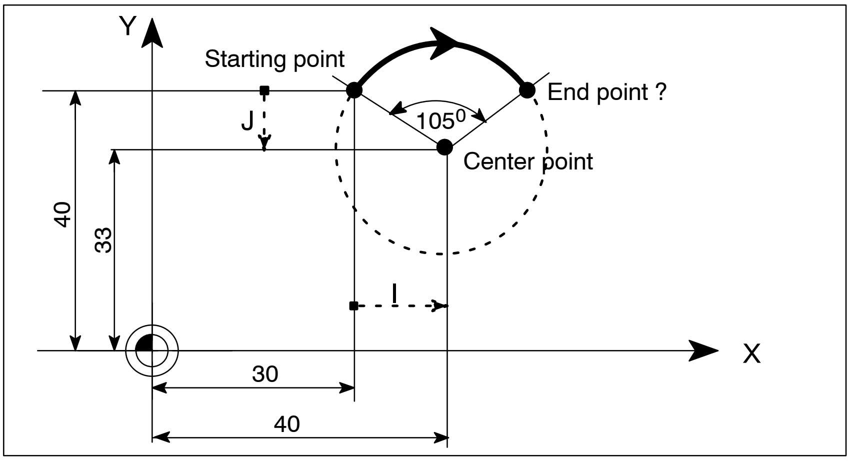


Fig. 8-22 Example for center point and aperture angle specification

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

Note: Center point values refer to the circle starting point!

Programming example for polar coordinates:

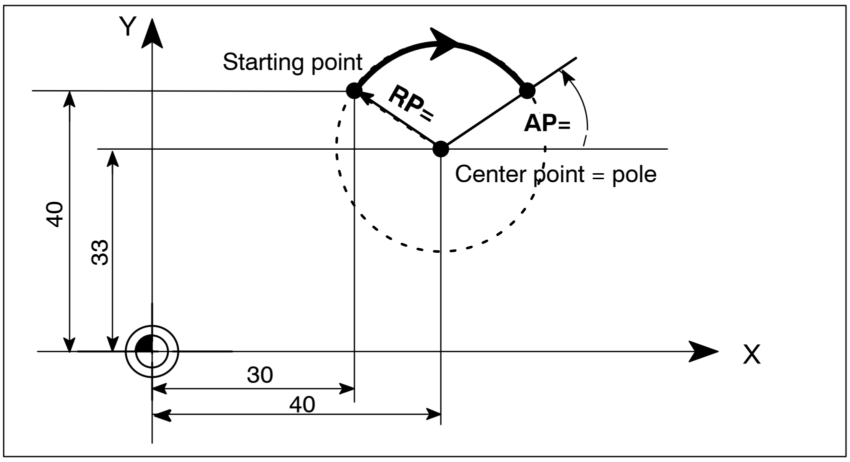


Fig. 8-23 Example for circle with polar coordinates

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

8.3.4 Circular interpolation via intermediate point: CIP

Functionality

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

The direction of the circle results here from the position of the intermediate point (between starting and end points). The intermediate point is written according to the axis assignment as follows:
I1=... for the X axis, J1=... for the Y axis, K1=... for the Z axis.

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

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

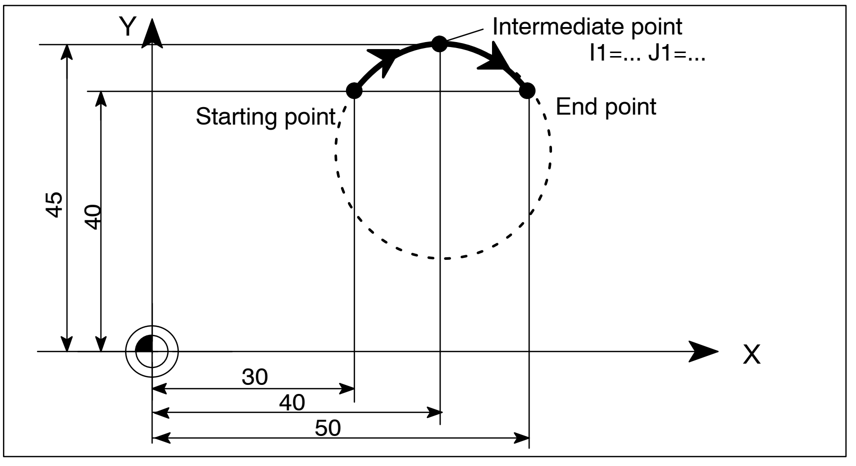


Fig. 8-24 Circle with end point and intermediate point specification using the example of G90

Programming example

```
N5 G90 X30 Y40                ;Circle starting point for N10
N10 CIP X50 Y40  I1=40 J1=45  ;End point and intermediate point
```

8.3.5 Circle with tangential transition: CT

Functionality

With CT and the programmed end point in the current plane G17 through G19, a circle is generated which is connected tangentially to the previous path segment (circle or straight line) in this plane.
This defines the radius and center point of the circle from the geometric relationships of the previous path section and the programmed circle end point.

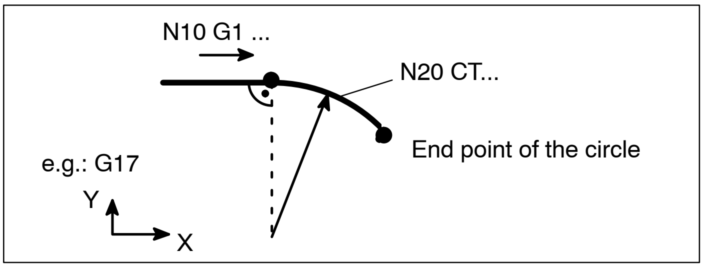


Fig. 8-25 Circle with tangential transition to the previous path section

Programming example

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

8.3.6 Helix interpolation: G2/G3, TURN

Functionality

With helix interpolation, two movements are overlaid:
- the circular movement in plane G17 or G18 or G19, and
- the linear movement of the axis standing vertically on this plane.

The number of additional full–circle passes is programmed with TURN=. These are added to the actual circle programming.
The helix interpolation can preferably be used for the milling of threads or of lubricating grooves in cylinders.

Programming

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

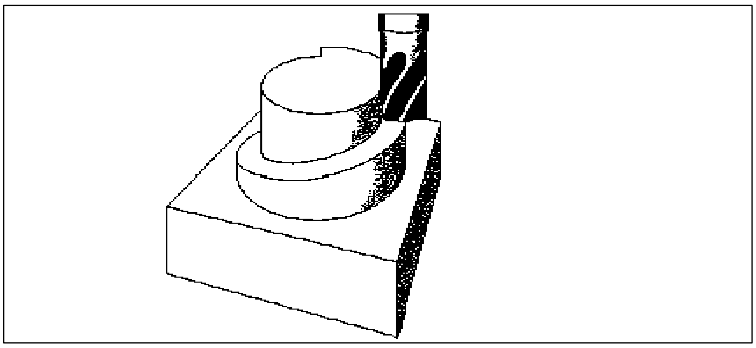


Fig. 8-26 Helix interpolation:

Programming example

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

8.3.7 Thread cutting with constant lead: G33

Functionality

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

The compensating chuck compensates the resulting path differences to a certain limited degree.
The drilling depth is specified via one of the axes X, Y or Z; the thread lead is specified via the relevant I, J or K.

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

Right–hand or left–hand threads

Right–hand or left–hand threads are set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) – see Section 8.4 ”Spindle movement”). To do this, the rotation value must be programmed under address S or a rotation speed set.

Remark:
A complete cycle of tapping with compensating chuck is provided by the standard cycle CYCLE84.

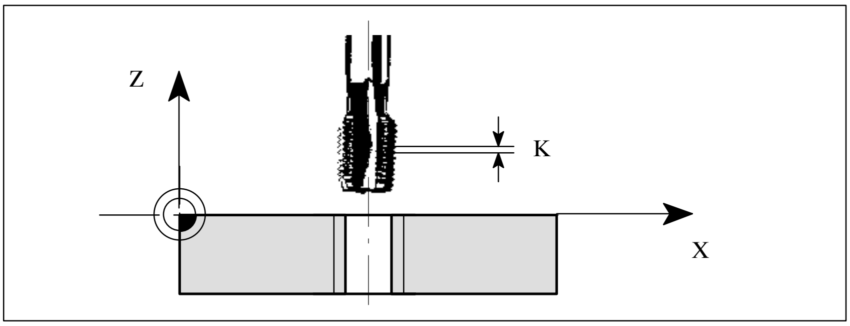


Fig. 8-27 Tapping using G33

Programming example

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

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

Axis velocity

With G33 threads, the velocity of the axis for the thread lengths is determined on the basis of the spindle speed and the thread lead. The **feedrate F is not relevant**. It is, however, stored. However, the maximum axis velocity (rapid traverse) defined in the machine data can not be exceeded. This will result in an alarm.

Information

Important

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

8.3.8 Tapping with compensating chuck: G63

Functionality

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

F [mm/min] = S [rpm] x thread lead [mm/rev.]

The compensating chuck compensates the resulting path differences to a certain limited degree.
The drill is retracted using G63, too, but with the spindle rotating in the opposite direction M3 <−> M4.

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

Right−hand or left−hand threads

Right−hand or left−hand threads are set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) − see Section 8.4 ”Spindle movement”).

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

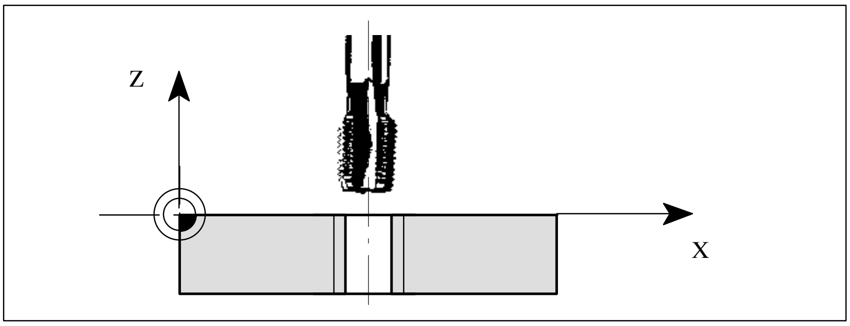


Fig. 8-28 Tapping using G63

Programming example

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

```
N10 G54 G0 G90 X10 Y10 Z5 S600 M3    ; Approach starting point, CW spindle rotation
N20 G63 Z–25 F480                    ; Tapping, end point –25 mm
N40 G63 Z5M4                         ; Retraction, CCW spindle rotation
N50 X... Y... Z...
```

8.3.9 Thread Interpolation: G331, G332

Functionality

This function requires a position–controlled spindle with position measuring system.
By using G331/G332, threads can be tapped **without** compensating chuck, if the dynamics of the spindle and the axis allow it.
If, however, a compensating chuck is used, the path differences to be compensated by the compensating chuck are reduced. This allows tapping at higher spindle speeds.

Drilling is done using G331, retraction is done using G332.
The drilling depth is specified via one of the axes X, Y or Z; the thread lead is specified via the relevant I, J or K.
For G332, the same lead is programmed as for G331. Reversal of the spindle's direction of rotation occurs automatically.
The spindle speed is programmed with S, and without M3/M4.
Before tapping the thread with G331/G332, the spindle must be switched to the position controlled mode with SPOS=... (see also Section 8.4.3 "Spindle positioning").

Right–hand or left–hand threads

The **leading sign of the thread lead** determines the direction of spindle rotation:
Positive: Right–hand (as with M3)
Negative: Left–hand (as with M4)

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

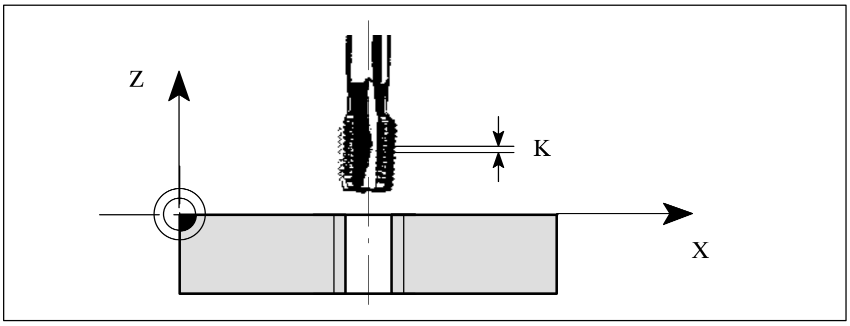


Fig. 8-29 Tapping using G331/G332

Axis velocity

For G331/G332, the speed of the axis for the thread length results from the spindle speed and the thread lead. The **feedrate F is not relevant**. It is, however, stored. However, the maximum axis velocity (rapid traverse) defined in the machine data can not be exceeded. This will result in an alarm.

Programming example

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

```
N5 G54 G0 G90 Y10  Z5          ;Approach starting point
N10 SPOS=0                     ;Spindle in position control mode
N20 G331 Z–25mm K0.8 S600      ;Thread tapping, K positive = clockwise
                               rotation of the spindle, end point –25 mm
N40 G332 Z5 K0.8               ;Retraction
N50 G0 X...Y... Z...
```

8.3.10 Fixed point approach: G75

Functionality

By using G75, a fixed point on the machine, e.g. tool change point, can be approached. The position is stored permanently in the machine data for all axes. No offset is effective. The speed of each axis is its rapid traverse.
G75 requires a separate block and is non–modal. The machine axis identifier must be programmed!
In the block after G75, the previous G command of the "Interpolation type" group (G0, G1,G2, ...) is active again.

Programming example

N10 G75 X1=0 Y1=0 Z1=0

Remark: The programmed position values for X1, Y1 (any value, here = 0) are ignored, but must still be written.

8.3.11 Reference point approach: G74

Functionality

The reference point can be approached in the NC program with G74. The direction and speed of each axis are stored in machine data.
G74 requires a separate block and is non–modal. The machine axis identifier must be programmed!
In the block after G74, the previous G command of the group "interpolation type" (G0, G1, G2, ...) is active again.

Programming example

N10 G74 X1=0 Y1=0 Z1=0

Remark: The programmed position values for X1, Y1 (any value, here = 0) are ignored, but must still be written.

8.3.12 Measuring with touch–trigger probe: MEAS, MEAW

Functionality

If the instruction MEAS=... or MEAW=... is in a block with traversing movements of axes, the positions of the traversed axes for the switching flank of a connected measuring probe are registered and stored. The measurement result can be read for each axis in the program.
For MEAS, the movement of the axes is halted when the selected switching flank of the probe appears and the remaining distance to go is deleted.

Programming

```
MEAS=1   G1 X... Y... Z... F...   ; Measuring with rising edge of the probe,
                                    delete distance–to–go
MEAS=–1  G1 X... Y... Z... F...   ; Measuring with falling edge of the probe,
                                    delete distance–to–go

MEAS=1   G1 X... Y... Z... F...   ; Measuring with rising edge of the probe,
                                    without deletion of the distance–to–go
MEAS=–1 G1 X... Y... Z... F...    ; Measuring with falling edge of the probe,
                                    without deletion of the distance–to–go
```

Caution

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

Measurement task status

If the probe has switched, the variable $AC_MEA[1] after the measuring block has the value =1; otherwise, the value =0.
At the start of a measuring block, the variable is set to the value=0.

Measurement result

When the measuring probe is successfully activated, the result of the measurement is available after the measuring block with the following variables for the axes traversed in the measuring block:
in the machine coordinate system:
$AA_MM[*axis*]
in the workpiece coordinate system:
$AA_MW[*axis*]

Programming example

```
N10 MEAS=1 G1 X300 Z–40 F4000            ; Measuring with deletion of the distance–
to–go, rising edge
N20 IF $AC_MEA[1]==0 GOTOF MEASERR       ; Measuring error ?
N30 R5=$AA_MW[X] R6=$AA_MW[Z]            ; Process measured values
..
N100  MEASERR: M0                        ; Measuring error
```

Note: IF instruction – see Section "Conditional program jumps"

8.3.13 Feedrate F

Functionality

The feed F is the **path velocity** and represents the value of the geometric sum of the velocity components of all involved axes. The individual axis velocities therefore result from the portion of the axis path in the overall distance to be traversed.

The feedrate F is effective for the interpolation types G1, G2, G3, CIP, and CT and is retained until a new F word is written.

Programming

F...

Remark:
For **integer values**, the decimal point is not required, e.g. F300.

Unit of measure for F with G94, G95

The dimension unit for the F word is determined by G functions:

- G94 F **as the feedrate in mm/min**
- G95 F **as feed in mm/rev.** of the spindle
  (only meaningful if the spindle is turning!)

Remark:
This unit of measure applies to metric dimensions. According to Section "Metric and inch dimensioning", settings with inch dimensioning are also possible.

Programming example

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

Remark: Write a new F word if you change G94 – G95!

8.3.14 Feedrate override for circles: CFTCP, CFC

Functionality

With the **tool radius compensation** activated (G41/G42, see Section 8.6.4) **and circle programming**, it is imperative to correct the feedrate at the cutter center point if the **programmed F value** is to act at the circle contour.
Internal and external machining of a circle and the current tool radius are taken into account automatically if the tool radius compensation is enabled.
This feedrate correction (override) is not necessary for linear paths; the path velocities at the cutter center point and at the programmed contour are identical.

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

Programming

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

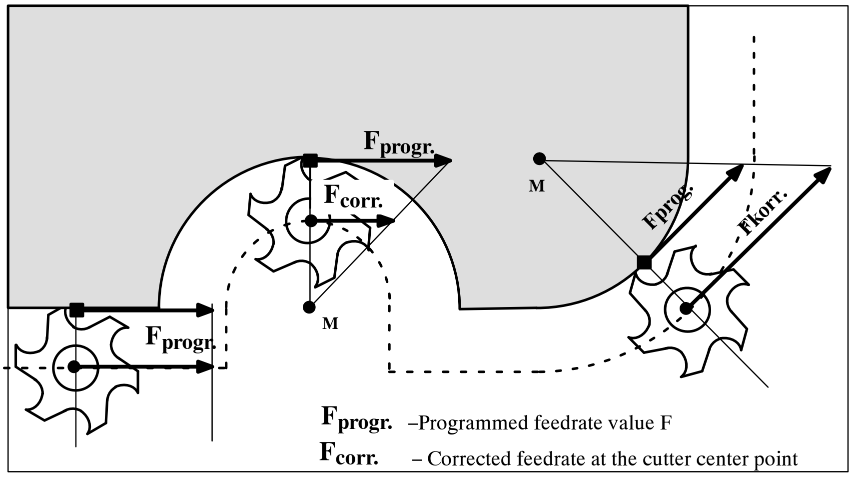


Fig. 8-30 Feedrate override G901 with internal / external machining

Corrected feedrate

- External circle machining: $F_{corr.} = F_{progr.} (r_{cont} + r_{tool}) / r_{cont}$
- Internal circle machining: $F_{corr.} = F_{progr.} (r_{cont} - r_{tool}) / r_{cont}$

r_{cont} : Radius of the circle contour
r_{tool}: Tool radius

Programming example

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

8.3.15 Exact stop / continuous–path control mode: G9, G60, G64

Functionality

To set the movement behavior at the block limits and to continue with the next block, G functions are provided for optimum adaptation to different requirements. Example: For example, you would like to quickly position with the axes or you would like to machine path contours over multiple blocks.

Programming

G60	;Exact stop – modally effective
G64	;Continuous–path–control mode
G9	;Exact stop – non–modal
G601	;Exact stop fine window
G602	;Exact stop coarse window

Exact stop G60, G9

If the exact stop function (G60 or G9) is active, the velocity for reaching the exact end position at the end of a block is decelerated to zero.

Another modal G group can be used here to set when the traversing movement of this block is considered ended and the next block is started.

- G601 Exact stop window fine
  Block advance takes place when all axes have reached the ”Exact stop window fine” (value in the machine data).
- G602 Exact stop window coarse
  Block advance takes place when all axes have reached the ”Exact stop window coarse” (value in the machine data).

The selection of the exact stop window has a significant influence on the total time if many positioning operations are executed. Fine adjustments require more time.

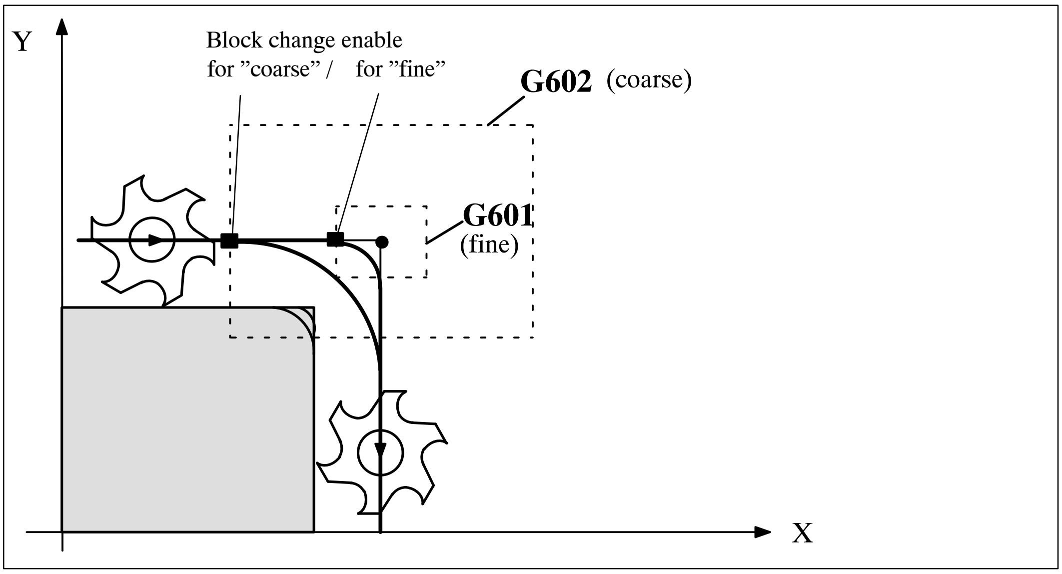


Fig. 8-31 Exact stop window coarse or fine, in effect for G60/G9; enlarged display of the windows

Programming example

```
N5 G602                 ;Exact stop window coarse
N10 G0 G60  X...        ;Exact stop modal
N20 X... Y...           ;G60 is still effective
...
N50 G1 G601 ...         ;Exact stop window fine
N80 G64 Z...            ;Switching to continuous–path control mode
...
N100 G0 G9 Z...         ;Exact stop is only effective for this block
N111 ...                ;Continuous–path control mode again
```

Remark: The command G9 only creates exact stop for the block in which it is programmed; G60, however, is active until it is canceled by G64.

Continuous–path control mode G64

The objective of the continuous–path control mode is to avoid deceleration at the block boundaries and to switch **to the next block** with the most **constant path velocity** possible (during tangential transitions). The function works with **look–ahead velocity control** via several blocks.

For non–tangential transitions (corners), the velocity can reduced rapidly enough so that the axes are subject to a relatively high velocity change over a short time. This may lead to a significant jerk (acceleration change). The magnitude of the jerk can be limited by activating the SOFT function.

Programming example

```
N10 G64 G1 X... F...    ;Continuous–path control mode
N20  Y..                ;Continuous–path control mode continued
...
N180 G60 ...            ;Switching to exact stop
```

Look–ahead velocity control

In the continuous–path control mode with G64, the control system automatically determines the velocity control for several NC block in advance. This enables acceleration and deceleration across multiple blocks with almost tangential transitions. For paths that consist of short sections in the NC blocks, higher velocities can be achieved than without look ahead.

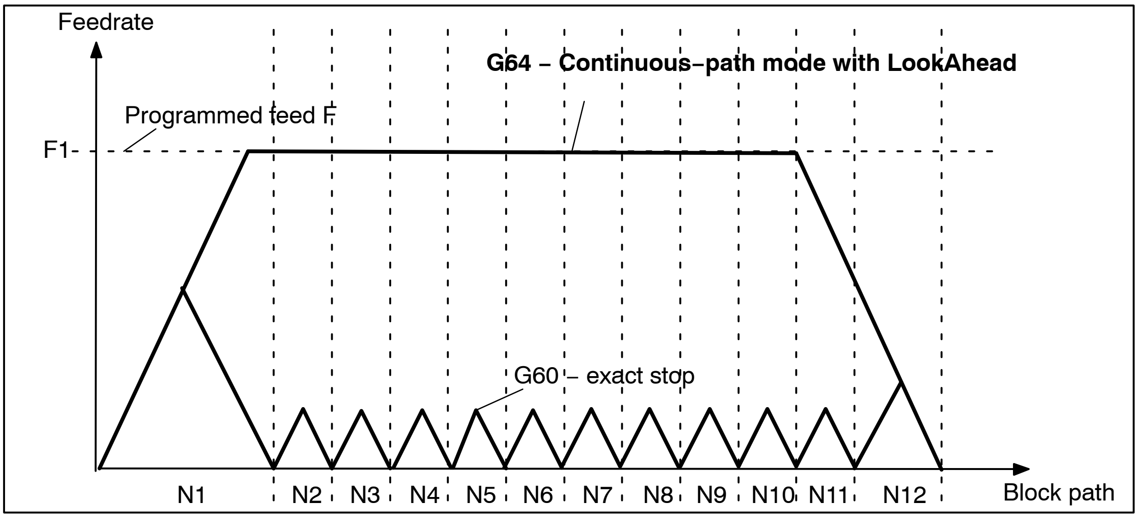


Fig. 8-32 Comparison of the G60 and G64 velocity behavior with short paths in the blocks

8.3.16 Acceleration pattern: BRISK, SOFT

BRISK

The axes of the machine change their speeds using the maximum allowable acceleration value until reaching the final speed. BRISK allows time–optimized working. The set velocity is reached in a short time. However, jumps are present in the acceleration pattern.

SOFT

The axes of the machine accelerate along a nonlinear, constant characteristic until reaching the final velocity. With this jerk–free acceleration, SOFT allows for reduced machine load. The same behavior can also be applied to braking procedures.

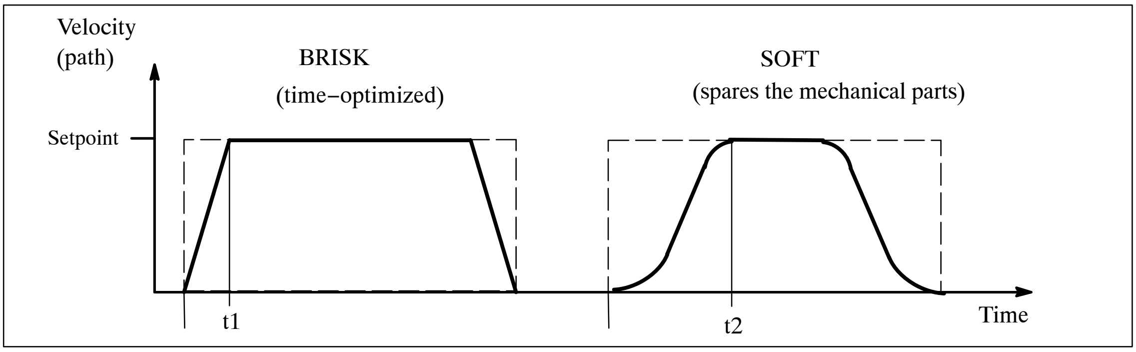


Fig. 8-33 Principle course of the path speed when using BRISK/SOFT

Programming

BRISK	;Stepped path acceleration
SOFT	;Path acceleration with jerk limitation

Programming example

```
N10 SOFT G1 X30 Z84 F650     ;Path acceleration with jerk limitation
...
N90 BRISK  X87 Z104          ;Continue with stepped path acceleration
...
```

8.3.17 Percentage acceleration override: ACC

Functionality

In certain program sections, it can be necessary to change the axis or spindle acceleration set via machine data via the program. This programmable acceleration is a percentage acceleration correction.

For each axis (e.g.: X) or spindle (S), a percentage value 0% and v 200% can be programmed. The axis interpolation is then carried out with this proportional acceleration. The reference value (100%) is the valid machine data value for the acceleration (depending on whether it is the axis or spindle; for the spindle it depends further on the gear step and whether it is positioning mode or speed mode).

Programming

ACC[*axis name*]= percentage value ;for the axis
ACC[S]= percentage value ;for the spindle

Programming example

```
N10 ACC[X]=80        ; 80% acceleration for the X axis
N20 ACC[S]=50        ; 50% acceleration for the spindle
...
N100 ACC[X]=100      ; Deactivate override for the X axis
```

Activation

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

The override is disabled by assigning the value ACC[...] = 100; this can also be achieved using RESET and end of program.

The programmed override value is also active with dry run feedrate.

Caution

A value greater than 100% may only be programmed if this load is permissible for the machine mechanics and the drives have the corresponding reserves. Failure to adhere to the limits can lead to damage to the mechanical parts and/or error messages.

8.3.18 Traversing with feedforward control: FFWON, FFWOF

Functionality

The feedforward control reduces the following error approximately to zero.
Traversing with feedforward control permits higher path accuracy and thus improved machining results.

Programming

```
FFWON                    ; Feedforward control ON
FFWOF                    ; Feedforward control OFF
```

Programming example

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

8.3.19 4th axis

Functionality

Depending on the machine design, a 4th axis could be necessary, for example, if the machine possesses a rotary table, a swing table or the like. The 4th axis can be designed either as a linear or as a rotary axis. Accordingly, the identifier for this axis can be configured, e.g.: U, C, or A etc. For rotary axes, the traversing range can be configured between 0 ...< 360 degrees (modulo–behavior).
The 4th axis can be linear traversed with the remaining axes at the same time with a corresponding machine design. If the axis is traversed in a block with G1 or G2/G3 with the remaining axes (X, Y, Z), it is not assigned a component of the feedrate F; its velocity will depend on the path time of the X, Y and Z axes. Its "linear" movement starts and ends with the remaining path axes. The velocity, however, must not be greater than the defined limiting value.
If only this 4th axis is programmed in a block, with G1, the axis will traverse with the active feedrate F. If it is a rotary axis, the unit of measurement for F is correspondingly degrees/min (with G94) or degrees/revolution of the spindle (with G95).
For this axis, it is also possible to specify (G54 ... G57) and program (TRANS, ATRANS) offsets.

Programming example

Assumed the 4th axis is swivel table (rotary axis) and has the axis identifier A:

```
N5 G94                          ; F in mm/min or degrees/min
N10 G0 X10  Y20 Z30 A45         ; Traverse X–Y–Z path with rapid traverse, A simultaneously
N20 G1 X12  Y21 Z33 A60 F400    ; X–Y–Z path at 400 mm/min, A – simultaneously
N30 G1 A90 F3000                ; A axis traverses alone to 90 deg. position
                                  at a velocity of 3,000 degrees/min
```

Special instructions for rotary axes: DC, ACP, ACN

e.g. for rotary axis A:
A=DC(...) ; Absolute dimensions, approach position directly (along shortest path) Absolute dimensions, approach position in positive direction Absolute dimensions, approach position in negative direction
Example:
N10 A=ACP(55.7) ; Absolute position 55.7 degrees; approach position in positive direction

8.3.20 Dwell Time: G4

Functionality

Between two NC blocks, you can interrupt the machining for a defined time by inserting a **separate block** with G4, for example, for relief cutting.
The words with F... or S... are only used for this block for the specified time. Any previously programmed feedrate F and a spindle speed S remain valid.

Programming

```
G4 F...          ;Dwell time in seconds
G4 S...          ;Dwell time in spindle rev.'s
```

Programming example

```
N5 G1 F200 Z–50 S300 M3     ;Feedrate F, spindle speed S
N10 G4 F2.5      ;Dwell time 2.5 s
N20 Z70
N30 G4 S30       ;Dwell for 30 spindle revolutions, with S=300 rpm and
                  a speed override of 100 %, this corresponds to:  t=0.1 min
N40 X...         ;Feedrate and spindle speed continue to be effective
```

Remark

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

8.3.21 Travel to fixed stop

Functionality

This function is an option and available as of SW 2.0.
The travel to fixed stop (FXS = Fixed Stop) function can be used to establish defined forces for clamping workpieces, such as those required for sleeves and grippers. This function can also be used for approaching mechanical reference points. With sufficiently reduced torque, it is also possible to perform simple measurement operations without connecting a probe.

Programming

```
FXS[axis]=1        ; Select "Travel to fixed stop"
FXS[axis]=0        ; Deselect "Travel to fixed stop"
FXST[axis]=...     ; Clamping torque, specified in % of the max. torque of the drive
FXSW[axis]=...     ; Width of the window for fixed–stop monitoring in mm/degrees
```

Remark: The **machine axis identifier**, e.g: X1, should be used as the axis identifier. X1. The channel axis identifier (e.g.: X) is only permitted, if e.g. no coordinate rotation is active and this axis is directly assigned to a machine axis.

The commands are modal. The traversing path and the selection of the function FXS[axis]=1 must be programmed **in a separate block**.

Programming example – selection

```
N10 G1 G94 ...
N100 X250 Z100 F100 FXS[Z1]=1 FXST[Z1]=12.3 FXSW[Z1]=2
                  ; For the Z1 machine axis   FXS function selected,
                  ; Clamping torque 12.3%,
                  ; Window width 2 mm
```

Notes

- When selected, the fixed stop must be located between the start and end positions.
- Torque (FXST[ ]=) and window width (FXSW[ ]=) may be specified optionally. If they are not written, the value of the existing setting data are used. Programmed values are imported into the setting data. At the start, the setting data are loaded with values from machine data. FXST[ ]=... or FXSW[ ]=... can be changed at any time in the program. The changes are active before traversing movements in the block.

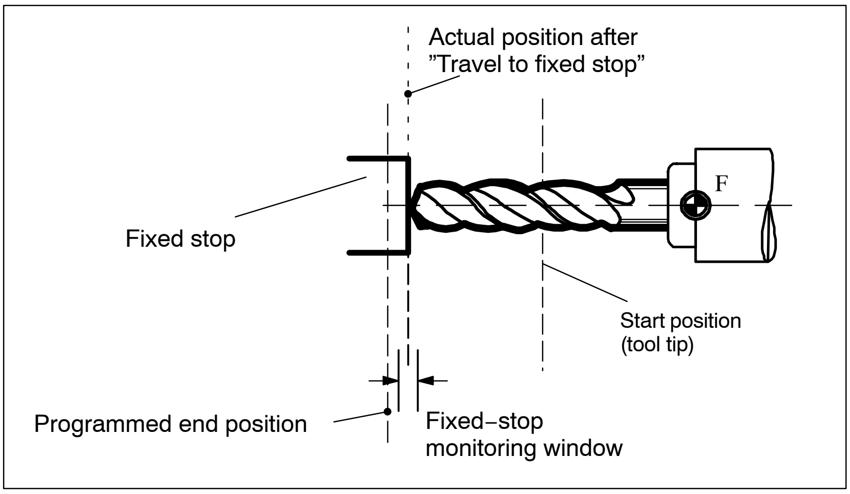


Fig. 8-34 Example of travel to fixed stop: The tool is approached against a stop

Further programming examples

```
N10 G1 G94 ...
N20 X250 Z100 F100 FXS[X1]=1                  ; FXS selected for machine axis X1
                                                Clamping torque and window width as
                                                specified in the SDs
N20 Y250 Z100 F100 FXS[X1]=1 FXST[X1]=12.3    ; FXS selected for machine axis X1
                                                Clamping torque 12.3 %, window width as
                                                specified in the SDs

N20 X250 Y100 F100 FXS[X1]=1 FXST[X1]=2       ; FXS selected for machine axis X1 FXS

                                                Clamping torque 12.3%,
                                                window width 12,3 mm
N20 X250 Z100 F100 FXS[X1]=1 FXSW[X1]=2       ; FXS selected for machine axis
                                                X1, clamping torque as
                                                specified in the SD, window
                                                width 2 mm
```

Fixed stop reached

After the fixed stop has been reached:

- The distance to go is deleted and the position setpoint is followed up.
- The drive torque increases to the programmed limit value FXST[ ]=... or the value from SD and then remains constant.
- The monitoring of the fixed stop is active within the specified window width (FXSW[ ]=... or value from SD).

Deselecting the function

Deselecting the function results in a preprocessing stop. In the block with FXS[X1]=0, traversing movements should stop.

Example:

```
N200 G1 G94 X200 Y400 F200 FXS[X1] = 0        ;The X1 axis is retracted to the position
                                              X= 200 mm.
```

Important

The traversing motion to the retraction position must lead away from the fixed stop; otherwise, the fixed stop or the machine may be damaged.

The block change takes place when the retraction position has been reached. If no retraction position is specified, the block change takes place immediately after the torque limit has been deactivated.

Further information

- ”Measure and delete distance–to–go” (MEAS command) and ”Travel to fixed stop” cannot be programmed in the same block.
- Contour monitoring is not performed while ”Travel to fixed stop” is active.
- If the torque limit is reduced too far, the axis will not be able to follow the specified setpoint; the position controller then goes to the limit and the contour deviation increases. In this operating state, an increase in the torque limit may result in sudden, jerky movements. To make sure that the axis may still follow, make sure that the contour deviation is not greater than with unlimited torque.
- A rate of rise ramp for the new torque limit can be defined in MD to prevent any abrupt changes to the torque limit setting (e.g. insertion of a spindle sleeve or quill).

System variable for status: $AA_FXS[*axis*]

This system variable provides the ”Travel to fixed stop” status for the axis specified:

Value = 0: Axis not at stop
1. Stop was approached successfully
(axis is within fixed–stop monitoring window)
2: Approach to fixed stop has failed (axis is not at fixed stop)

Travel to fixed stop activated
4: Fixed stop was detected
5: Travel to fixed stop is deselected. The deselection is not yet completed.

The interrogation of the system variable in the part program triggers a preprocessing stop.

With the SINUMERIK 802D, only the static states before selection/deselection may be acquired.

Alarm suppression

The issuing of the following alarms can be suppressed with machine data:

- 20091 ”Fixed stop not reached”
- 20094 ”Fixed stop aborted”

References: ”Description of Functions”, Section ”Travel to fixed stop”

8.4 Spindle movements

8.4.1 Spindle speed S, directions of rotation

Functionality

The spindle speed is programmed under the address S in RPM, if the machine has a controlled spindle.
The direction of rotation and the start or end of the movement are specified via M commands (also see Section 8.7 "Miscellaneous function M").

M3	Spindle CW
M4	Spindle CCW
M5	Spindle STOP

Remark: For integer S values, the decimal point can be omitted, e.g. S270

Information

If you write M3 or M4 in a **block with axis movements**, the M commands become active **before** the axis movements.
Default setting: The axis movements will only start once the spindle has accelerated to speed (M3, M4). M5 is also issued before the axis movement. However, it does not wait for the spindle to stop. The axis movements begin before the spindle stops.
The spindle is stopped using program end or RESET.
At program start, spindle speed zero (S0) is in effect.
Remark: Other settings can be configured via machine data.

Programming example

N10 G1 X70 Z20 F300 S270 M3	;**prior** to traversing of the X, Z axes, the spindle accelerates to 270 r.p.m.
...	
N80 S450 ...	;Speed change
...	
N170 G0 Z180 M5	;Z motion in the block; Spindle STOP

8.4.2 Spindle speed limitation: G25, G26

Functionality

In the program, you can limit the limit values that would otherwise apply for a controlled spindle by writing G25 or G26 and the spindle address S with the speed limit value. This overwrites the values entered in the setting data at the same time.
G25 or G26 each requires a separate block. A previously programmed speed S is maintained.

Programming

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

Information

The outmost limits of the spindle speed are set in machine data. By making inputs via the operator panel, setting data can be active for further limiting.

Programming example

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

Note

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

8.4.3 Spindle positioning: SPOS

Functionality

Prerequisite: The spindle must be technically designed for position control.

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

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

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

Exception: The spindle movement is completed first, when the measurement system is not yet synchronized. In this case, the direction is specified in machine data.

Other movement specifications for the spindle are possible with SPOS=ACP(...), SPOS=ACN(...), ... as for rotary axes (see Section ”3rd and 4th axes”).

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

Programming

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

Programming example

```
N10   SPOS=14.3                        ;Spindle position 14.3 degrees
...
N80 G0 X89  Z300 SPOS=25.6   ;Positioning of the spindle with axis motions
                                                This block is ended when both movements are finished.
N81 X200 Z300                        ;The N81 block will only start if the spindle position from
N80.
                                                ;is reached.
```

8.4.4 Gear stages

Function

Up to 5 gear stages can be configured for a spindle for speed / torque adaptation. The gear stage is selected in the program via M commands
(see Section 8.7 ”Miscellaneous function M”):

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

8.5 Contour programming support

8.5.1 Rounding, chamfer

Functionality

In a contour corner, you can insert the elements chamfer or rounding.
The respective instruction CHF= ... or RND=... is written in the block, which leads to the corner.

Programming

CHF=... ;insert chamfer, value: **Length** Side length of the chamfer
RND=... ;Insert rounding, value: Radius of the rounding

Information

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

Note:
The programmed value for chamfer and rounding is automatically reduced if the contour length of an involved block is insufficient.
No chamfer/rounding is inserted, if

- more than three blocks in the connection are programmed that do not contain any information for traversing in the plane
- or a plane change is carried out.

Chamfer CHF=

A linear contour element is inserted between **linear and circle contours** in any combination. The edge is broken.

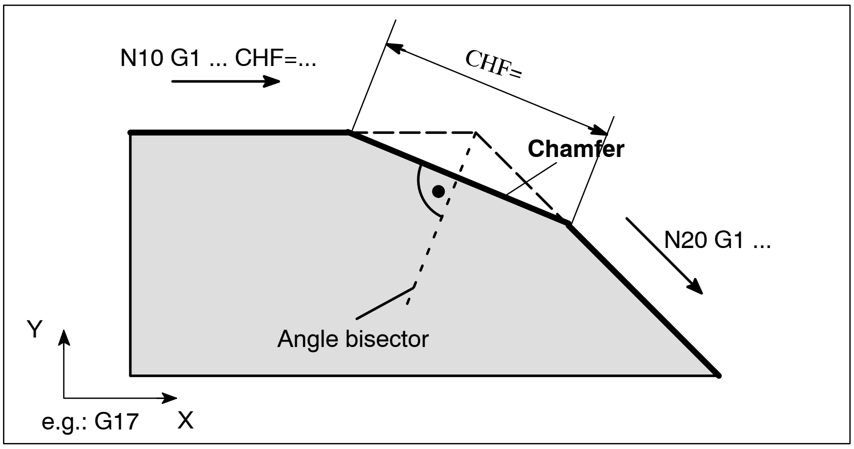


Fig. 8-35 Inserting a chamfer CHF between to straight lines (example)

Programming example for a chamfer

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

Rounding RND=

A circle contour element can be inserted with tangential link between the **linear and circle contours** in any combination.

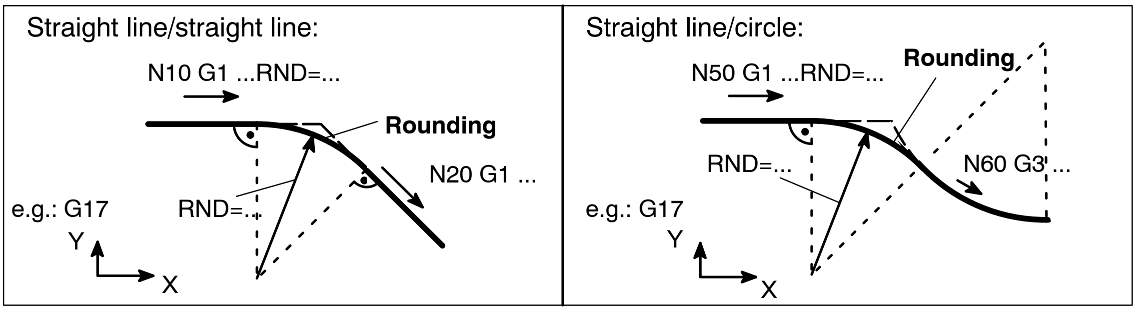


Fig. 8-36 Inserting roundings as examples

Programming example for a rounding

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

8.5.2 Blueprint programming

Functionality

If direct end point values for the contour are not visible in a machining drawing, angle values can also be used for straight line determination. In a contour corner, you can insert the elements chamfer or rounding. The respective instruction CHR= ... or RND=... is written in the block, which leads to the corner.
The contour definition programming can be used in blocks with G0 or G1.
Theoretically, any number of straight line blocks can be linked and a rounding or a chamfer can be inserted between them. Every straight line must be clearly identified by point values and/or angle values.

Programming

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

Angle ANG=

If only one end point coordinate of the plane is known for a straight line, or for contours across multiple blocks the cumulative end point, an angle parameter can be used for unique definition of the straight line path. The angle is always referred to the abscissa of the current plane G17 to G19; e.g.: with G17 to the X axis. Positive angles are aligned counterclockwise.

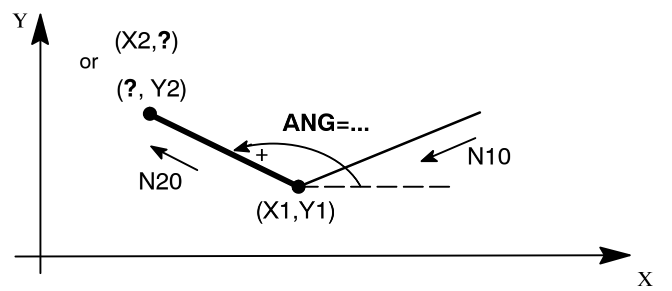
Contour	Programming
	End point in N20 not completely known N10 G1 X1 Y1 N20 X2 ANG=... or: N10 G1 X1 Y1 N20 Y2 ANG=... The values are only examples.

Fig. 8-37 Specification of an angle for determination of a straight line using the example of the G17 plane

Rounding RND=

A contour element is inserted into the corner of two linear blocks with tangential corner (see also Fig. 8-36).

Chamfer CHR=

Another linear contour element (chamfer) is inserted into the corner of two linear blocks. The programmed value is the side length of the chamfer.

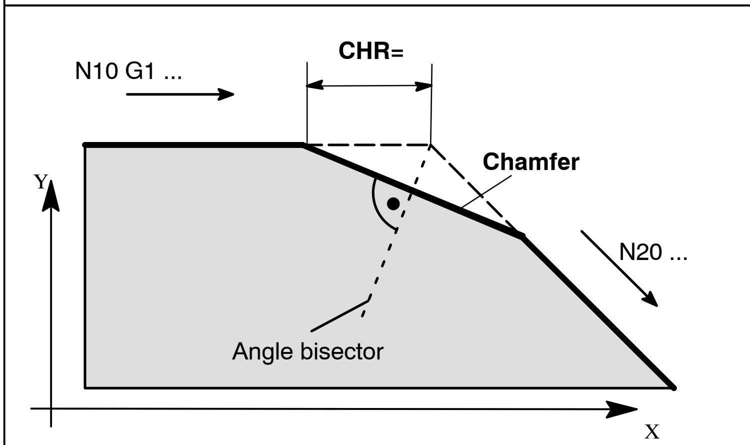
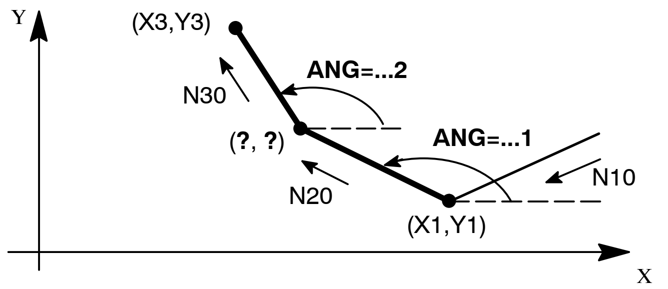
Contour	Programming
	Insert a chamfer with side length e.g. 5 mm: N10 G1 X... CHR=5 N20 X... Y..

Fig. 8-38 Inserting a chamfer using CHR

<table>
<tr><th>Contour</th><th>Programming</th></tr>
<tr><td>
</td><td>End point in N20 unknown

N10 G1 X1 Y1
N20 ANG=...1
N30 X3 Y3 ANG=...2

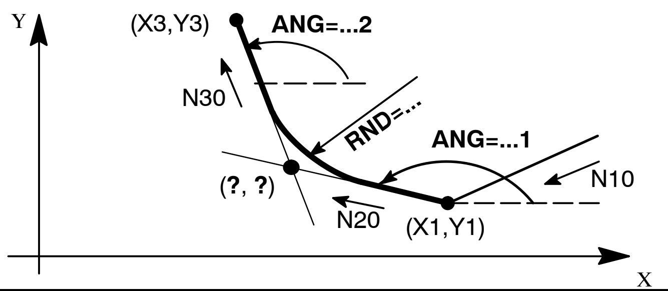
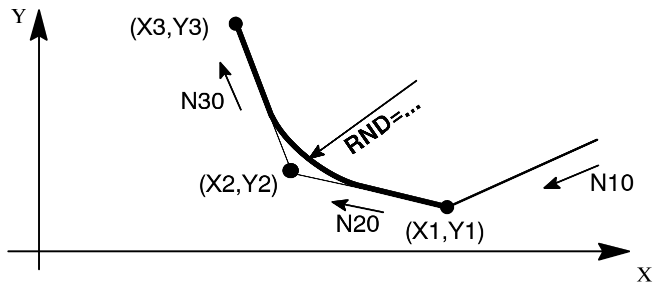
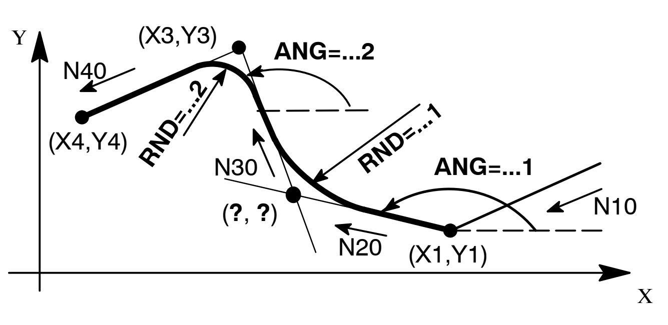
The values are only examples.</td></tr>
<tr><td>
</td><td>End point in N20 unknown;
insert chamfer:
N10 G1 X1 Y1
N20 ANG=...1 RND=...
N30 X3 Y3 ANG=...2
analogously
Insert chamfer:
N10 G1 X1 Y1
N20 ANG=...1 CHR=...
N30 X3 Y3 ANG=...2</td></tr>
<tr><td>
</td><td>End point in N20 known;
insert rounding:
N10 G1 X1 Y1
N20 X2 Y2 RND=...
N30 X3 Y3
analogously
Insert chamfer:
N10 G1 X1 Y1
N20 X2 Y2 CHR=...
N30 X3 Y3</td></tr>
<tr><td>
</td><td>End point in N20 unknown;
insert roundings:
N10 G1 X1 Y1
N20 ANG=...1 RND=...1
N30 X3 Y3 ANG=...2 RND=...2
N40 X4 Y4
analogously
Insert chamfer:
N10 G1 X1 Y1
N20 ANG=...1 CHR=...1
N30 X3 Y3 ANG=...2 CHR=...2
N40 X4 Y4</td></tr>
</table>

Fig. 8-39 Multiple block contours using the example of the G17 plane

Information

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

Notes:

- If radius and chamfer are programmed in one block, only the radius is inserted regardless of the programming sequence.
- In fields other than blueprint programming, you will also find specification of a chamfer in the form CHF=. In such cases, the value after CHF= is the chamfer length, instead of CHR=.

8.6 Tool and tool offset

8.6.1 General notes

Functionality

When creating programs for machining workpieces, it is not necessary to take into account the tool length or the tool radius. You can program workpiece dimensions directly, e.g. as specified in the drawing.

You enter the tool data separately in a special data section.
You will simply call the required tool with its offset data in the program and enable the tool radius compensation if necessary. The control executes the required path corrections based on this data to create the described workpiece.

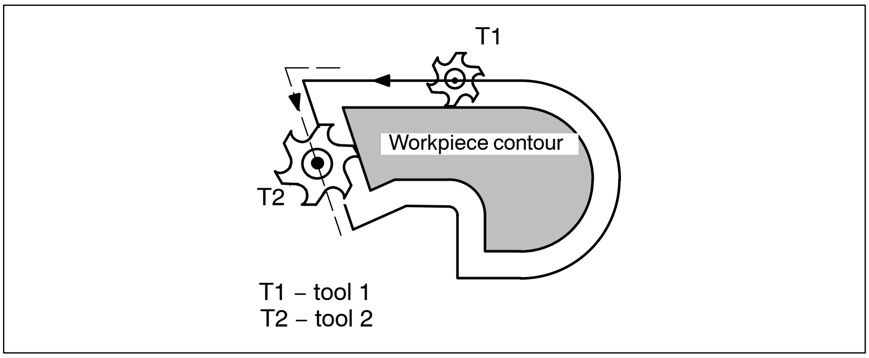


Fig. 8-40 Machining of a workpiece with different tool

radii

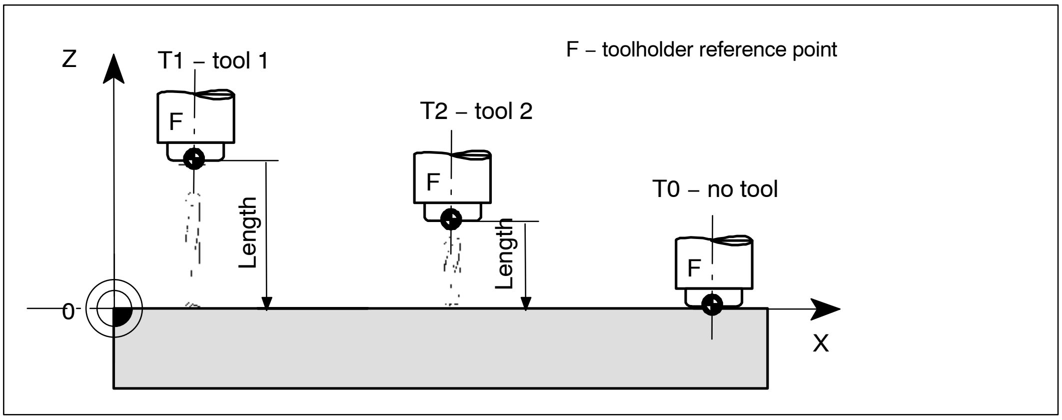


Fig. 8-41 Approaching the workpiece position Z0 – different length compensations

8.6.2 Tool T

Functionality

The tool selection takes place when the T word is programmed. Whether this is a **tool change** or only a **preselection**, is defined in the machine data:

- The tool change (tool call) is performed either directly using the T word or
- the change takes place after the preselection with the T word by an additional instruction **M6** (see also Section 8.7 "Miscellaneous functions M").

Please note:
If a certain tool was activated, it remains stored as an active tool even beyond the end of the program and after turning off / turning on the control system.
If you change a tool manually, input the change also in the control system so that the control system 'knows' the correct tool. For example, you can start a block with the new T word in the MDA mode.

Programming

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

Note

Max. **48** 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 using M6:
N10 T14  ...      ;Preselect tool 14
...
N15 M6            ;Execute tool change, thereafter, T14 is active
```

8.6.3 Tool offset number D

Functionality

It is possible to assign between 1 to 9 (12) data fields with different tool offset blocks (for multiple cutting edges) to a specific tool. If a special cutting tool is required, it can be programmed with D and the corresponding number.
If no D word is written, D1 is **automatically** in effect.
If **D0** is programmed, the offsets for the tool are **ineffective**.

Note

Max. **96** 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!

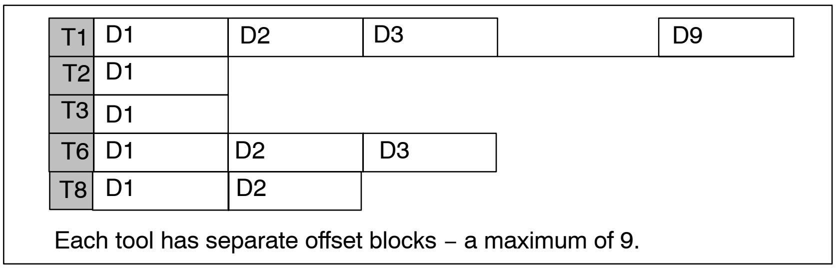


Fig. 8-42 Examples for assigning tool offset numbers / tool

Information

Tool length offsets are active **immediately** if the tool is active; if no D numbers have been programmed, the values of D1 will be used.
The offset is applied with the first programmed traverse of the respective length compensation axis. Observe any active G17 to G19.

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

Programming example

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

```
N5 G17            ;Defines the axis assignment for corrections
N10 T1            ;Tool 1 with the relevant D1 is activated
N11 G0 Z...       ;With G17, Z is the length compensation axis; the length
                   compensation is overlaid here
N50 T4 D2         ;Load tool 4, D2 of T4 is active
...
N70 G0 Z... D1    ;D1 is active for tool 4, only edge changed
```

Tool change **using the M6** command:

```
N5 G17          ;Defines the axis assignment for compensations
N10 T1          ;Tool preselection
...
N15 M6          ;Tool change, T1 with appropriate D1 is active
N16 G0 Z...     ;With G17, Z is the length compensation axis; the
                length compensation is overlaid here
...
N20 G0 Z... D2  ;D2 is active for tool 1; with G17, Z is the length compensation axis; the
                difference of the length compensation D1–>D2 is overlaid here
N50 T4          ;Tool preselection T4, please note: T1 with D2 is still active !
...
N55 D3 M6       ;Tool change, T4 with the appropriate D3 is active
...
```

Contents of an offset memory

Enter the following in the offset memory:

- Geometric quantities: Length, radius
  These consist of several components (geometry, wear). The control computes the components to a certain dimension (e.g. overall length 1, total radius). The respective overall dimension becomes effective when the offset memory is activated.
  How these values are calculated in the axes is determined by the tool type and the commands G17, G18, G19 (see following illustrations).
- Tool type
  The tool type (drill, cutter) defines which geometry data are necessary and how they are taken into account.

Tool special cases

For the tool types 'cutter' and 'drill', the parameters for length 2 and length 3 are only required for special cases (e.g.: multidimensional length compensation when using an angular head attachment).

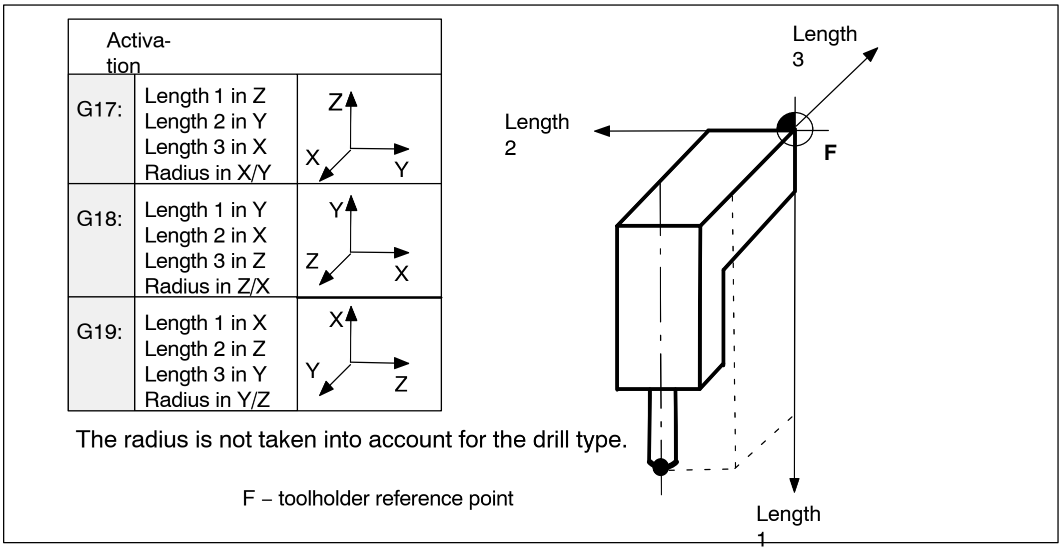


Fig. 8-43 Effect of the tool length compensation – 3D (special case)

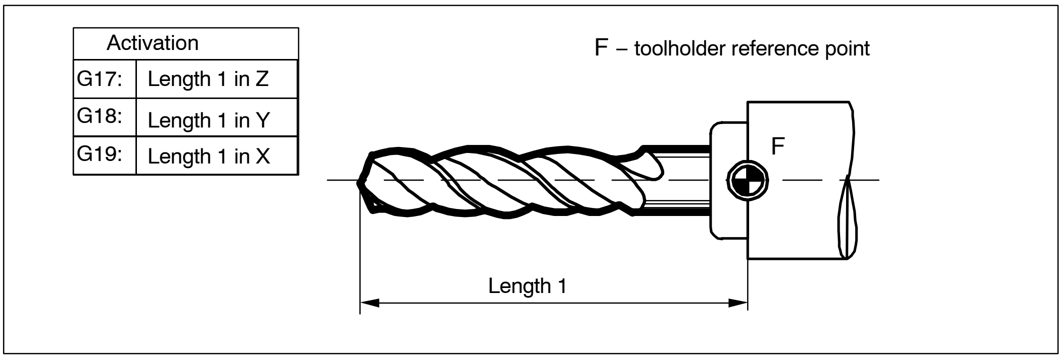


Fig. 8-44 Effect of the offsets with the tool type 'drill'

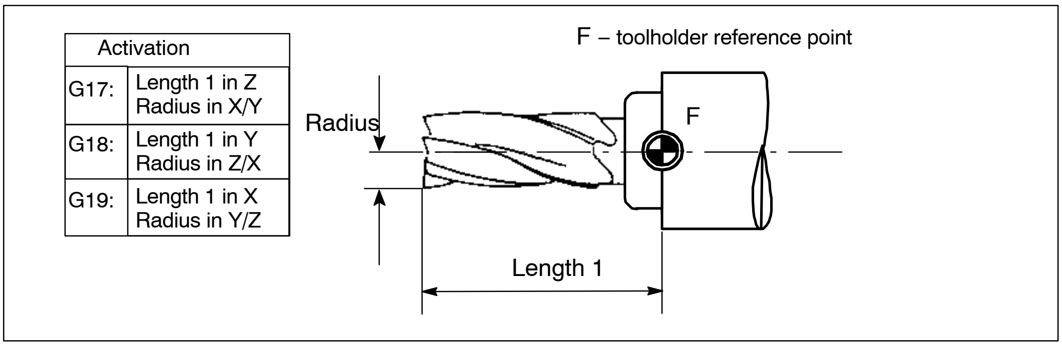


Fig. 8-45 Effect of the offsets with the tool type 'cutter'

8.6.4 Selecting the tool radius compensation: G41, G42

Functionality

The control system is working with tool radius compensation in the selected plane G17 to G19.
A tool with a corresponding D number must be active. The tool radius compensation is activated by G41/G42. The controller automatically calculates the required equidistant tool paths for the programmed contour for the respective current tool radius.

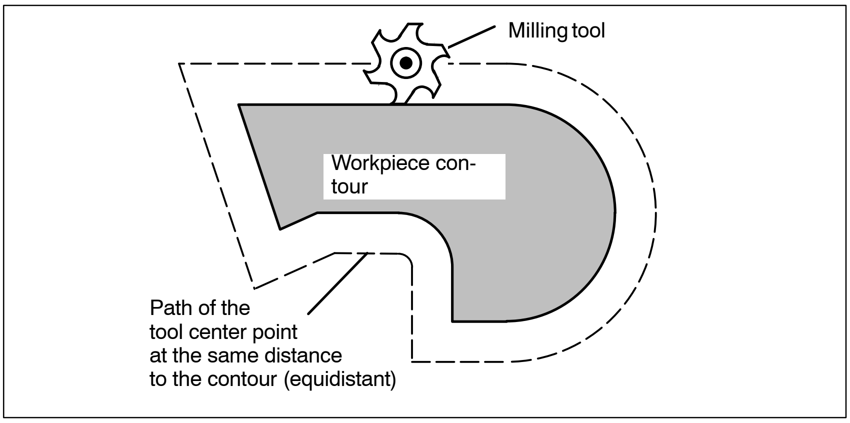


Fig. 8-46 Tool radius compensation

Programming

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

Remark: The selection can only be made for linear interpolation (G0, G1).
Program both axes of the plane (e.g. with G17: X, Y). If you only specify one axis, the second axis is automatically completed with the last programmed value.

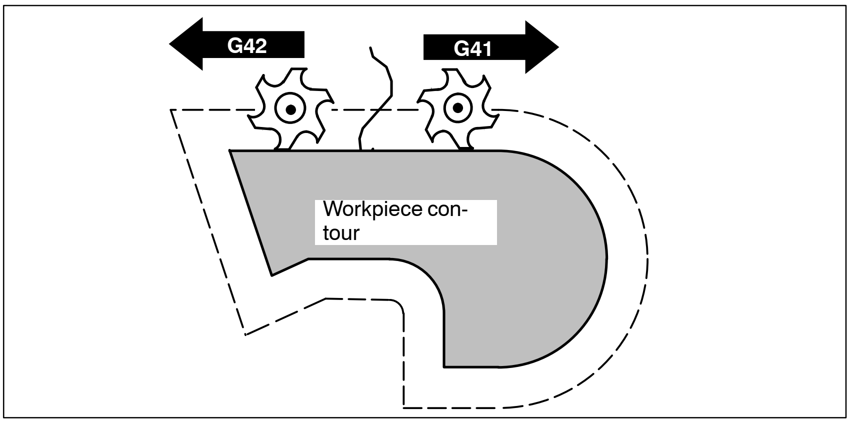


Fig. 8-47 Compensation to the right/left of the contour

Starting the compensation

The tool travels directly to the contour along a straight line and is positioned perpendicular to the path tangent at the starting point of the contour.
Select the starting point such that a collision−free travel is ensured.

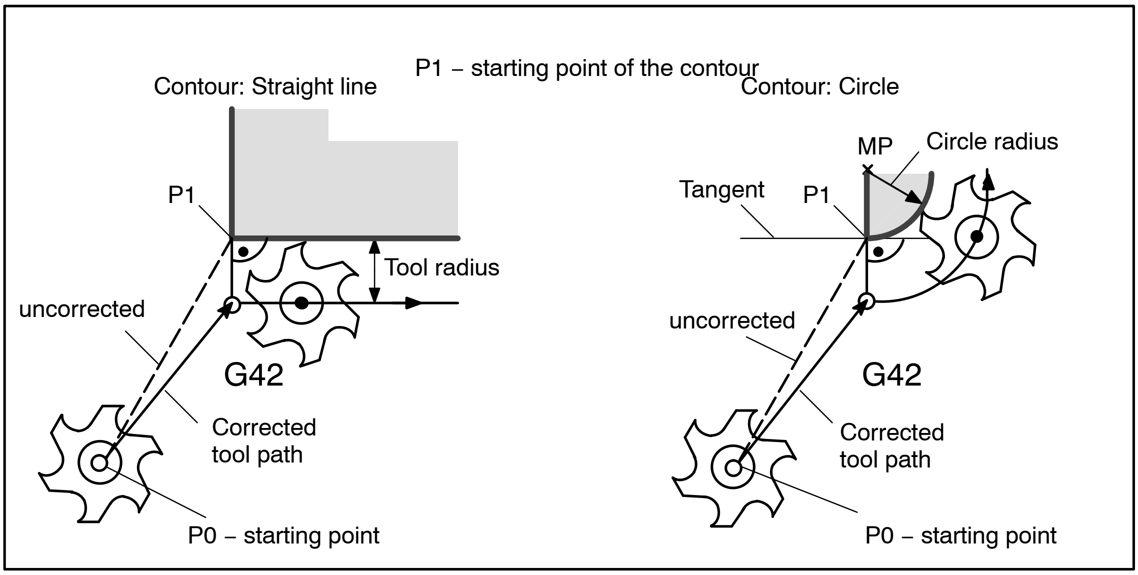


Fig. 8-48 Start of the tool radius compensation with G42 as example

Information

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

Programming example

```
N10 T...
N20 G17 D2 F300           ;Compensation no. 2, feedrate 300 mm/min
N25 X... Y...             ;P0 – starting point
N30 G1  G42 X... Y...     ;Selection right of the contour, P1
N31 X... Y...             ;Starting contour, circle or straight line
```

After selection, it is also possible to execute blocks with infeed movements or M outputs:

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

8.6.5 Corner behavior: G450, G451

Functionality

Using the functions G450 and G451, you can set the behavior for non–continuous transition from one contour element to another contour element (corner behavior) when G41/G42 is active.
Internal and external corners are detected by the control system itself. For inside corners, the intersection of the equidistant paths is always approached.

Programming

G450 ;Transition circle
G451 ;Point of intersection

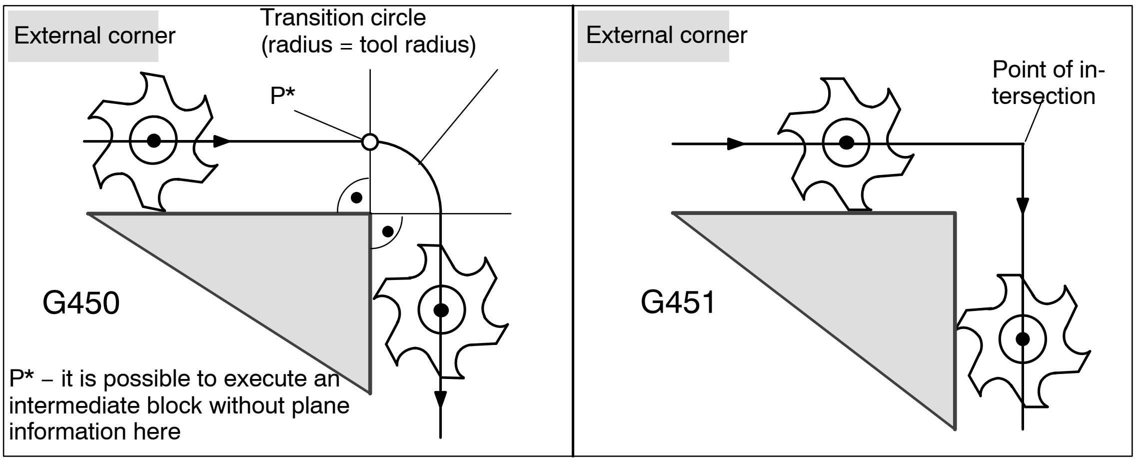


Fig. 8-49 Corner behavior at an external corner

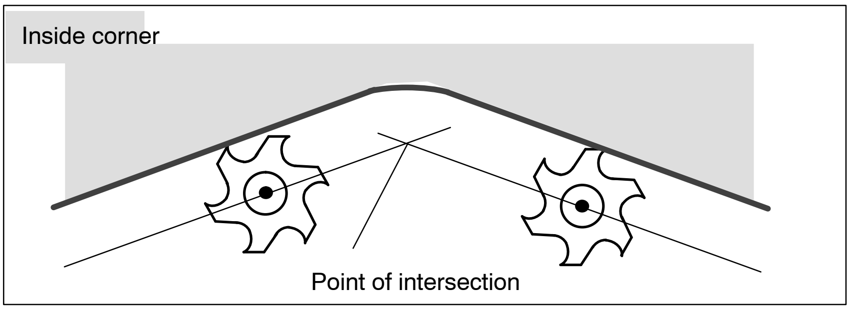


Fig. 8-50 Corner behavior at an internal corner

Transition circle G450

The tool center point travels around the workpiece external corner in an arc with the tool radius.
In view of the data, for example, as far as the feedrate value is concerned, the transition circle belongs to the next block containing traversing movements, for example, with reference to the feedrate value.

Point of intersection G451

For a G451 intersection of the equidistant paths, the point (intersection) that results from the center point paths of the tool (circle or straight line) is approached.
With acute contour angles and active point of intersection, depending on the tool radius, unnecessary idle motions could result for the tool.
In this case, the control automatically switches to transition circle for this block if a certain set angle value (100°) is reached.

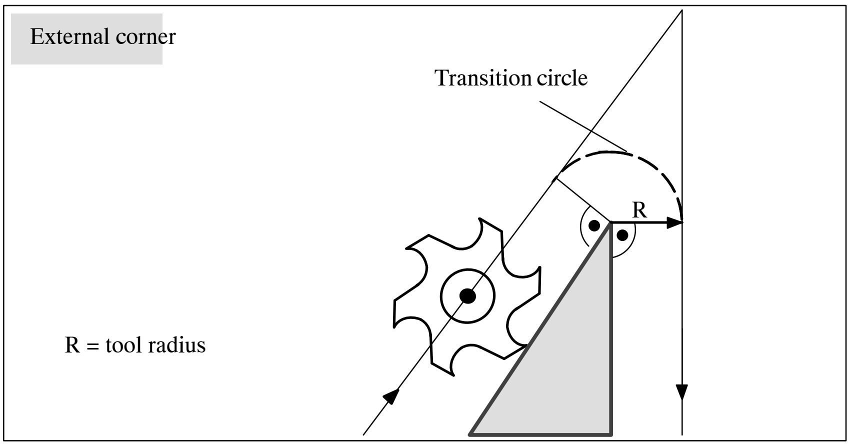


Fig. 8-51 Acute contour angle and switching to transition circle

8.6.6 Tool radius compensation OFF: G40

Functionality

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

The tool ends the **block before G40** in the normal position (compensation vector vertically to the tangent at the end point); irrespective of the approach angle.
If G40 is active, the reference point is the tool center point. The tool tip then travels to the programmed point upon deselection.

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

Programming

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

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

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

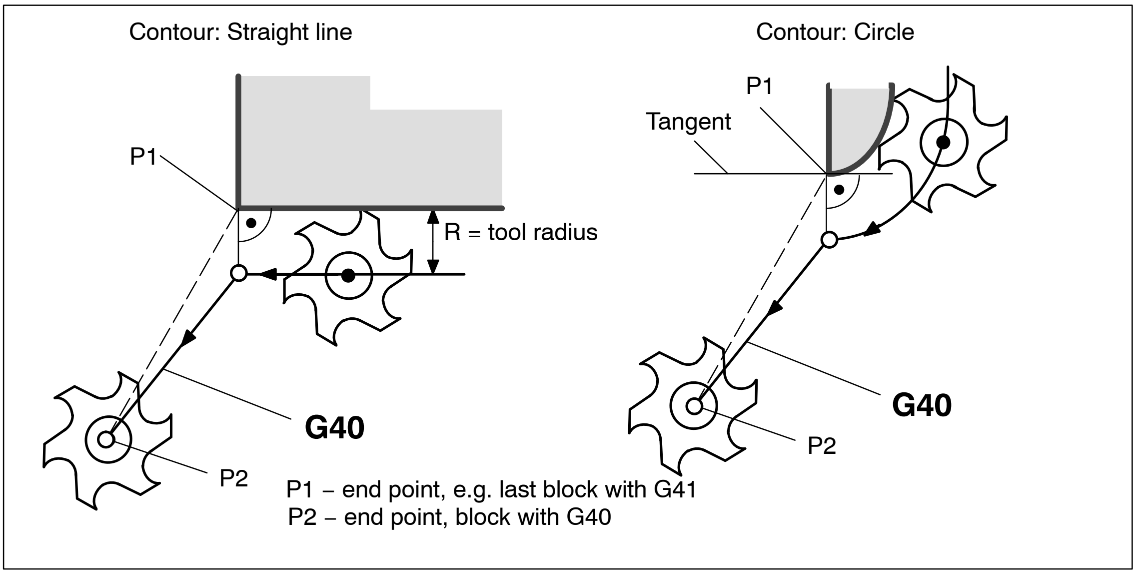


Fig. 8-52 Quitting the tool radius compensation

Programming example

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

8.6.7 Special cases of the tool radius compensation

Repetition of the compensation

The same compensation (e.g. G41 –> G41) can be programmed once more without writing G40 between these commands.
The last block in front of the new compensation call ends with the normal position of the compensation vector at the end point. The new compensation is carried out as a compensation start (behavior as described for change in compensation direction).

Changing the offset number

The offset number D can be changed in compensation mode. A modified tool radius is active with effect from the block in which the new D number is programmed. Its complete modification is only achieved at the end of the block. In other words: The modification is traversed continuously over the entire block; this also applies to circular interpolation.

Change of the compensation direction

The compensation direction G41 <–> G42 can be changed without writing G40.
The last block with the old compensation direction ends with the normal position of the compensation vector at the end point. The new compensation direction is executed as a compensation start (normal position at starting point).

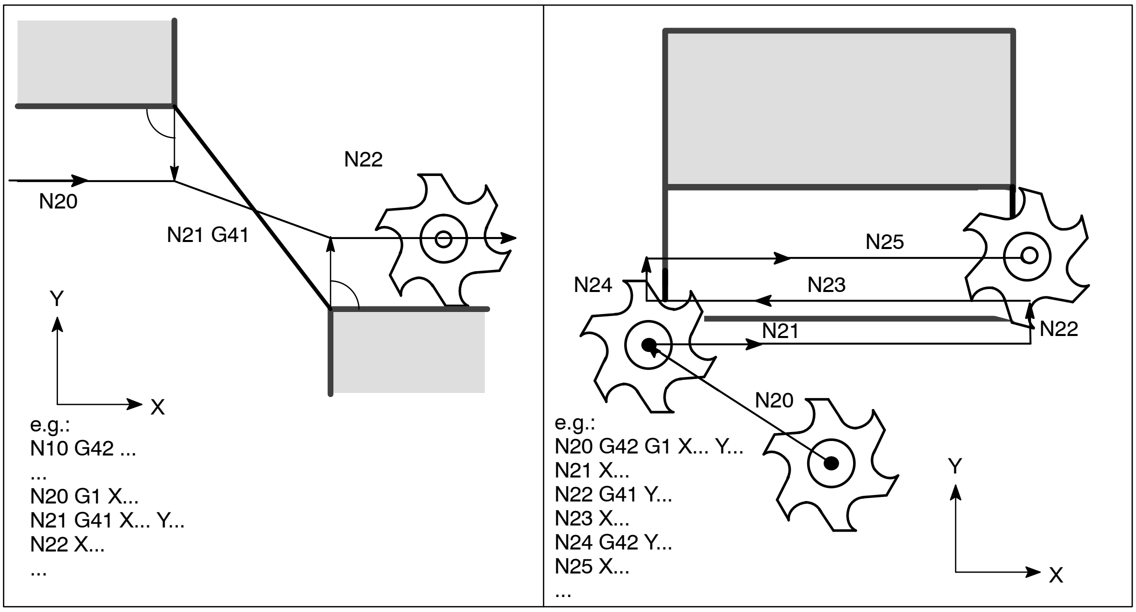


Fig. 8-53 Change of the compensation direction

Cancellation of compensation by M2

If the compensation mode is canceled using M2 (end of program) without writing the command G40, the last block with coordinates of the plane (G17 to G19) will end in the normal position of the compensation vector. **No** compensating movement is executed. The program ends with this tool position.

Critical machining cases

When programming, pay special attention to cases where the contour travel is smaller than the tool radius; in case of two successive internal corners, this is smaller than the diameter.

Such cases should be avoided.

Also check over multiple blocks that the contour contains no ”bottlenecks”.

When carrying out a test/dry run, use the largest tool radius you are offered.

Acute contour angles

If very sharp outside corners occur in the contour with active G451 intersection, the control system automatically switches to transition circle. This helps avoid long idle motions (see Fig. 8-51).

8.6.8 Example of tool radius compensation

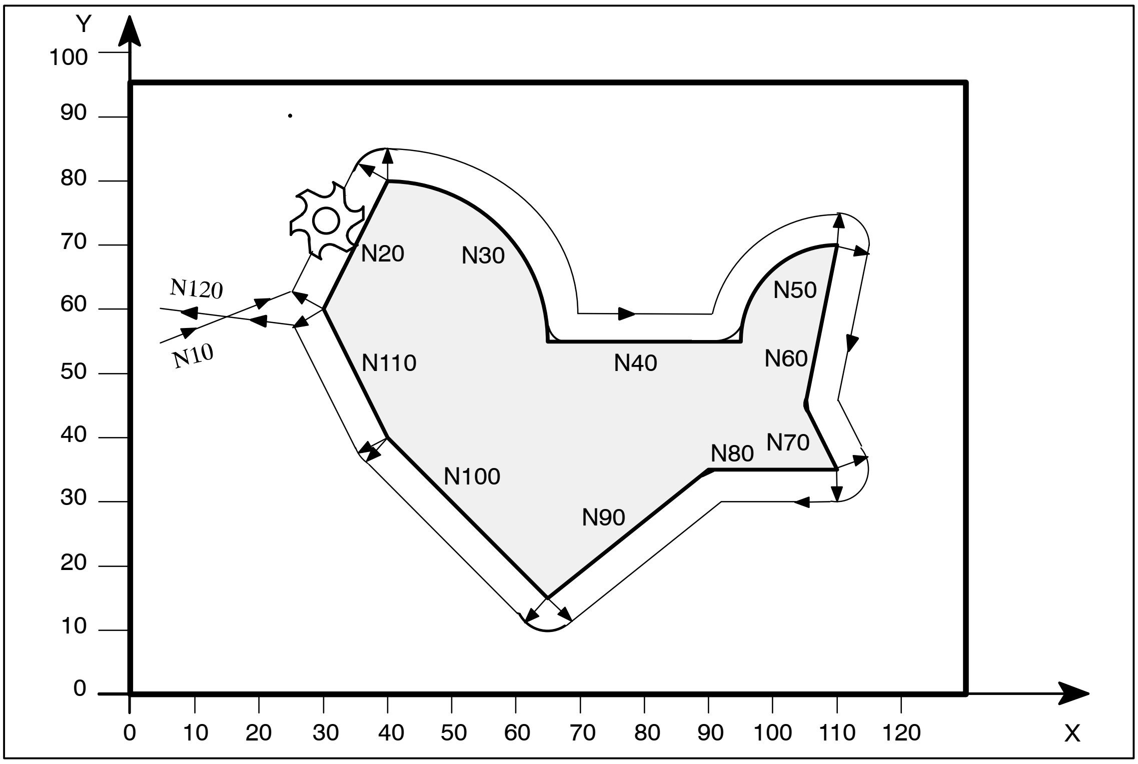


Fig. 8-54 Example of tool radius compensation

Programming example

```
N1 T1                        Tool 1 with compensation 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 the contour, transition circle
N20 X40 Y80
N30 G2 X65 Y55 I0 J–25
N40 G1 X95
N50 G2 X110 Y70 I15 J0
N60 G1 X105 Y45
N70 X110 Y35
N80 X90
N90 X65 Y15
N100 X40 Y40
N110 X30 Y60
N120 G40 X5 Y60              ;Quit compensation mode
N130 G0 Z50 M2
```

8.7 Miscellaneous function (M)

Functionality

The special function M initiates switching operations, such as ”Coolant ON/OFF” and other functions.

Permanent functions have already been assigned to some of the M functions by the control manufacturer. The machine manufacturer can freely dispose of the other functions.

Note:
An overview of the M special functions used and reserved in the control can be found in Section 8.1.6 ”Overview of the instructions”.

Programming

```
M...            ; a maximum of 5 M functions per block,
```

Activation

Activation in blocks with axis movements:
If the functions **M0, M1, M2** are contained in a block with traversing movements of the axes, then these M functions become effective **after the traversing movements.**

The functions **M3, M4, M5** are output to the internal PLC **prior to the traversing movements**. The axis movements only begin once the controlled spindle has ramped up for M3, M4. For M5, however, the spindle standstill is not waited for. The axis movements already begin before the spindle stops (default setting).

The remaining M functions are output to the PLC **with** the traversing movements.

If you would like to program an M function directly before or after an axis movement, then insert a separate block with this M function. **Please note**: This block interrupts G64 continuous path mode and generates exact stop!

Programming example

```
N10 S...
N20 X... M3                  ;M function in a block with axis motion
                              Spindle ramps up prior to the X axis motion
N180 M78 M67 M10 M12 M37     ;Max. 5 M functions in the block
```

Note

In addition to M and H functions, T, D, and S functions can also be transferred to the PLC (programmable logic controller). In all, a maximum of 10 such function outputs are possible in a block.

8.8 H function

Functionality

With H functions, floating point data (REAL data type – as with arithmetic parameters, see Section "Arithmetic Parameters R") can be transferred from the program to the PLC.
The meaning of the values for a given H function is defined by the machine manufacturer.

Programming

```
H0=...   to   H9999=...          ; max. 3 H functions per block
```

Programming example

```
N10   H1=1.987  H2=978.123  H3=4; 3 H functions in the block
N20 G0 X71.3 H99=–8978.234      ; with axis movements in the block
N30 H5                          ; corresponds to: H0=5.0
```

Note

In addition to M and H functions, T, D, and S functions can also be transferred to the PLC (programmable logic controller). In all, a maximum of 10 such function outputs are possible in a block.

8.9 Arithmetic parameters R, LUD and PLC variables

8.9.1 Arithmetic parameters R

Functionality

The arithmetic parameters are used if an NC program is not only to be valid for values assigned once, or if you must calculate values. The required values can be set or calculated by the control during program execution.
The arithmetic parameter values can also be set by operator inputs. If values have been assigned to the arithmetic parameters, they can be assigned to other variable–setting NC addresses in the program.

Programming

R0=... to R299=...

Value assignment

You can assign values in the following range to the arithmetic parameters:

±(0.000 0001 ... 9999 9999)
(8 decimal places and leading sign and decimal point).

The decimal point can be omitted for integer values. A plus sign can always be omitted.

Example:
R0=3.5678 R1=–37.3 R2=2 R3=–7 R4=–45678.123

Use the **exponential notation** to assign an extended range of numbers:

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

The exponent value is written after the **EX** characters; max. number of characters: 10 (including the leading 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

Remark: Several assignments can be provided per block; this also applies to the assignment of arithmetic expressions.

Assignments to other addresses

The flexibility of an NC program lies in assigning these arithmetic parameters or expressions with arithmetic parameters to other NC addresses. Values, arithmetic expressions and arithmetic parameters can be assigned to all addresses; **Exception: addresses N, G, and L**.

When assigning, write the ” = ” sign after the address character. It is also possible to have an assignment with a minus sign.

A separate block is required for assignments to axis addresses (traversing instructions).

Example:

```
N10 G0 X=R2     ;Assignment to the X axis
```

Arithmetic operations/arithmetic functions

When operators/arithmetic functions are used, it is necessary to use conventional mathematical notation. Machining priorities are set by parentheses. Otherwise, multiplication and division take precedence over addition and subtraction.
Degrees are used for the trigonometrical functions.
Permitted arithmetic functions: see Section ”Overview of the instructions”

Programming example: R parameters

```
N10 R1= R1+1                    ;The new R1 results from the old R1 plus 1
N20 R1=R2+R3   R4=R5–R6   R7=R8* R9   R10=R11/R12
N30 R13=SIN(25.3)               ;R13 yields the sine of 25.3 degrees
N40 R14=R1*R2+R3                ;Multiplication before addition R14=(R1*R2)+R3
N50 R14=R3+R2*R1                ;Result as block N40
N60 R15=SQRT(R1*R1+R2*R2)  ; Meaning:  R15 =
```

$$R15 = \sqrt{R1^2 + R2^2}$$

Programming example: Assignment to the axes

```
N10 G1 G91 X=R1 Z=R2 F300
N20 Z=R3
N30 X=–R4
N40 Z=–R5
...
```

8.9.2 Local User Data (LUD)

Functionality

The operator/programmer (user) can define his/her own variable in the program from various data types (LUD = Local User Data). These variables are only available in the program in which they were defined. The definition takes place immediately at the start of the program and can also be associated with a value assignment at the same time. Otherwise the starting value is zero.
The name of a variable can be defined by the programmer. The naming is subject to the following rules:

- maximum length 32 characters
- The first two characters must be letters. Only use letters, digits, or underscore.
- Do not use a name that is already being used in the controller (NC addresses, keywords, names of programs, subroutines, etc.)

Programming

```
DEF BOOL  varname1        ; Bool type, values: TRUE (=1), FALSE (=0)
DEF CHAR  varname2        ; Char type, 1 character in the ASCII code: ”a”, ”b”, ...
                          ; Code numerical value: 0 ... 255
DEF INT  varname3         ; Integer type, integer values, 32–bit value range:
                          ;  –2 147 483 648 to +2 147 483 648 (decimal)
DEF REAL  varname4        ; Real type, natural number  (such as arithmetic parameters R),
                          ; Range of values:  ±(0.000 0001 ... 9999 9999)
                          ; (8 decimal places, arithmetic sign and decimal point) or
                          ; exponential notation: ± ( 10^-300  ... 10^+300 ).
```

Each type requires a separate program line. It is, however, possible to define several variables of the same type in a line.

Example:

```
DEF INT PVAR1, PVAR2, PVAR3=12, PVAR4          ; 4 variables of the INTtype
```

Fields

In addition to the individual variables, one or two–dimensional fields of variables of these data types can also be defined:

```
DEF INT PVAR5[n]          ; One–dimensional field of the INT type, n: integer
DEF INT PVAR6[n,m]        ; Two–dimensional field of the INT type, n, m: integer
```

Example:

```
DEF INT PVAR7[3]          ; Field with 3 elements of the INTtype
```

Access to the individual field elements is granted in the program via the field index; each individual field element can be handled as an individual variable. The field index ranges from 0 to ”less number of elements”.

Example:

```
N10 PVAR7[2]=24           ; The third field element (with index 2) is assigned the value 24.
```

Value assignment for field with SET instruction:

N20 PVAR5[2]=SET(1,2,3) ; Beginning with the 3rd field element, different values are assigned.

Value assignment for field with REP instruction:

N20 PVAR7[4]=REP(2) ; As of field element [4], all elements is assigned the same value, here 2.

Number of LUDs

With the SINUMERIK 802D, max. 200 LUDs may be defined. However, please note: The standard cycles of SIEMENS also use LUDs and they share this quantity with the user. Always keep a sufficient reserve if you are working with these cycles.

Note for display

There is no special display for LUDs. They would only be visible during execution of the program anyway.
For testing purposes, when creating the program, the LUDs may be assigned to the arithmetic parameters R and are thus visible via the arithmetic parameter display, but are converted into the REAL type.
Another possibility of displaying is offered in the STOP condition of the program via a message output:
MSG(" VAR1 value: "<<PVAR1<<" VAR2 value: ": "<<PVAR2) ;Values of PVAR1, PVAR2
M0

8.9.3 Reading and writing PLC variables

Functionality

To allow rapid data exchange between NC and PLC, a special data area exists in the PLC user interface with a length of 512 bytes. In this area, PLC data are compatible in data type and position offset. In the NC program, these compatible PLC variables can be read or written.
Special system variables are provided here:

$A_DBB[n] ; Data byte (8-bit value)
$A_DBW[n] ; Data word (16-bit value)
$A_DBD[n] ; Data double-word (32-bit value)
$A_DBR[n] ; REAL data (32-bit value)
n stands here for the position offset (between the data area start and the variable start) in bytes

Example:
R1=$A_DBR[5] ; Reading of a REAL value, offset 5 (starts at byte 5 of this range)

Notes

- When reading variables, a preprocessing stop is generated (internal STOPRE).
- A maximum of 3 variables can be written simultaneously (in a block).

8.10 Program jumps

8.10.1 Jump destination for program jumps

Functionality

A **label** or a **block number** serve to mark blocks as jump destinations for program jumps. Program jumps can be used to branch to the program sequence.

Labels can be freely selected, but must contain a minimum of 2 and a maximum of 8 letters or numbers, and the **first two characters** must be **letters** or underscores.

Labels that are in the block that serves as the jump destination are ended by a **colon**. They are always at the start of a block. If a block number is also present, the label is located **after the block number**.

Labels must be unique within a program.

Programming example

```
N10 LABEL1: G1 X20      ; LABEL1 is the label, jump destination
...
TR789: G0 X10 Z20       ; TR789 is the label, jump destination
                          – no block number exists
N100 ...                ; A block number can be a jump destination
...
```

8.10.2 Unconditional program jumps

Functionality

NC programs process their blocks in the sequence in which they were arranged when they were written.

The processing sequence can be changed by introducing program jumps.

The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

The unconditional jump instruction requires a separate block.

Programming

GOTOF *Label* ;Jump forward (towards the last block of the program)
GOTOB *Label* ;Jump backward (towards the first block of the program)

Label ;Selected character string for the label (jump mark) or block number

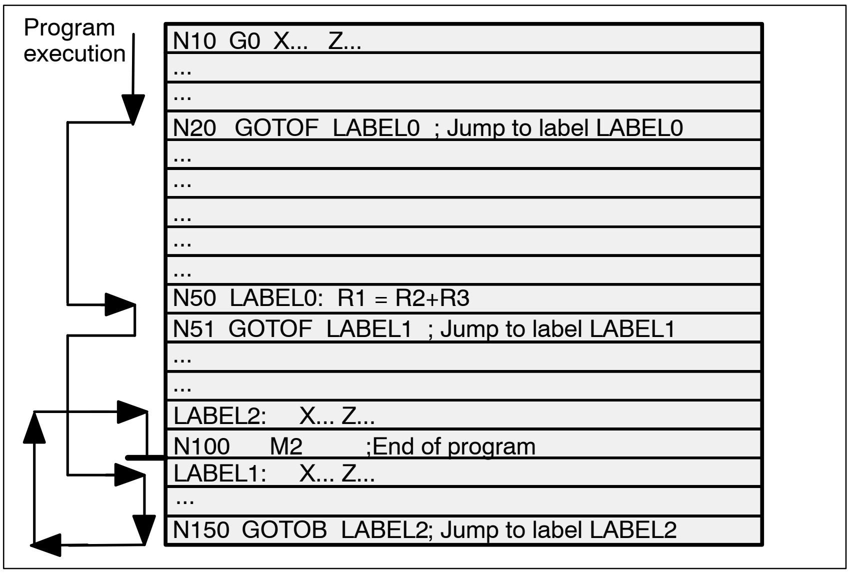


Fig. 8-55 Unconditional jumps using an example

8.10.3 Conditional program jumps

Functionality

Jump conditions are formulated after the **IF instruction**. If the jump condition (**value not zero**) is satisfied, the jump takes place.
The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

Conditional jump instructions require a separate block. Several conditional jump instructions can be located in the same block.

By using conditional program jumps, you can also considerably shorten the program, if necessary.

Programming

IF *condition* GOTOF *label* ;Jump forward
IF *condition* GOTOB *label* ;Jump backward

GOTOF ;Jump direction forward (in the direction of the last block of the program)
GOTOB ;Jump direction backward (in the direction of the first block of the program)
Label ;Selected string for the label (jump label) or block number
IF ;Starting of the jump condition
condition ;Arithmetic parameter, arithmetic expression for formulating the condition

Comparative operations

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

The comparative operations support the formulating of a jump condition. Arithmetic expressions can also be compared.

The result of comparing operations is ”satisfied” or ”not satisfied.” ”Not satisfied” sets the value equal to zero.

Programming example for comparing 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  LABEL1        ; If R1 is not zero, go to block with LABEL1
...
N90 LABEL1: ...
N100 IF R1>1 GOTOF  LABEL2  ; If R1 is greater than 1, go to the block with LABEL2
...
N150 LABEL2: ...
...
N800 LABEL3: ...
...
N1000 IF R45==R7+1 GOTOB  LABEL3; If R45 is equal to R7 plus 1, go to the
                                   block with LABEL3
...
Several conditional jumps in the block:
N10 MA1: ...
...
N20 IF R1==1 GOTOB  MA1  IF R1==2 GOTOF  MA2 ...
...
N50 MA2: ...
```

Remark: The jump is executed for the first fulfilled condition.

8.10.4 Program example for jumps

Task

Approaching points on a circular section:

Given:	Starting angle:	30°	in R1
	Circle radius:	32 mm	in R2
	Distance of the positions:	10°	in R3
	Number of points:	11	in R4
	position of the circle center point in Z:	50 mm	in R5
	Position of the circle center point in X:	20 mm	in R6

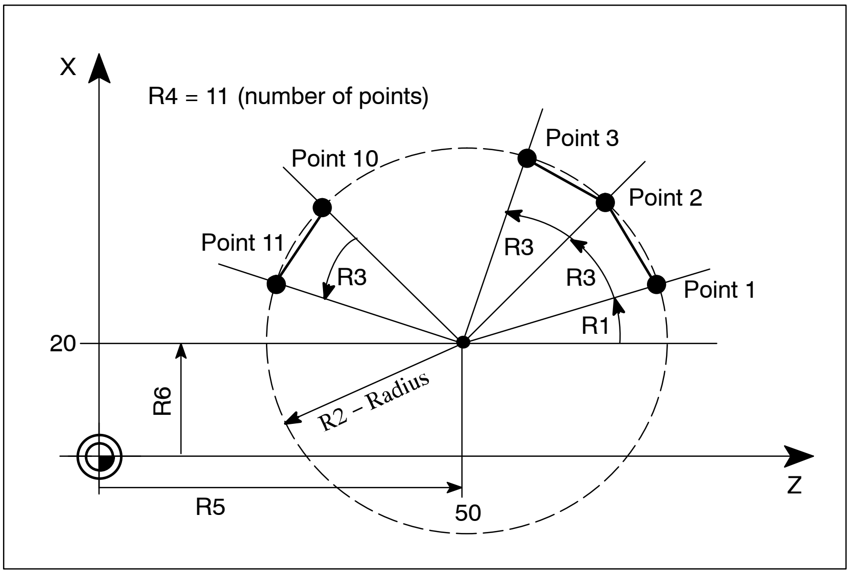


Fig. 8-56 Approaching points on a circular section

Programming example

```
N10 R1=30  R2=32  R3=10  R4=11  R5=50  R6=20       ; Assignment of the initial 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

In block N10, the starting conditions are assigned to the corresponding arithmetic parameters. The calculation of the coordinates in X and Z and the processing takes place in N20.

In block N30, R1 is incremented by the clearance angle R3; R4 is decremented by 1. If R4 > 0, N20 is executed again; otherwise, N50 with end of program.

8.11 Subroutine technique

8.11.1 General

Use

Basically, there is no difference between a main program and a subroutine.

Frequently recurring machining sequences are stored in subroutines, e.g certain contour shapes. These subroutines are called at the appropriate locations in the main program and then executed.

One form of subroutine is the **machining cycle**. Machining cycles contain universally valid machining scenarios (e.g.: Drilling, tapping, groove milling, etc.). By assigning values via included transfer parameters, you can adapt the subroutine to your specific application.

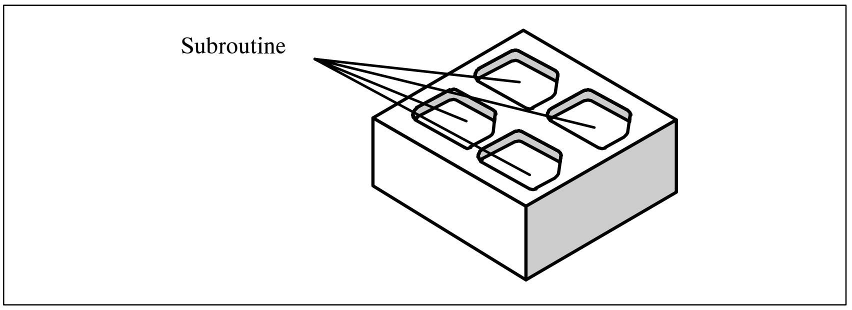


Fig. 8-57 Example for using a subroutine for a workpiece four times

Structure

The structure of a subroutine is identical to that of a main program (see Section 8.1.2 ”Program structure”). Like main programs, subroutines contain
M2 program end in the last block of the program sequence. This means a return to the program level where the subroutine was called from.

End of program

The end instruction **RET** can also be used instead of the M2 program end in the subroutine.

RET requires a separate block.

The RET instruction is used when G64 continuous–path mode is not to be interrupted by a return. With M2, G64 is interrupted and exact stop is initiated.

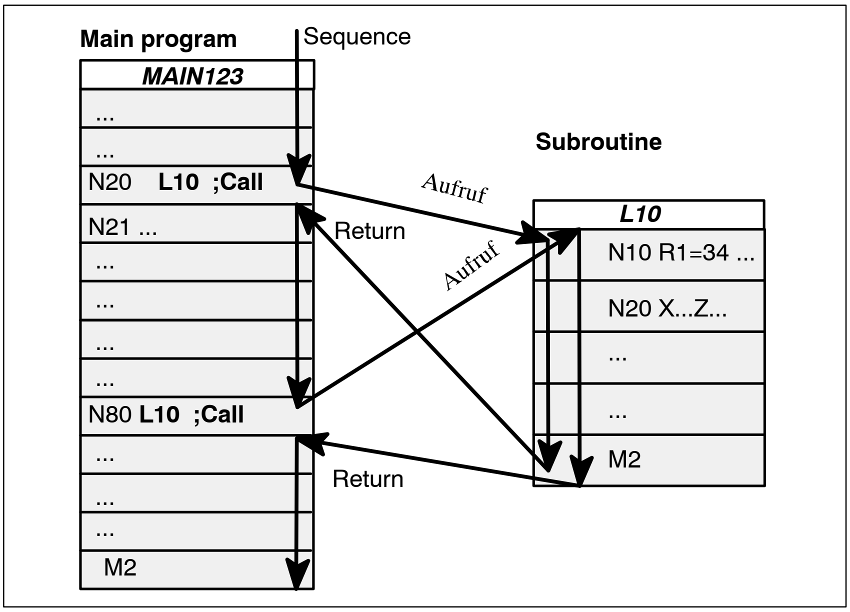


Fig. 8-58 Example of sequence when calling a subroutine twice

Subroutine name

The subprogram is given a unique name allowing it to be selected from among the others. When you create the program, the program name may be freely selected provided the following conventions are observed:

The same rules apply as for the names of main programs.

Example: **LRAHMEN7**

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

Note: With address L, leading zeros are meaningful for differentiation.

Example: **L128** is not **L0128** or **L00128** !
These are three different subroutines.

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

Subroutine call

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

Example:

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

Program repetition P...

If a subroutine is to be executed several times in succession, write the number of times it is to be executed in the block of the call after the subroutine name under the **address P**. A maximum of **9999 passes** are possible (P1 ... P9999).

Example:

N10 L785 P3 ; Call of the subroutine L785, 3 passes

Nesting depth

Subroutines can also be called from a subroutine, not only from a main program. Totally, **8 program levels**, including the main program level are available for such a nested call.

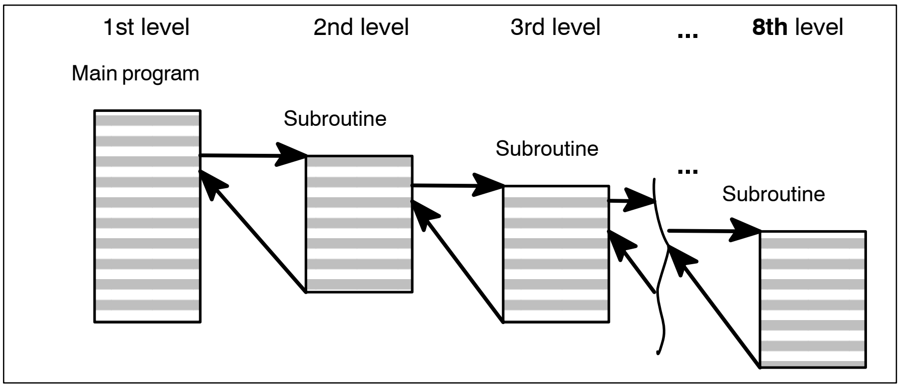


Fig. 8-59 Execution with 8 program levels

Information

Modal G functions can be changed in the subroutine, e.g. G90 –> G91. When returning to the calling program, ensure that all modal functions are set the way you need them to be.

This also applies to the arithmetic parameters R. Please make sure that the values of your arithmetic parameters used in upper program levels are not inadvertently changed in lower program levels.

When working with SIEMENS cycles, up to 4 program levels are needed.

8.11.2 Calling machining cycles

Functionality

Cycle are technology subroutines that realize a certain machining process, such as drilling or thread cutting, generally. Adaptation to the concrete problem is done directly via supply parameters/values when calling the respective cycle.

Programming example

```
N10  CYCLE83(110, 90, ...)        ; Call of cycle 83, transfer values directly, separate block
...
N40 RTP=100  RFP= 95.5 ...        ; Set transfer parameters for cycle 82
N50 CYCLE82(RTP, RFP, ...)        ; Call of cycle 82, separate block
```

8.11.3 Modal subroutine call

Functionality

The subroutine in the block containing MCALL is called automatically after each successive block containing a **path motion**. The call acts until the next MCALL is called.
The modal call of the subroutine which contains MCALL or quitting of the call requires a separate block.
MCALL is advantageous, for example, when producing drill patterns.

Programming example

Application example: Drilling a row of holes

```
N10 MCALL CYCLE82(...)   ; Drilling cycle 82
N20 HOLES1(...)          ; Cycle for the row of holes, CYCLE82 is executed with the trans-
fer parameters N30  MCALL each time after
                           approaching the hole position (...)
; Modal call of CYCLE82(...)  completed
```

8.12 Timers and workpiece counters

8.12.1 Runtime timer

Functionality

The timers are prepared as system variables ($A...) that can be used for monitoring the technological processes in the program or only in the display.
These timers can only be read. There are timers that are always active. Others can be deactivated via machine data.

Timers – always active

- Time since the last ”Control power–up with default values” (in minutes):
  $AN_SETUP_TIME (read–only)
  ”Power–up of the control system with default values” will reset it to zero automatically.
- Time since the last ”Power–up of the control system” (in minutes):
  $AN_POWERON_TIME (read–only)
  ”It is reset to zero automatically with each power–up of the control system.

Timers that can be deactivated

The following timers are activated via machine data (default setting).
The start is timer–specific. Each active run–time measurement is automatically interrupted in the stopped program state or for feedrate–override–zero.
The behavior of the activated timers for active dry run feedrate and program testing can be specified using machine data.

- Total runtime of NC programs in the AUTOMATIC mode(in seconds)
  $AC_OPERATING_TIME
  In the AUTOMATIC mode, the runtimes of all programs between NC START and end of program / RESET are summed up. The timer is zeroed with each power–up of the control system.
- Runtime of the selected NC program (in seconds):
  $AC_CYCLE_TIME
  The runtime between NC START and end of program / RESET is measured in the selected NC program. The timer is reset with the start of a new NC program.
- Tool action time (in seconds):
  $AC_CUTTING_TIME
  The runtime of the path axes is measured in all NC programs between NC START and end of program / RESET without rapid traverse active and with the tool active.
  The measurement is also interrupted if a dwell time is active.
  The timer is automatically reset to zero with each ”Control power–up with default values”.

Programming example

```
N10 IF $AC_CUTTING_TIME>=R10 GOTOF WZZEIT      ;Tool action time limit value?
...
N80 WZZEIT:
N90 MSG(”Tool action time: Limit value reached”)
N100 M0
```

Display

The contents of the active system variables is displayed on the screen in the ”OFFSET/PARAM” operating area –> ”Setting data” softkey (2nd page):

Run time	= $AC_OPERATING_TIME
Cycle time	= $AC_CYCLE_TIME
Cutting time	= $AC_CUTTING_TIME
Setup time	= $AN_SETUP_TIME
Power on time	= $AN_POWERON_TIME

”Cycle time” is also displayed in the AUTOMATIC mode in the ”Position” operating area in the information line.

8.12.2 Workpiece counter

Functionality

The ”Workpiece counter” function provides counters for counting workpieces.
These counters exist as system variables with write and read access from the program or via operator input (observe the protection level for writing!).
Machine data can be used to control counter activation, counter reset timing and the counting algorithm.

Counter

- Number of workpieces required (required number of workpieces):
  $AC_REQUIRED_PARTS
  The number of workpieces at which the number of current workpieces $AC_ACTUAL_PARTS is set to zero can be defined in this counter.
  The generation of the display alarm 21800 ”Workpiece setpoint reached” can be activated via machine data.
- Number of workpieces produced in total (actual total):
  $AC_TOTAL_PARTS
  The counter specifies the number of all workpieces manufactured from the time of starting.

  The counter is automatically set to zero upon every booting of the control system.
- Number of current workpieces (actual total):
  $AC_ACTUAL_PARTS
  This counter registers the number of all workpieces produced since the starting time.
  When the workpiece setpoint is reached ($AC_REQUIRED_PARTS, value greater than zero), the counter is automatically zeroed.
- Number of workpieces specified by the user:
  $AC_SPECIAL_PARTS
  This counter allows user–defined workpiece counting. Alarm output can be defined for the case of identity with
  $AC_REQUIRED_PARTS (workpiece target). The user must reset the counter himself.

Programming example

```
N10 IF $AC_TOTAL_PARTS==R15 GOTOF SIST            ;Count reached?
...
N80 SIST:
N90 MSG("Workpiece setpoint reached")
N100 M0
```

Display

The contents of the active system variables is displayed on the screen in the "OFFSET/PARAM" operating area –> "Setting data" softkey (2nd page):

Part total = $AC_TOTAL_PARTS
Part required = $AC_REQUIRED_PARTS
Part count = $AC_ACTUAL_PARTS
$AC_SPECIAL_PARTS (not displayed)

"Part count" is also displayed in the AUTOMATIC mode in the "Position" operating area in the information line.

8.13 Language commands for tool monitoring

8.13.1 Tool monitoring overview

Functionality

This function is an option and available as of SW 2.0.

The tool monitoring is activated via machine data.
The following types of active cutting edge monitoring for the active tool are possible:

- Monitoring of the **tool life**
- Monitoring of the **workpiece count**

For a tool, the above–mentioned types of monitoring can be activated simultaneously.

The control / data input of tool monitoring is preferably done by operator input. In addition, functions are programmable.

Monitoring counter

Monitoring counters exist for each monitoring type. The monitoring counters count from a set value > 0 down to zero. When a counter has decremented to a value of <=0, the limit value is reached. A corresponding alarm message is issued.

System variable for type and condition of the monitoring

- $TC_TP8[t] – status of the tool with number t:
  Bit 0 =1: Tool is active
  =0: Tool not active
  Bit 1 =1: Tool is enabled
  =0: not enabled
  Bit 2 =1: Tool is disabled
  =0: not disabled
  Bit 3 : Reserved
  Bit 4 =1: Prewarning limit reached
  =0: Not reached
- $TC_TP9[t] – type of the monitoring function for the tool with number t :
  = 0: No monitoring
  = 1: (Tool monitored for the tool life
  = 2: Count–monitored tool

These system variables can be read/written in the NC program.

System variables for tool monitoring data

Table 8-3 Tool monitoring data

Identifier	Description	Data type	Default setting
$TC_MOP1[t,d]	Prewarning limit for tool life in minutes	REAL	0.0
$TC_MOP2[t,d]	Residual tool life in minutes	REAL	0.0
$TC_MOP3[t,d]	Count prewarning limit	INT	0

Table 8-3 Tool monitoring data, cont'd

Identifier	Description	Data type	Default setting
$TC_MOP4[t,d]	Remaining part quantity	INT	0
...	...		
$TC_MOP11[t,d]	Tool life setpoint	REAL	0.0
$TC_MOP13[t,d]	Required count	INT	0

t for tool number T, d for D number

System variables for active tool

The following can be read in the NC program via system variables:

- $P_TOOLNO – number of the active tool T
- $P_TOOL – active D number of the active tool

8.13.2 Tool life monitoring

Tool life monitoring is done for the tool cutting edge that is currently in use (active cutting edge D of the active tool T).
As soon as the path axes traverse (G1, G2, G3, ... but not for G0), the residual tool life ($TC_MOP2[t,d]) of this tool cutting edge is updated. If the residual tool life of a tool's cutting edge runs below the value of "Prewarning limit for tool life" ($TC_MOP1[t,d]), it is reported via an interface signal " to the PLC.
If the residual tool life <=0 , an alarm is issued and an additional interface signal is set. The tool changes to the "disabled" condition and cannot be programmed again until this condition changes. The operator must intervene: The operator must change the tool or ensure that he has an operational tool for machining.

$A_MONIFACT system variable

The **$A_MONIFACT** system variable (REAL data type) allows the monitoring clock to be run slower or faster. This factor can be set before using the tool, in order to take the different kinds of wear into consideration according to the workpiece material used, for example.

After booting of the control, reset / program end, the factor $A_MONIFACT has the value 1.0. Real time is in effect.
Examples for accounting:
$A_MONIFACT=1 1 minute realtime = 1 minute tool life which is decremented
$A_MONIFACT=0.1 1
minute realtime = 0.1 minute tool life which is decremented
$A_MONIFACT=5 1 minute real time = 5 minutes tool life which are decremented

Setpoint update with RESETMON( )

The RESETMON(state, t, d, mon) function sets the actual value to the set-point:

– either for all cutting edges or only for a certain cutting edge of a certain tool
– either for all monitoring type or only for a certain monitoring type.

Transfer parameters:

INT state Status of executing the command :
- = 0 Command executed successfully
- = −1 The edge with the specified D number d does not exist.
- = −2 The tool with the specified T number t does not exist.
- = −3 The specified tool t does not have a defined monitoring function.
- = −4 The monitoring function is not activated, i.e. the command is not executed.

INT t Internal T number :
- = 0 For all tools
- <> 0 For this tool (t < 0 : Absolute value formation |t|)

INT d *optional:* D number of the tool with the number t:
- > 0 for this D number
- without d / = 0 all cutting edges of tool t

INT mon *optional:* bit−coded parameter for the monitoring type (values analogously to $TC_TP9):
- = 1: Tool life
- = 2: Count
- without monitoring or = 0:

All actual values of the monitoring functions active for tool t are set to the setpoints.

Notes:

- RESETMON( ) has no effect during active ”Program test.”
- The variable for the **state** status feedback must be defined at the beginning of the program using a DEF instruction: DEF INT state
  You can also define a different name for the variable (instead of state, with a maximum of 15 characters, beginning with 2 letters). The variable is only available in the program in which it was defined.
  The same applies to the monitoring type variable **mon.** If an entry is required for this, it may also be passed directly as a number (1 or 2).

8.13.3 Workpiece count monitoring

The workpiece count of the active cutting edge of the active tool is monitored.
Monitoring of the unit quantity covers all tool cutting edges, which are used to manufacture a workpiece. If the count is changed by new parameters, the monitoring data are adapted to all of the tool cutting edges that became active since the last unit count.

Updating the workpiece count by operator input or SETPIECE( )

The workpiece count can be updated by an operator input (HMI) or in the NC program through the SETPIECE( ) language command.

By using the **SETPIECE** function, the programmer can update the count monitoring data of the tools involved in the machining process. All tools that became active since the last activation of SETPIECE are acquired with their D numbers. If a tool is active at the time when SETPIECE( ) is called, it is also counted.
If a block containing path axis motions is programmed after SETPIECE( ), the appropriate tool is also taken into account in the next SETPIECE call.

SETPIECE(x) ;
x : = 1... 32000 Number of workpieces produced since the last execution of the SETPIECE function. The counter status for the remaining part quantity ($TC_MOP4[t,d]) is reduced by this value.
x : = 0 Deletion of all counters for the residual count ($TC_MOP4[t,d]) for the tools/D number involved in machining since then.
Alternatively, the deletion via operation is recommended (HMI).

Programming example

```
N10 G0 X100
N20 ...
N30 T1
N40 M6
N50 D1
N60 SETPIECE(2)        ;$TC_MOP4[1,1 ] (T1,D2) is decremented by "2"

N70 T2
N80 M6
N90 SETPIECE(0)        ;Deletion command for the tools stored
N91 D2
N100 SETPIECE(1)       ;$TC_MOP4[2,2 ] (T2,D2) is decremented by "1"

N110 SETPIECE(0)       ;Deletion command for the tools stored
N120 M30
```

Notes:

- The SETPIECE( ) command is not active in the block search.
- Direct writing of $TC_MOP4[t,d] is recommended only in simple cases. A subsequent block with the STOPRE command is required.

Setpoint update

The setpoint update, i.e. setting the remaining workpiece counters ($TC_MOP4[t,d]) to the workpiece count setpoint ($TC_MOP13[t,d]) is typically performed via operator input (HMI). It can, however, also be performed through the RESETMON (state, t, d, ,mon) function as already described for the service life monitoring.

Example:

```
DEF INT state    ; Defining a variable for the status feedback in the beginning of the program
...
N100 RESETMON(state,12,1,2)   ;Updating the setpoint of the workpiece counter for T12, D1
...
```

Programming example

```
DEF INT state                              ; Define variable for the status feedback from RE-
SETMON
                                            ()
;
G0 X...                                    ; Retraction
T7                                         ; Load new tool, possibly with M6
$TC_MOP3[$P_TOOLNO,$P_TOOL]=100            ; Prewarning limit 100 pcs.
$TC_MOP4[$P_TOOLNO,$P_TOOL]=700            ; Residual count
$TC_MOP13[$P_TOOLNO,$P_TOOL]=700           ; Setpoint of count
; ; Activation after setting:
$TC_TP9[$P_TOOLNO,$P_TOOL]=2               ; Activation of count monitoring, active tool
STOPRE
ANF:
BEARBEIT                                   ; Subroutine for workpiece machining
SETPIECE(1)                                ; Update counter
M0                                         ; Next tool; press NC START to continue
IF ($TC_MOP4[$P_TOOLNO,$P_TOOL]]>1) GOTOB ANF
MSG(”Tool T7 worn – please change”)
M0                                         ; after changing the tool, press NC START to con-
tinue
RESETMON(state,7,1,2)                      ;Workpiece counter setpoint update
IF (state<>0) GOTOF ALARM
GOTOB ANF
ALARM: ; display errors:
MSG(”Error RESETMON: ” <<state)
M0
M2
```

8.14 Smooth approach and retraction

Functionality

This function is available with SW 2.0 and higher.

The function ”Smooth approach and retraction” (SPR) is intended to approach the beginning of a contour tangentially (”smooth”), to a large degree independently of the position of the starting point. The control system will calculate the intermediate points and will generate the required traversing blocks. This function is used preferably in conjunction with the tool radius compensation (TRC). The commands G41, G42 determine the approach / retraction direction to the left or right of the contour (see also Section 8.6.4 ”Selecting the tool radius compensation: G41, G42)”.

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

Programming

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

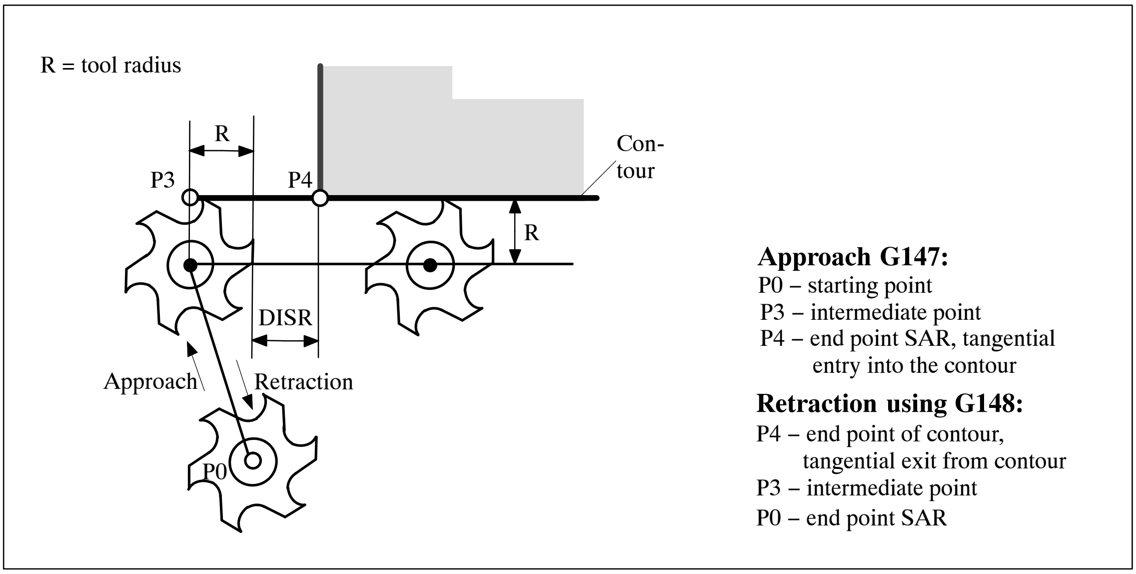


Fig. 8-60 Approach along a straight line using the example of G42 or retraction using G41 and completion with G40

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

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

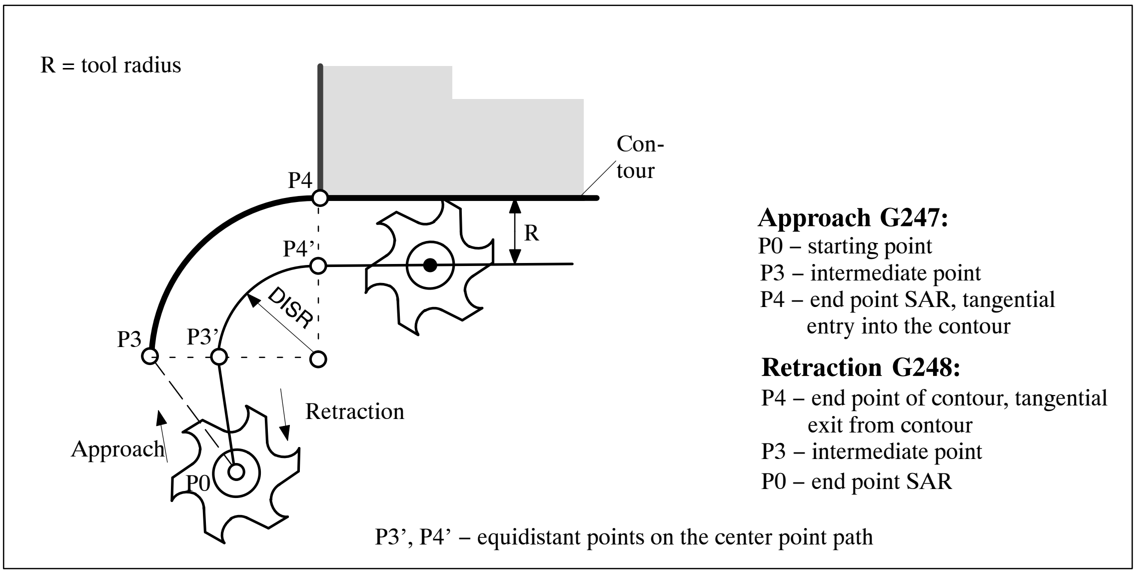


Fig. 8-61 Approach along a quadrant using the example of G42 or retraction using G41 and completion with G40

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

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

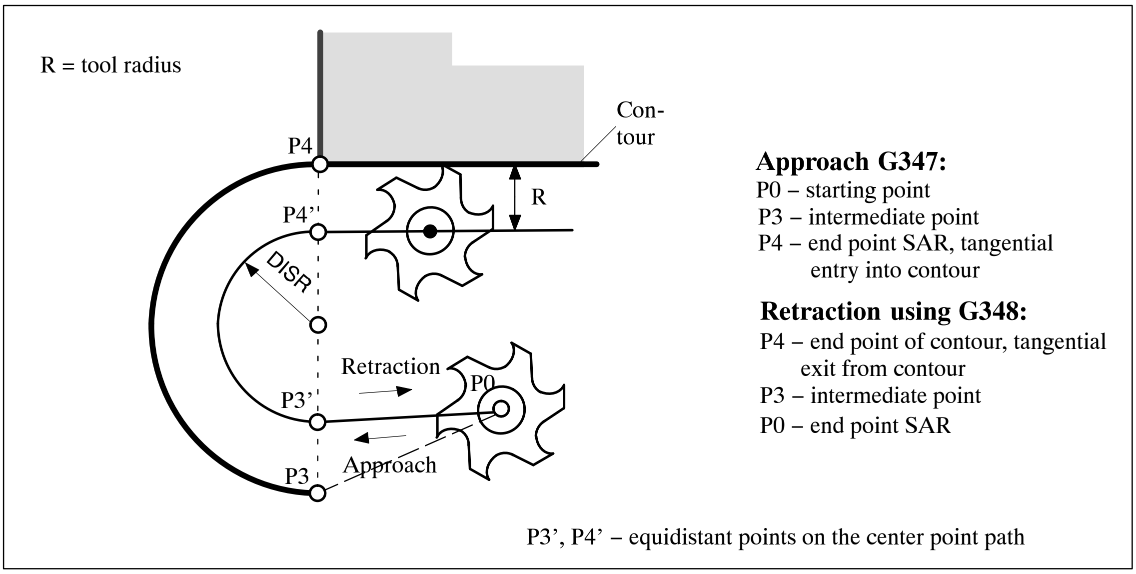


Fig. 8-62 Approach along a semicircle using the example of G42 or retraction using G41 and completion with G40

Note

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

Controlling the infeed motion using DISCL and G340, G341

DISCL=... specifies the distance of point P2 from the machining plane (see Fig. 8-63).

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

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

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

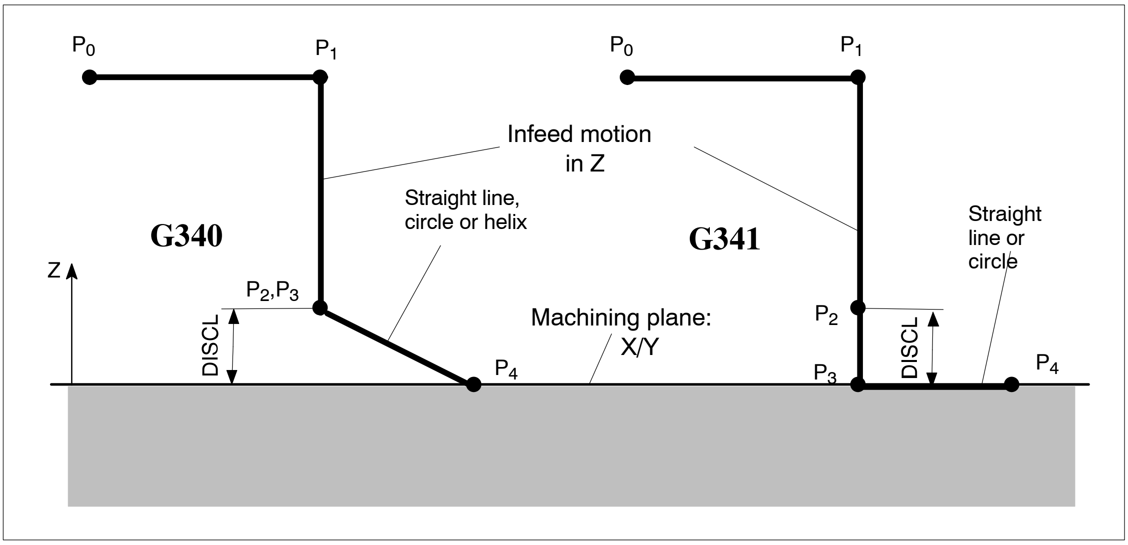


Fig. 8-63 Sequence of the approach motion dependent on G340 / G341 (example with G17)

Programming example: Approach along a semicircle with infeed

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

alternatively N30 / N40:

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

or

```
N30 G41 G347 G340 DISCL=3 DISR=13 F500
N40 G1 X40 Y–10 Z0
```

Explanation with regard to N30 / N40:
Using G0 (from N20), the point P1 (starting point of the semicircle, corrected by the tool radius) is approached in the plane Z=30, then lowering to the depth (P2, P3) with Z=3 (DISCL). The contour is reached at point X40 Y–10 in the depth Z=0 (P4) along a helix curve at a feedrate of 500 mm/min.

Approach and retraction velocities

- Velocity of the previous block (e.g.: G0):
  This is the velocity at which all motions from P0 to P2 are carried out, i.e. the motion which runs parallel to the machining plane and constitutes a part of the infeed motion until the safety clearance DISCL is provided.
- Programmed feedrate F:
  This feedrate value acts from P3 or P2 if FAD is not programmed. If no F word is programmed in the SAR block, the velocity of the previous block will act.

- Programming using FAD:
  Specification of the feedrate velocity with
  – G341: Infeed motion vertically to the machining plane from P2 to P3
  – G340: From point P2 or P3 to P4
  If FAD is not programmed, this part of the contour will also be traversed using the modally effective velocity of the previous block if no F word is programmed in the SAR block.
- **During retraction,** the roles of the modally effective feedrate from the previous block and the feedrate programmed in the SAR block are changed, i.e. the actual retraction contour is traversed using the old feedrate, and a new velocity programmed using the F word will apply correspondingly from P2 to P0.

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

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

Explanation with regard to N30:
Using G0 (from N20), the point P1 (starting point of the quarter, corrected by the tool radius) is approached in the plane Z=30, then lowering to the depth (P2, P3) with Z=5 (DISCL). Using a feedrate of FAD=500 mm/min, it is lowered to a depth of Z=0 (P3) (G341). Then, the contour is approached at point X40,Y−10 along a quarter in the plane (P4) using F=800 mm/min.

Intermediate blocks

A maximum of 5 blocks **without** moving the geometry axes can be inserted between an SAR block and the next traversing block.

Information

Programming when retracting:

- With an SAR block with a geometry axis programmed, the contour ends at P2. The positions on the axes that constitute the machining plane result from the retraction contour. The axis component standing vertically on it is defined by DISCL. With DISCL=0, the motion will run completely in the plane.
- If in the SAR block only the axis is programmed vertically to the machining plane, the contour will end at P1. The positions of the remaining axes will result, as described above. If the SAR block is also the TRC disable block, an additional path from P1 to P0 is inserted such that no motion results at the end of the contour when disabling the TRC.
- If only one axis is programmed, the 2nd missing axis is added modally by its last position from the previous block.

8.15 Milling of the peripheral surface – TRACYL

With SINUMERIK 802D, this function is an option and available with software release 2.0 and higher.

Functionality

- The kinematic transformation function TRACYL is used for milling machining of the peripheral surface of cylindrical objects and allows the production of slots at any position.
- The path of the slots is programmed in the **level** peripheral surface, which was logically developed for a specific machining cylinder diameter.

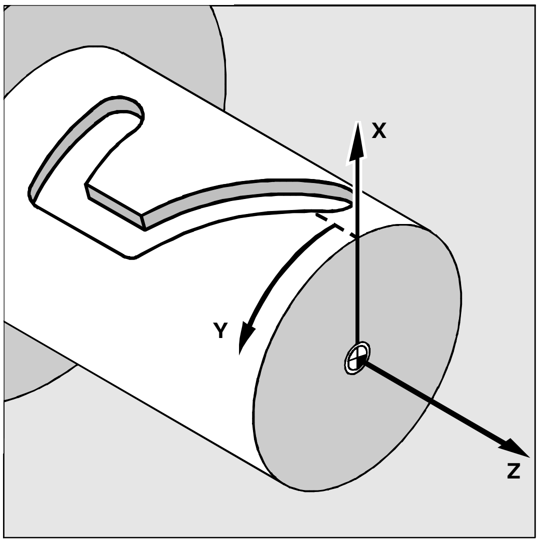


Fig. 8-64 Cartesian coordinate system X, Y, Z when programming TRACYL

- The control system transforms the programmed traversing motions in the Cartesian coordinate system X, Y, Z into motions of the real machine axes. A rotary axis (rotary table) is required.
- TRACYL must be configured using special machine data. The rotary axis position at which the value Y=0 is also defined here.
- Milling machines possess a real machine Y axis (YM). It is possible to configure an extended TRACYL variant for them. This allows slots with slot side offset to be produced: The slot side and base are perpendicular to each other – even if the milling tool's diameter is smaller than the slot width. This is otherwise only possible with exact fitting milling cutters.

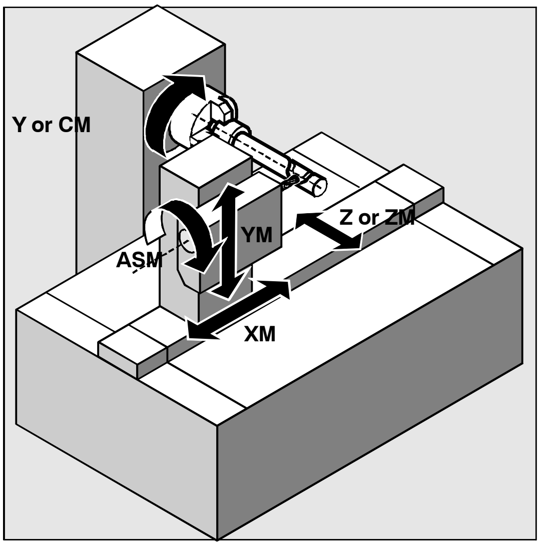


Fig. 8-65 Machine kinematics with a machine Y axis (YM)

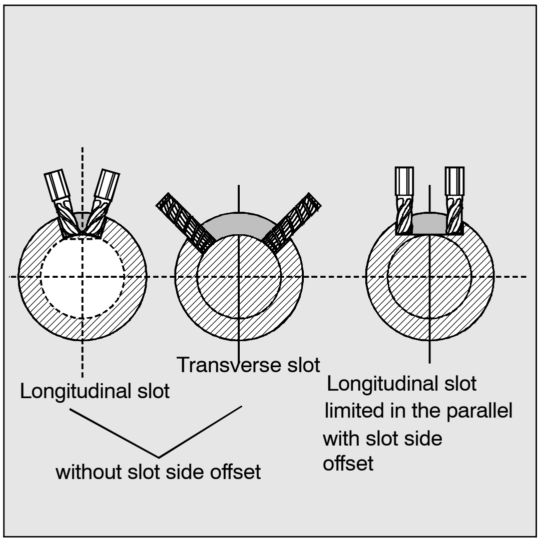


Fig. 8-66 Various slots (cross section)

Programming

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

TRAFOOF deactivates any active transformation function.

OFFN address

Distance from the slot side to the programmed path
The slot center line is generally programmed. OFFN defines the (half) slot width for activated cutter radius compensation (G41, G42).
Programming: OFFN=... ; Distance in mm

Note:
Set OFFN=0 once the slot has been completed. OFFN is also used outside of TRACYL – for offset programming in combination with G41, G42.

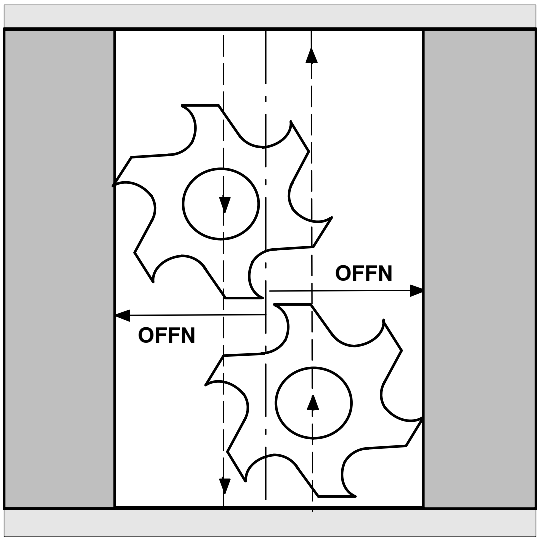


Fig. 8-67 Use of OFFN for the slot width

Programming notes

In order to mill with TRACYL, the slot center line is programmed in the part program with the coordinates and the (half) slot width is programmed with OFFN.
OFFN does not go into effect until tool radius compensation is selected. Furthermore, it must be guaranteed that OFFN >= tool radius to avoid that the opposite slot side is damaged.
A part program for milling a slot generally consists of the following steps:
1. Select a tool Select TRACYL Select a suitable work offset
4. Positioning
5. Program OFFN
6. Select the tool radius compensation
7. Approach block (position TRC and approach slot side)
8. Program the slot path via slot center line
9. Deselect the tool radius compensation
10. Retraction block (retract TRC from slot side)
11. Positioning
12. Clear OFFN
13.TRAFOOF (deselect TRACYL)
14. Reselect the original work offset
(see also following programming example)

Information

- Guiding slots:
  By using a tool diameter that corresponds exactly to the slot width, it is possible to produce an exact slot. The tool radius compensation is not activated for this.
  With TRACYL, slots can also be produced, whose tool diameter is smaller than the slot width. For this, the tool radius compensation (G41, GG42) and OFFN are used.
  To avoid problems of accuracy, the tool diameter should only be slightly smaller than the slot width.
- When working with TRACYL with slot side correction, the axis used for the correction (YM) should stand on the turning center of the rotary axis. Thus, the slot is created centered on the programmed slot center line.
- Selecting the tool radius compensation (TRC) :
  The TRC is in effect for the programmed slot center line. The slot side results. G42 is input so that the tool traverses to the left of the slot side (to the right of the slot center line). Accordingly, G41 is to be written to the right of the slot side (to the left of the slot center line).
  As an alternative to exchanging G41<–>G42, you can input the slot width with a minus sign in OFFN.
- Since, even without TRACYL, OFFN is included when TRC is active, OFFN should be reset to zero after TRAFOOF. OFFN acts differently with TRACYL than it does without TRACYL.
- It is possible to change OFFN within a part program. This allows the actual slot center line to be offset from the center.

References: Description of Functions, Section ”Kinematic transformations”

Programming example

Making a hook–shaped slot

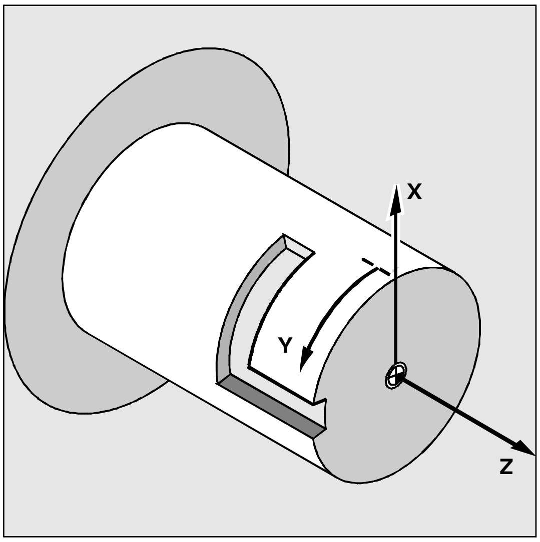


Fig. 8-68 Producing a slot (example)

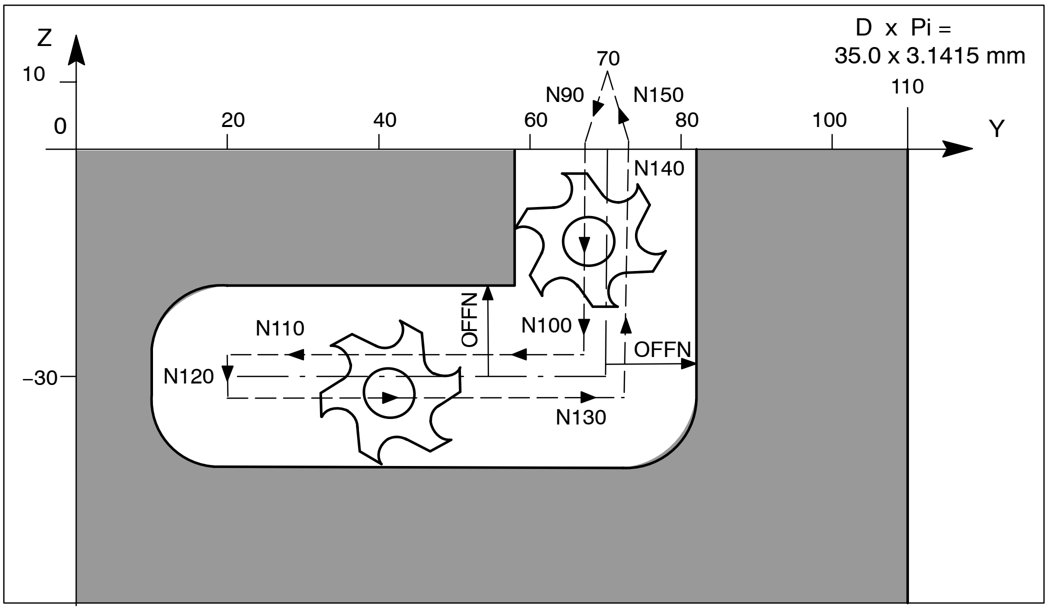


Fig. 8-69 Programming the slot, values at the slot base

; Machining diameter of the cylinder at the slot base: 35.0 mm
; Desired total slot width: 24.8 mm, the cutter in use has a radius of: 10,123 mm

```
N10 T1 F400 G94 G54             ; Milling tool, feedrate, feedrate type, zero offset
N15 G153 Y60                    ; Approach Y to turning center of C axis
N30 G0 X25 Z50 C120             ; Approach the starting position
N40 TRACYL (35.0)               ; Activate TRACYL, machining diameter 35.0 mm
N50 G55 G19                     ; Work offset, plane selection: Y/Z plane
N60 S800 M3                     ; Turn on spindle
N70 G0 Y70 Z10                            ; Starting position Y / Z,
                                ; Y is now geometry axis of the peripheral surface
N80 G1 X17.5                    ; Infeed cutter to slot base
N70 OFFN=12.4                             ; 12.4 slot side distance to slot center line
N90 G1 Y70 Z1 G42               ; Activate TRC, approach slot side
N100 Z−30                       ; Slot section parallel to cylinder axis Slot section parallel to
circumference
N120  G42 G1 Y20  Z−30          ; Restart TRC, approach other slot side,
                                ; slot distance continues to be 12.4 mm to slot center line
N130 Y70 F600                   ; Slot section parallel to the circumference
N140 Z1                         ; Slot section parallel to cylinder axis
N150 Y70 Z10 G40                ; Deactivate TRC
N160 G0 X25                     ; Retract cutter
N170 M5  OFFN=0                 ; Turn off spindle, delete slot side distance
N180 TRAFOOF                              ; Disable TRACYL
N200 G54 G17 G0 X25 Z50 C120              ; Approach starting position
N210 M2
```

8.16 G functions equivalent to the SINUMERIK 802S/C – Milling

SINUMERIK 802S/C	SINUMERIK 802D
G5	CIP
G158	TRANS
G258	ROT
G259	AROT
G900	CFTCP
G901	CFC

All the other G functions are the same as with 802S/C and 802D provided that they exist there.

Cycles

9

9.1 Overview of cycles

Cycles are generally applicable technology subroutines that can be used to carry out a specific machining process, such as drilling of a thread (tapping) or milling of a pocket. These cycles are adapted to individual tasks by parameter assignment.

The cycles described here are the same as supplied for the SINUMERIK 840D/810D.

Drilling cycle, drilling pattern cycles and milling cycles

The following standard cycles can be carried out using the SINUMERIK 802D control system:

- Drilling cycles

CYCLE81	Drilling, centering
CYCLE82	Drilling, counterboring
CYCLE83	Deep hole drilling
CYCLE84	Rigid tapping
CYCLE840	Tapping with compensating chuck
CYCLE85	Reaming 1 (boring 1)
CYCLE86	Boring (boring 2)
CYCLE87	Drilling with stop 1 (boring 3)
CYCLE88	Drilling with stop 2 (boring 4)
CYCLE89	Reaming 2 (boring 5)

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

- Drilling pattern cycles

HOLES1	Row of holes
HOLES2	Circle of holes

- Milling cycles

CYCLE71	Face Milling
CYCLE72	Contour milling
CYCLE76	Rectangular spigot milling
CYCLE77	Circular spigot milling

LONGHOLE	Slot
SLOT1	Groove milling pattern on a circle
SLOT2	Circumferential groove milling pattern
POCKET3	Milling a rectangular pocket (with any milling tool)
POCKET4	Milling a circular pocket (with any milling tool)
CYCLE90	Thread milling

The cycles are supplied with the tool box. They are loaded via the RS232 interface into the part program memory during the start–up of the control system.

Auxiliary cycle subroutines

The cycle package includes the following auxiliary subroutines:

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

These must always be loaded in the control.

9.2 Programming cycles

Call and return conditions

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

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

- 1st axis of the plane (abscissa)
- 2nd axis of the plane (ordinate)
- Drilling axis/infeed axis, 3rd axis standing vertically to the plane (applicate).

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

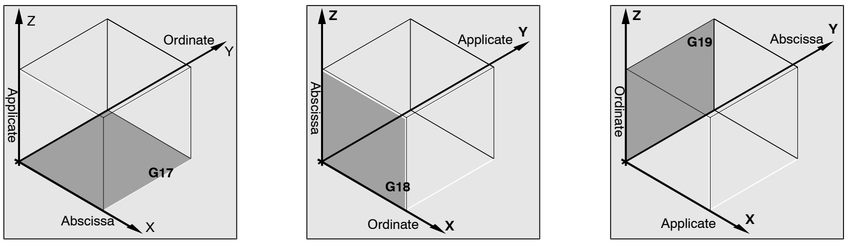


Fig. 9-1 Plane and axis assignment

Table 9-1 Plane and axis assignment

Command	Plane	Vertical infeed axis
G17	X/Y	Z
G18	Z/X	Y
G19	Y/Z	X

Messages output during execution of a cycle

During some cycles, messages that refer to the state of machining are displayed on the screen of the control system during program execution.

These message do not interrupt the program execution and continue to be displayed on the screen until the next message appears.

The message texts and their meaning are listed together with the cycle to which they refer. A summary is to be found in Section 9.7.4.

Block display during execution of a cycle

The cycle call is displayed in the current block display for the duration of the cycle.

Cycle call and parameter list

The defining parameters for the cycles can be transferred via the parameter list when the cycle is called.

Note

Cycle calls must always be programmed in a separate block.

Basic instructions with regard to the assignment of standard cycle parameters

The Programming Guide describes the parameter list of every cycle with the

- order and the
- type.

It is imperative to observe the order of the defining parameters.

Each defining parameter of a cycle has a certain data type. The parameter being used must be specified when the cycle is called. In the parameter list, you can transfer

- R parameters (only numerical values)
- constants.

If R parameters are used in the parameter list, they must first be assigned values in the calling program. Cycles can be called

- with an incomplete parameter list
  or
- by leaving out parameters.

If you want to exclude the last transfer parameters that have to be written in a call, you can prematurely terminate the parameter list with ")". If any parameters are to be omitted within the list, a comma "..., ,..." must be written as a placeholder.

No plausibility checks are made for parameter values with a limited range of values unless an error response has been specifically described for a cycle.

If when calling the cycle the parameter list contains more entries than parameters are defined in the cycle, the general NC alarm 12340 "Too many parameters" is displayed and the cycle is not executed.

Cycle call

The individual methods for writing a cycle are shown in the programming examples provided for the individual cycles.

Simulation of cycles

Programs with cycle calls can be tested first in simulation.

During simulation, the traversing movements of the cycle are visualized on the screen.

9.3 Graphical cycle support in the program editor

The program editor in the control system provides you with programming support to add cycle calls to the program and to enter parameters.

Function

The cycle support consists of three components:

1. Cycle selection
2. Input screenforms for parameter assignment
3. Help screen for each cycle (is to be found in the interactive screenform).

Overview of required files

The following files constitute the basis for cycle support:

- cov.com
- sc.com

Note

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

Operating the cycle selection

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

- Branching to selection bars for the individual cycles is possible in the horizontal softkey bar using the ”**Drilling**” and ”**Milling**” softkeys provided.
- The cycle selection is carried out using the vertical softkey bar until the appropriate input screenform with the help display appears on the screen.
- Then enter the values for the parameters.
  The values can be entered either directly (numerical values) or indirectly (R parameters, e.g. R27, or expressions consisting of R parameters, e.g. R27+10).
  If numerical values are entered, a check is carried out to see whether the value is within the admissible range.
- Some parameters that may have only a few values are selected using the toggle key.
- For drilling cycles, it is also possible to call a cycle modally using the vertical ”Modal Call” softkey.
  The modal call is selected via ”**Deselect modal”** from the drilling cycles list box.
- Press ”**OK**” to confirm (or ”**Abort**” in case of error).

Recompiling

Recompiling of program codes serves to make modifications to an existing program using the cycle support.

Position the cursor on the line to be modified and press the ”**Recompile**” softkey.

This will reopen the input screenform from which the program piece has been created, and you can modify the values.

9.4 Drilling cycles

9.4.1 General

Drilling cycles are motional sequences defined to DIN 66025 for drilling, boring, tapping etc.

They are called in the form of a subroutine with a defined name and a parameter list.

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

Table 9-2

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

The drilling cycles can be modal, i.e. they are executed at the end of each block that contains motion commands. Further cycles created by the user can also be called modally (see also Section 8.1.6 or 9.3).

There are two types of parameters:

- Geometrical parameters and
- Machining parameters

The geometrical parameters are identical for all drilling cycles, drilling pattern cycles and milling cycles. They define the reference and retraction planes, the safety clearance and the absolute or relative final drilling depth. Geometrical parameters are assigned once during the first drilling cycle CYCLE81.

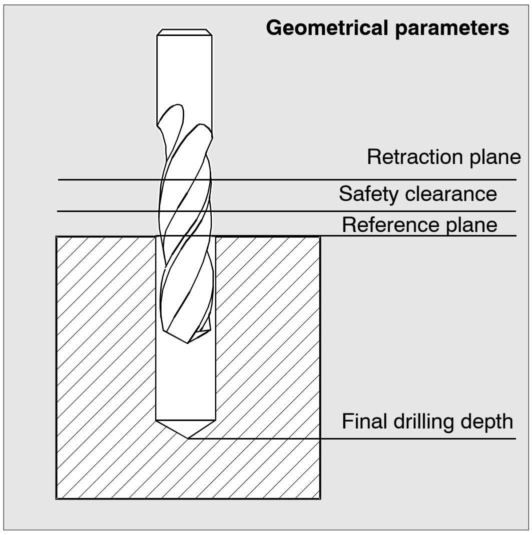


Fig. 9-2

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

9.4.2 Preconditions

Call and return conditions

Drilling cycles are programmed independently of the actual axis names. The drilling position must be approached in the higher-level program before the cycle is called.

The required values for feedrate, spindle speed and direction of spindle rotation must be programmed in the part program if there are no defining parameters in the drilling cycle.

The G functions and the current data record active before the cycle was called remain active beyond the cycle.

Plane definition

In the case of drilling cycles, it is generally assumed that the current workpiece coordinate system in which the machining operation is to be performed is to be defined by selecting plane G17, G18 or G19 and activating a programmable offset. The drilling axis is always the axis of this coordinate system which stands vertically to the current plane.

A tool length compensation must be selected before the cycle is called. Its effect is always perpendicular to the selected plane and remains active even after the end of the cycle.

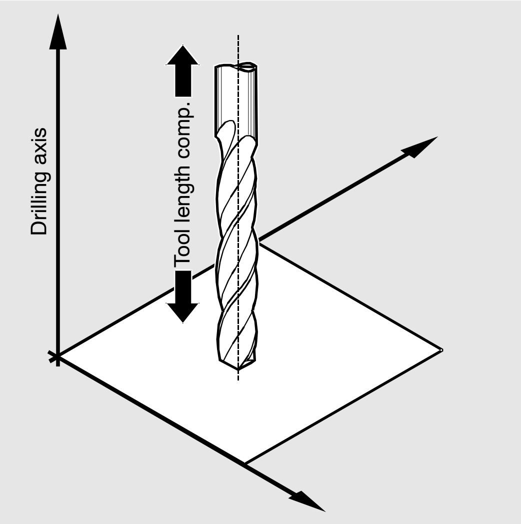


Fig. 9-3

Dwell time programming

The parameters for dwell times in the drilling cycles are always assigned to the F word and must therefore be assigned with values in seconds. Any deviations from this procedure must be expressly stated.

9.4.3 Drilling, centering – CYCLE81

Programming

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

Table 9-3 Parameters for CYCLE81

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

Function

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

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

Approach of the reference plane brought forward by the safety clearance by using G0

- Traversing to the final drilling depth at the feedrate programmed in the calling program (G1)
- Retraction to the retraction plane with G0

Explanation of the parameters

RFP and RTP (reference plane and retraction plane)

Normally, reference plane (RFP) and return plane (RTP) have different values. In the cycle, it is assumed that the retraction plane is ahead of the reference plane. This means that the distance from the retraction plane to the final drilling depth is larger than the distance from the reference plane to the final drilling depth.

SDIS (safety clearance)

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

DP and DPR (final drilling depth)

The final drilling depth can be specified either absolute (DP) or relative (DPR) to the reference plane.

With relative specification, the cycle will calculate the resulting depth automatically using the positions of reference and retraction planes.

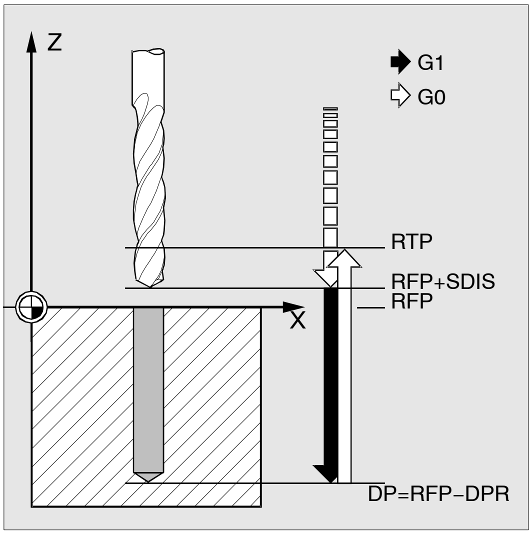


Fig. 9-4

Further notes

If a value is entered both for DP and for DPR, the final drilling depth is derived from DPR. If this differs from the absolute depth programmed via DP,

the message ”Depth: Corresponding to value for relative depth” is output in the dialog line.

If the values for reference and retraction planes are identical, a relative depth specification is not permitted. The error message

61101 ”Reference plane not correctly defined”, and the cycle is not executed. This error message is also output if the retraction plane is located after the reference plane, i.e. its distance to the final drilling depth is smaller.

Programming example: Drilling_centering

By using this program, you can produce 3 drill holes using the CYCLE81 drilling cycle, whereby this is called using different parameters. The drilling axis is always the Z axis.

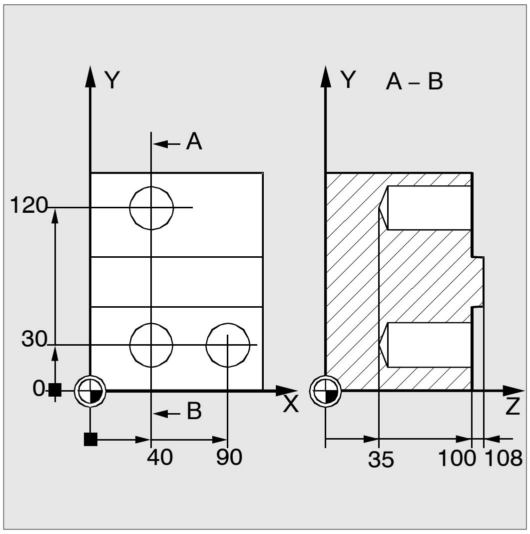


Fig. 9-5

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

9.4.4 Drilling, counterboring – CYCLE82

Programming

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

Parameters

Table 9-4 Parameters for CYCLE82

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DTB	real	Dwell time at final drilling depth (chip breaking)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. A dwell time can be allowed to elapse when the final drilling depth has been reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the final drilling depth with the feedrate (G1) programmed prior to the cycle call
- Dwell time at final drilling depth
- Retraction to the retraction plane with G0

Explanation of the parameters

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

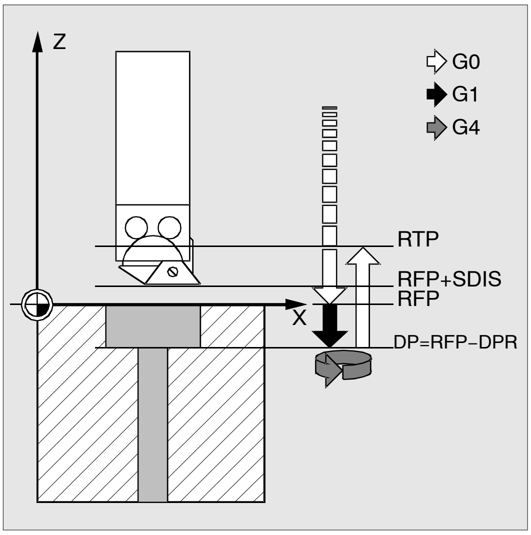


Fig. 9-6

DTB (dwell time)

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

Note

If a value is entered both for DP and for DPR, the final drilling depth is derived from DPR. If this differs from the absolute depth programmed via DP, the message ”Depth: Corresponding to value for relative depth” is output in the message line.

If the values for reference and retraction planes are identical, a relative depth specification is not permitted. The error message 61101 ”Reference plane defined incorrectly” is output and the cycle is not executed. This error message is also output if the retraction plane is located after the reference plane, i.e. its distance to the final drilling depth is smaller.

Programming example: Boring_counterboring

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

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

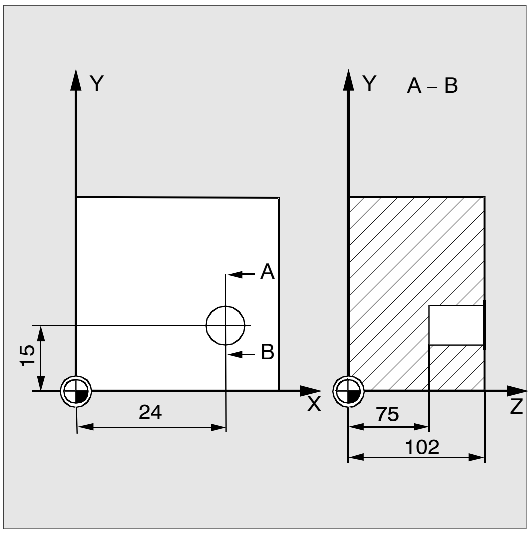


Fig. 9-7 Example

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

9.4.5 Deep hole drilling – CYCLE83

Programming

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

Parameters

Table 9-5 Parameters for CYCLE83

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
FDEP	real	First drilling depth (absolute)
FDPR	real	First drilling depth relative to the reference plane (enter without sign)
DAM	real	Amount of degression (enter without sign)
DTB	real	Dwell time at final drilling depth (chip breaking)
DTS	real	Dwell time at starting point and for swarf removal
FRF	real	Feedrate factor for the first drilling depth (enter without sign) Range of values: 0.001 ... 1
VARI	int	Machining type: Chip breaking=0 Swarf removal=1

Function

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

Deep hole drilling is performed with a depth infeed of a maximum definable depth executed several times, increasing gradually until the final drilling depth is reached.

The drill can either be retracted to the reference plane + safety clearance after every infeed depth for swarf removal or retracted in each case by 1 mm for chip breaking.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

Deep hole drilling with swarf removal (VARI=1):

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the first drilling depth with G1, the feedrate for which is derived from the feedrate defined with the program call which is subject to parameter FRF (feedrate factor)
- Dwell time at final drilling depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance for swarf removal by using G0
- Dwell time at the starting point (parameter DTS)
- Approach of the drilling depth last reached, reduced by anticipation distance by using G0
- Traversing to the next drilling depth with G1 (sequence of motions is continued until the final drilling depth is reached)
- Retraction to the retraction plane with G0

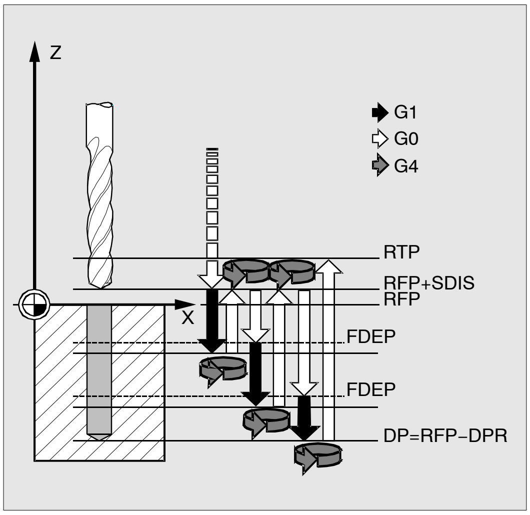


Fig. 9-8 Deep hole drilling with swarf removal (VARI=1)

Deep hole drilling with chip breaking (VARI=0):

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the first drilling depth with G1, the feedrate for which is derived from the feedrate defined with the program call which is subject to parameter FRF (feedrate factor)
- Dwell time at final drilling depth (parameter DTB)
- Retraction by 1 mm from the current drilling depth with G1 and the feedrate programmed in the calling program (for chip breaking)
- Traversing to the next drilling depth with G1 and the programmed feedrate (sequence of motions is continued until the final drilling depth is reached)
- Retraction to the retraction plane with G0

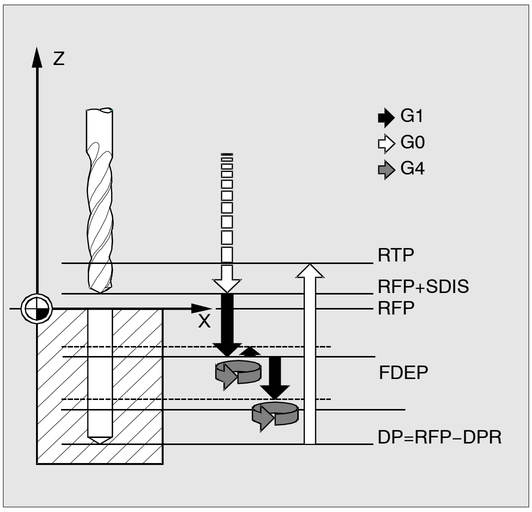


Fig. 9-9 Deep hole drilling with swarf removal (VARI=0)

Explanation of the parameters

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

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

The intermediate drilling depth are calculated in the cycle on the basis of final drilling depth, first drilling depth and amount of degression as follows:

- In the first step, the depth parameterized with the first drilling depth is traversed as long as it does not exceed the total drilling depth.
- From the second drilling depth on, the drilling stroke is obtained by subtracting the amount of degression from the stroke of the last drilling depth, provided that the latter is greater than the programmed amount of degression.
- The next drilling strokes correspond to the amount of degression, as long as the remaining depth is greater than twice the amount of degression.
- The last two drilling strokes are divided and traversed equally and are therefore always greater than half of the amount of degression.
- If the value for the first drilling depth is incompatible with the total depth, the error message 61107 ”First drilling depth defined incorrectly” is output and the cycle is not executed.

The parameter FDPR has the same effect in the cycle as the parameter DPR. If the values for the reference and retraction planes are identical, the first drilling depth can be defined as a relative value.

If the first drilling depth is programmed larger than the final drilling depth, the final drilling depth is never exceeded. The cycle will reduce the first drilling depth automatically as far as the final drilling depth is reached when drilling only once, and will therefore drill only once.

DTB (dwell time)

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

DTS (dwell time)

The dwell time at the starting point is only performed if VARI=1 (swarf removal).

FRF (feedrate factor)

With this parameter, you can specify a reduction factor for the active feedrate which only applies to the approach to the first drilling depth in the cycle.

VARI (machining type)

If parameter VARI=0 is set, the drill retracts 1 mm after reaching each drilling depth for chip breaking. If VARI=1 (for swarf removal), the drill traverses in each case to the reference plane brought forward by the safety clearance.

Note

The anticipation distance is calculated internally in the cycle as follows:

- If the drilling depth is 30 mm, the value of the anticipation distance is always 0.6 mm.
- For larger drilling depths, the formula drilling depth /50 is used (maximum value 7 mm).

Programming example – deep hole drilling

This program executes the cycle CYCLE83 at the positions X80 Y120 and X80 Y60 in the XY plane. The first drill hole is drilled with a dwell time zero and machining type chip breaking. The final drilling depth and the first drilling depth are entered as absolute values. In the second cycle call, a dwell time of 1 s is programmed. Machining type swarf removal is selected, the final drilling depth is relative to the reference plane. The drilling axis in both cases is the Z axis.

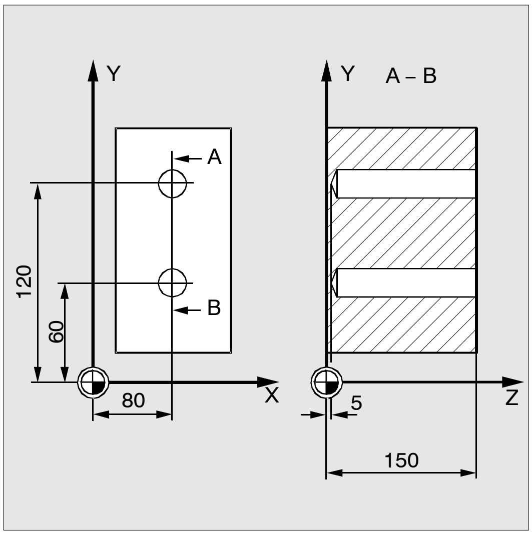


Fig. 9-10

N10 G0 G17 G90 F50 S500 M4	Specification of the technological values
N20 D1 T12	Approaching the retraction plane
N30 Z155	
N40 X80 Y120	Approach of the first drilling position

N50 CYCLE83(155, 150, 1, 5, 0 , 100, , 20, 0, 0, 1, 0)	Call of cycle; depth parameters with absolute values
N60 X80 Y60	Approach of next drill position
N70 CYCLE83(155, 150, 1, , 145, , 50, 20, 1, 1, 0.5, 1)	Call of cycle with relative specifications of final drilling depth and 1st drilling depth; the safety clearance is 1 mm, the feedrate factor 0.5
N80 M02	End of program

9.4.6 Rigid tapping – CYCLE84

Programming

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

Parameters

Table 9-6 Parameters for CYCLE84

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DTB	real	Dwell time at thread depth (chip breaking)
SDAC	int	Direction of rotation after end of cycle Values: 3, 4 or 5 (for M3, M4 or M5)
MPIT	real	Pitch as thread size (signed) Range of values 3 (for M3) ... 48 (for M48); the sign determines the direction of rotation in the thread
PIT	real	Pitch as a value (signed) Range of values: 0.001 ... 2000.000 mm); the sign determines the direction of rotation in the thread
POSS	real	Spindle position for oriented spindle stop in the cycle (in degrees)
SST	real	Speed for tapping
SST1	real	Speed for retraction

Function

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

CYCLE84 can be used to perform rigid tapping operations. For tapping with compensating chuck, a separate cycle CYCLE840 is provided.

Note

CYCLE84 can be used if the spindle to be used for the boring operation is technically able to be operated in the position–controlled spindle operation.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Oriented spindle stop (value in the parameter POSS) and switching the spindle to axis mode
- Tapping to final drilling depth and speed SST
- Dwell time at thread depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance, speed SST1 and direction reversal
- Retraction to the retraction plane with G0; spindle mode is reinitiated by reprogramming the spindle speed active before the cycle was called and the direction of rotation programmed under SDAC

Explanation of the parameters

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

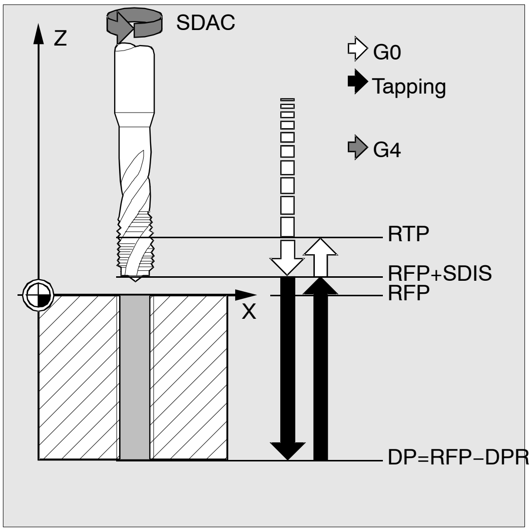


Fig. 9-11

DTB (dwell time)

The dwell time must be programmed in seconds. When tapping blind holes, it is recommended to omit the dwell time.

SDAC (direction of rotation after end of cycle)

Under SDAC, the direction of rotation after end of cycle is programmed.
The direction reversal when tapping is carried out automatically internally in the cycle.

MPIT and PIT (as a thread size and as a value)

The value for the thread pitch can be defined either as the thread size (for metric threads between M3 and M48 only) or as a value (distance from one thread turn to the next as a numerical value). The parameter not required in each case is omitted in the call or is assigned the value zero.

RH or LH threads are defined by the sign of the pitch parameters:

- positive value → RH (as for M3)
- negative value → LH (as for M4)

If the two thread pitch parameters have conflicting values, alarm 61001 "Thread pitch wrong" is generated by the cycle and cycle execution is aborted.

POSS (spindle position)

Before tapping, the spindle is stopped with orientation in the cycle and switched to position control.
The spindle position for this spindle stop is programmed under POSS.

SST (speed)

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

SST1 (retraction speed)

The speed for retraction from the tapped hole is programmed under SST1.
If this parameter is assigned the value zero, retraction is carried out at the speed programmed under SST.

Note

The direction of rotation when tapping in the cycle is always reversed automatically.

Programming example: Rigid tapping

A thread is tapped without compensating chuck at position X30 Y35 in the XY plane; the tapping axis is the Z axis. No dwell time is programmed; the depth is programmed as a relative value. The parameters for the direction of rotation and for the pitch must be assigned values. A metric thread M5 is tapped.

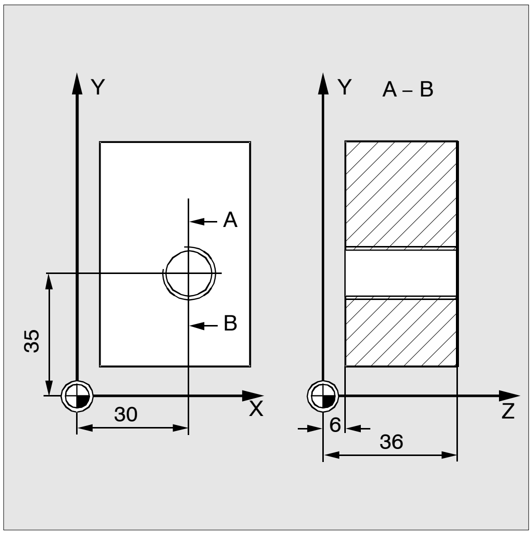


Fig. 9-12

N10 G0 G90 T11 D1	Specification of the technological values
N20 G17 X30 Y35 Z40	Approaching the drill position
N30 CYCLE84(40, 36, 2, , 30, , 3, 5, , 90, 200, 500)	Cycle call; parameter PIT has been omitted; no value is entered for the absolute depth or the dwell time; spindle stop at 90 degrees; speed for tapping is 200, speed for retraction is 500
N40 M02	End of program

9.4.7 Tapping with compensating chuck – CYCLE840

Programming

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

Parameters

Table 9-7 Parameter of CYCLE840

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)

Table 9-7 Parameter of CYCLE840, cont'd

SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DTB	real	Dwell time at thread depth (chip breaking)
SDR	int	Direction of rotation for retraction Values: 0 (automatic reversal of the direction of rotation) 3 or 4 (for M3 or M4)
SDAC	int	Direction of rotation after end of cycle Values: 3, 4 or 5 (for M3, M4 or M5)
ENC	int	Tapping with/without encoder Values: 0 = with encoder 1 = without encoder
MPIT	real	Thread pitch as the thread size (signed) Range of values 3 (for M3) ... 48 (for M48)
PIT	real	Pitch as a value (signed) Range of values: 0.001 ... 2000.000 mm

Function

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

Using this cycle, you can perform tapping with compensating chuck

- without encoder and
- with encoder.

Sequence of operations: Tapping with compensating chuck without encoder

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

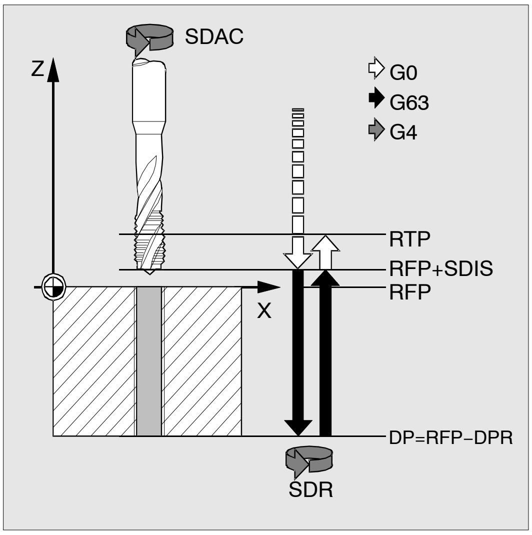


Fig. 9-13

- Approach of the reference plane brought forward by the safety clearance by using G0
- Tapping to the final drilling depth
- Dwell time at tapping depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance
- Retraction to the retraction plane with G0

Sequence of operations: Tapping with compensating chuck with encoder

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

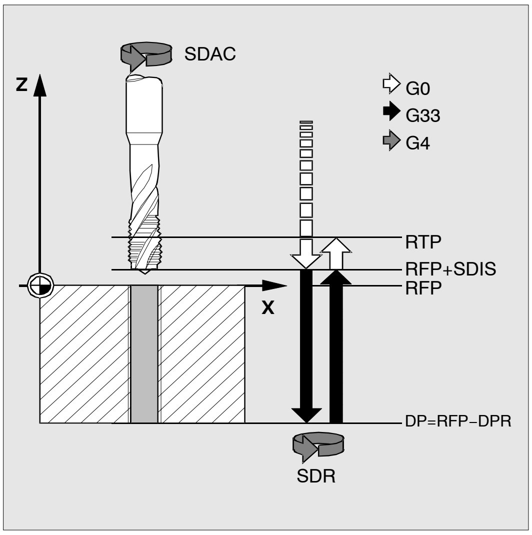


Fig. 9-14

- Approach of the reference plane brought forward by the safety clearance by using G0
- Tapping to the final drilling depth
- Dwell time at thread depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance
- Retraction to the retraction plane with G0

Explanation of the parameters

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

DTB (dwell time)

The dwell time must be programmed in seconds.

SDR (direction of rotation for retraction)

SDR=0 must be set if the spindle direction is to reverse automatically.

If the machine data are defined such that no encoder is set (in this case, machine data MD30200 NUM_ENCS is 0), the parameter must be assigned the value 3 or 4 for the direction of rotation; otherwise, alarm 61202 "No spindle direction programmed" is output and the cycle is aborted.

SDAC (direction of rotation)

Because the cycle can also be called modally (see Section 9.3), it requires a direction of rotation for tapping further threaded holes. This is programmed in parameter SDAC and corresponds to the direction of rotation programmed before the first call in the higher-level program. If SDR=0, the value assigned to SDAC has no meaning in the cycle and can be omitted in the parameterization.

ENC (tapping)

If tapping is to be performed without encoder although an encoder exists, parameter ENC must be assigned value 1.

If, however, no encoder is installed and the parameter is assigned the value 0, it is ignored in the cycle.

MPIT and PIT (as a thread size and as a value)

The parameter for the spindle pitch is only relevant if tapping is performed with encoder. The cycle calculates the feedrate from the spindle speed and the pitch.

The value for the thread pitch can be defined either as the thread size (for metric threads between M3 and M48 only) or as a value (distance from one thread turn to the next as a numerical value). The parameter not required in each case is omitted in the call or is assigned the value zero.

If the two thread pitch parameters have conflicting values, alarm 61001 "Thread pitch wrong" is generated by the cycle and cycle execution is aborted.

Further notes

Depending on the settings in machine data MD30200 NUM_ENCS, the cycle selects whether tapping is to be performed with or without encoder.

The direction of rotation for the spindle must be programmed with M3 orM4.

In thread blocks with G63, the values of the feedrate override switch and spindle speed override switch are frozen to 100%.

A longer compensating chuck is usually required for tapping without encoder.

Programming example: Tapping without encoder

In this program, a thread is tapped without encoder at position X35 Y35 in the XY plane; the tapping axis is the Z axis. The parameters SDR and SDAC for the direction of rotation must be assigned; parameter ENC is assigned the value 1, the value for the depth is the absolute value. Pitch parameter PIT can be omitted. A compensating chuck is used in machining.

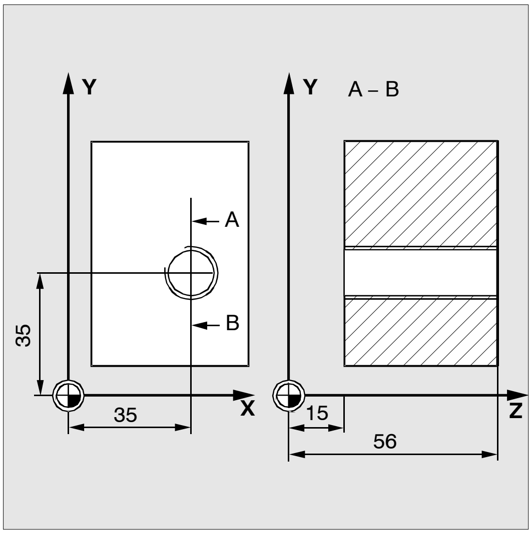


Fig. 9-15

N10 G90 G0 T11 D1 S500 M3	Specification of the technological values
N20 G17 X35 Y35 Z60	Approaching the drill position
N30 G1 F200	Determination of the path feed
N40 CYCLE840(59, 56, , 15, 0, 1, 4, 3, 1, ,)	Cycle call, dwell time 1 s, direction of rotation for retraction M4, direction of rotation after cycle M3, no safety clearance Parameters MPIT and PIT are omitted
N50 M02	End of program

Example: Tapping with encoder

In this program, a thread is tapped with encoder at position X35 Y35 in the XY plane. The drilling axis is the Z axis. The pitch parameter must be defined, automatic reversal of the direction of rotation is programmed. A compensating chuck is used in machining.

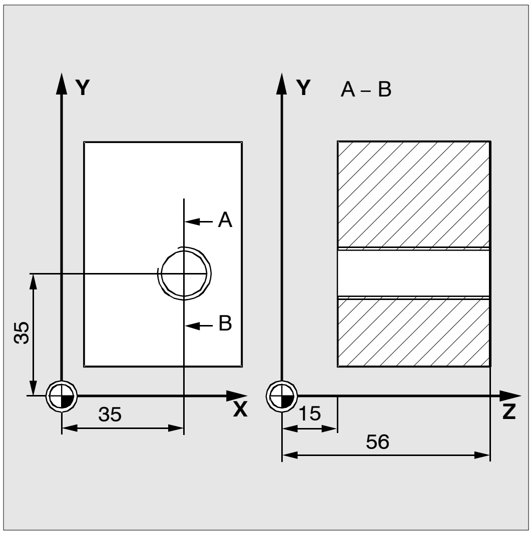


Fig. 9-16

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

9.4.8 Reaming 1 (boring 1) – CYCLE85

Programming

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

Parameters

Table 9-8 Parameters for CYCLE85

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

Table 9-8 Parameters for CYCLE85, cont'd

DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DTB	real	Dwell time at final drilling depth (chip breaking)
FFR	real	Feedrate
RFF	real	Retraction feedrate

Function

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

The inward and outward movement is performed at the feedrate assigned to FFR and RFF respectively.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the final drilling depth with G1 and at the feedrate programmed under the parameter FFR
- Dwell time at final drilling depth
- Retraction to the reference plane brought forward by the safety clearance with G1 and the retraction feedrate defined under the parameter RFF
- Retraction to the retraction plane with G0

Explanation of the parameters

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

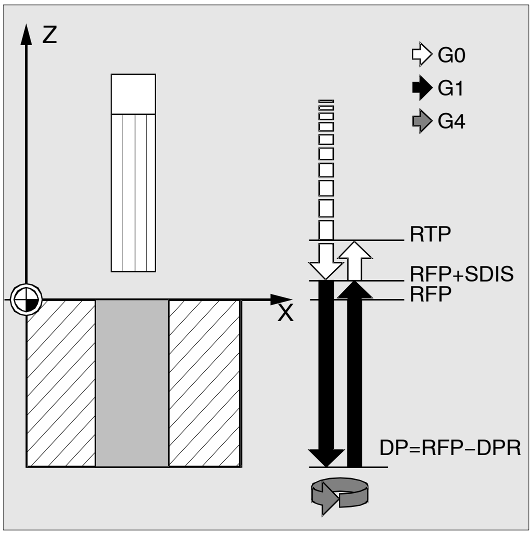


Fig. 9-17

DTB (dwell time)

The dwell time to the final drilling depth is programmed under DTB in seconds.

FFR (feedrate)

The feedrate value programmed under FFR is active in drilling.

RFF (retraction feedrate)

The feedrate value programmed under RFF is active when retracting from the hole to the reference plane + safety clearance.

Programming example: First boring pass

CYCLE85 is called at position Z70 X50 in the ZX plane. The boring axis is the Y axis. The value for the final drilling depth in the cycle call is programmed as a relative value; no dwell time is programmed. The workpiece upper edge is at Y102.

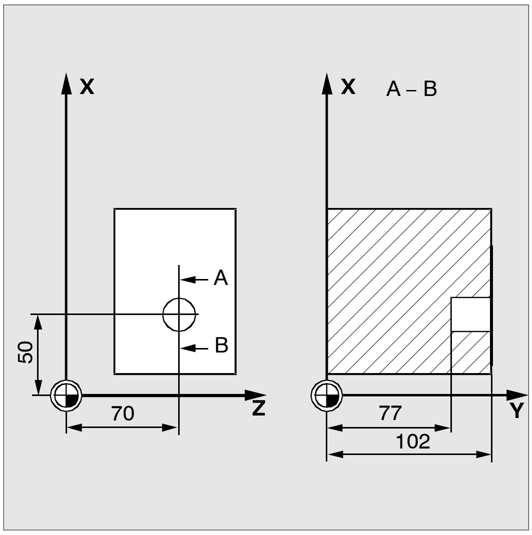


Fig. 9-18

N10 T11 D1	
N20 G18 Z70 X50 Y105	Approaching the drill position
N30 CYCLE85(105, 102, 2, , 25, , 300, 450)	Cycle call, no dwell time programmed
N40 M02	End of program

9.4.9 Boring (boring 2) – CYCLE86

Programming

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

Parameters

Table 9-9 Parameters for CYCLE86

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DTB	real	Dwell time at final drilling depth (chip breaking)
SDIR	int	Direction of rotation Values: 3 (for M3) 4 (for M4)
RPA	real	Retraction path in the 1st axis of the plane (incremental, enter with sign)
RPO	real	Retraction path in the 2nd axis of the plane (incremental, enter with sign)
RPAP	real	Retraction path in the boring axis (incremental, enter with sign)
POSS	real	Spindle position for oriented spindle stop in the cycle (in degrees)

Function

The cycle supports the boring of holes with a boring bar.

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

With boring 2, oriented spindle stop is activated once the drilling depth has been reached. Then, the programmed retraction positions are approached in rapid traverse and, from there, the retraction plane.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time to final drilling depth
- Oriented spindle stop at the spindle position programmed under POSS
- Traverse retraction path in up to three axes with G0
- Retraction in the boring axis to the reference plane brought forward by the safety clearance by using G0
- Retraction to the retraction plane with G0 (initial drilling position in both axes of the plane)

Explanation of the parameters

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

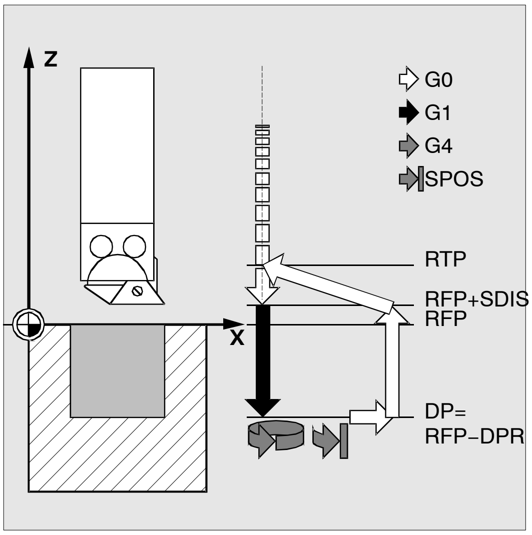


Fig. 9-19

DTB (dwell time)

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

SDIR (direction of rotation)

With this parameter, you determine the direction of rotation with which boring is performed in the cycle. If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is generated and the cycle is not executed.

RPA (retraction path in the 1st axis)

Use this parameter to define a retraction movement in the 1st axis (abscissa), which is executed after the final drilling depth has been reached and oriented spindle stop has been performed.

RPO (retraction path in the 2nd axis)

Use this parameter to define a retraction movement in the 2nd axis (ordinate), which is executed after the final drilling depth has been reached and oriented spindle stop has been performed.

RPAP (retraction path in the boring axis)

Use this parameter to define a retraction movement in the boring axis, which is executed after the final drilling axis has been reached and oriented spindle stop has been performed.

POSS (spindle position)

Use POSS to program the spindle position for the oriented spindle stop in degrees which is performed after the final drilling depth has been reached.

Note

It is possible to stop the active spindle with orientation. The angular value is programmed using a transfer parameter.

CYCLE84 can be used if the spindle to be used for the drilling operation is technically able to execute the SPOS command.

Programming example: Second boring pass

CYCLE86 is called at position X70 Y50 in the ZX plane. The drilling axis is the Z axis. The final drilling depth is programmed as an absolute value; no safety clearance is specified. The dwell time at the final drilling depth is 2 s. The workpiece upper edge is at Z110. In the cycle, the spindle is to rotate with M3 and to stop at 45 degrees.

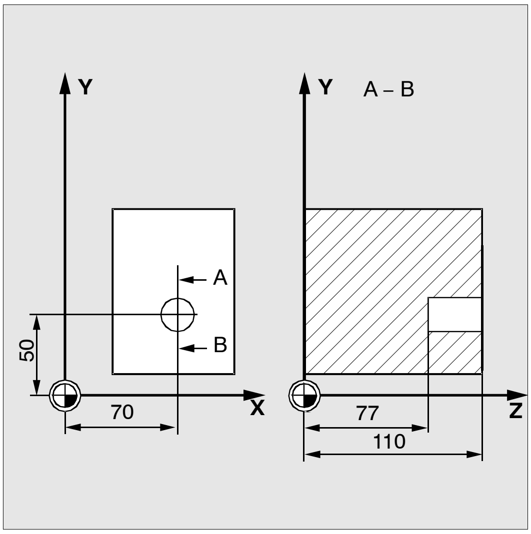


Fig. 9-20

N10 G0 G17 G90 F200 S300 M3	Specification of the technological values
N20 T11 D1 Z112	Approaching the retraction plane
N30 X70 Y50	Approaching the drill position
N40 CYCLE86(112, 110, , 77, 0, 2, 3, –1, –1, 1, 45)	Cycle call with absolute drilling depth
N50 M02	End of program

9.4.10 Boring with Stop 1 (boring 3) – CYCLE87

Programming

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

Parameters

Table 9-10 Parameter CYCLE87

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
SDIR	int	Direction of rotation Values: 3 (for M3) 4 (for M4)

Function

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

During boring 3, a spindle stop without orientation M5 is generated after reaching the final drilling depth, followed by a programmed stop M0. Pressing the NC START key continues the retraction movement at rapid traverse until the retraction plane is reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Spindle stop with M5
- Press NC START
- Retraction to the retraction plane with G0

Explanation of the parameters

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

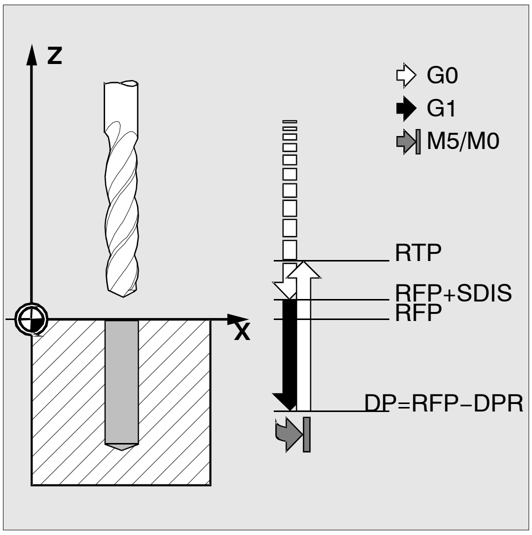


Fig. 9-21

SDIR (direction of rotation)

This parameter determines the direction of rotation with which the drilling operation is carried out in the cycle.

If values other than 3 or 4 (M3/M4) are generated, alarm 61102 ”No spindle direction programmed” is generated and the cycle is aborted.

Programming example: Third boring

CYCLE87 is called at position X70 Y50 in the XY plane. The drilling axis is the Z axis. The final drilling depth is specified as an absolute value. The safety clearance is 2 mm.

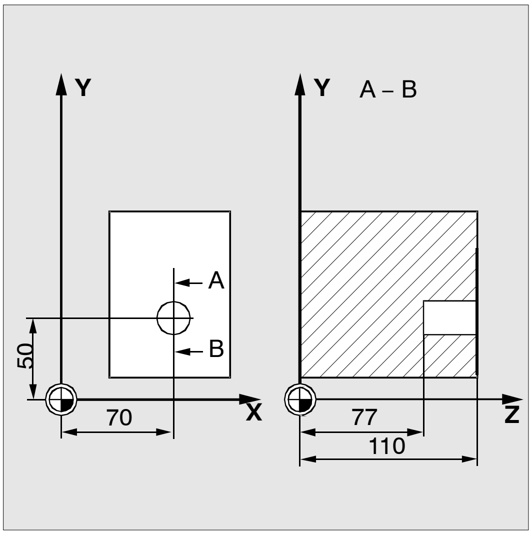


Fig. 9-22

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

9.4.11 Drilling with stop 2 (boring 4) – CYCLE88

Programming

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

Parameters

Table 9-11 Parameters for CYCLE88

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

Table 9-11 Parameters for CYCLE88, cont'd

DTB	real	Dwell time at final drilling depth (chip breaking)
SDIR	int	Direction of rotation Values: 3 (for M3) 4 (for M4)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. When boring with stop, a spindle stop without orientation M5 and a programmed stop are generated when the final drilling depth is reached. Pressing the NC START key continues the retraction movement at rapid traverse until the retraction plane is reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time at final drilling depth
- Spindle and program stop with M5 M0. After program stop, press the NC START key.
- Retraction to the retraction plane with G0

Explanation of the parameters

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

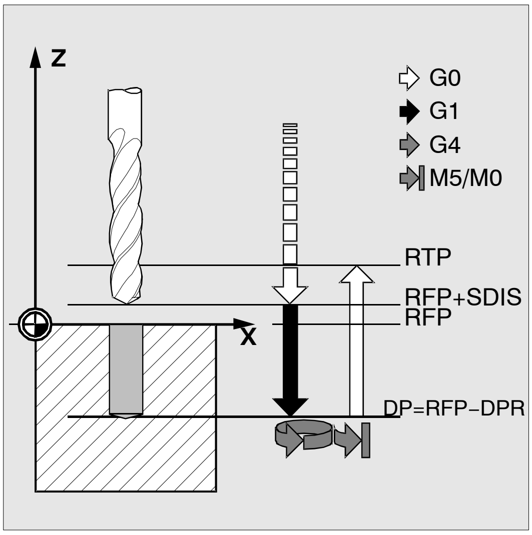


Fig. 9-23

DTB (dwell time)

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

SDIR (direction of rotation)

The programmed direction of rotation is active for the distance to be traversed to the final drilling depth.

If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is generated and the cycle is aborted.

Programming example: Fourth boring pass

CYCLE88 is called at position X80 Y90 in the XY plane. The drilling axis is the Z axis. The safety clearance is programmed with 3 mm; the final drilling depth is specified relative to the reference plane.

M4 is active in the cycle.

N10 G17 G90 F100 S450	Specification of the technological values
N20 G0 X80 Y90 Z105	Approach drilling position
N30 CYCLE88 (105, 102, 3, , 72, 3, 4)	Cycle call with programmed spindle direction M4
N40 M02	End of program

9.4.12 Reaming 2 (boring 5) – CYCLE89

Programming

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

Parameters

Table 9-12 Parameter CYCLE89

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DTB	real	Dwell time at final drilling depth (chip breaking)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. When the final drilling depth is reached, the programmed dwell time is active.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time to final drilling depth
- Retraction up to the reference plane brought forward by the safety clearance using G1 and the same feedrate value
- Retraction to the retraction plane with G0

Explanation of the parameters

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

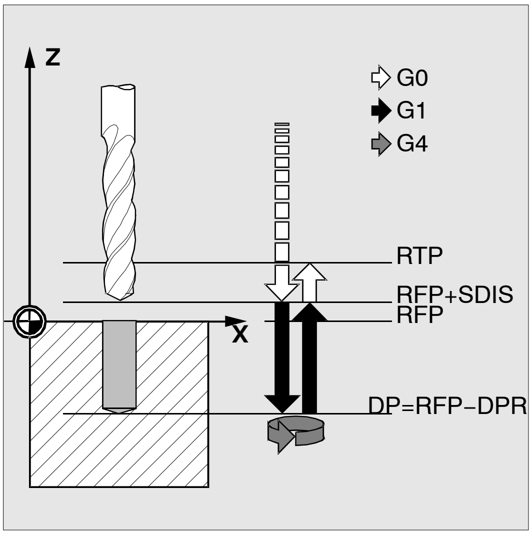


Fig. 9-24

DTB (dwell time)

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

Programming example: Fifth boring

At X80 Y90 in the XY plane, the drilling cycle CYCLE89 is called with a safety clearance of 5 mm and specification of the final drilling depth as an absolute value. The drilling axis is the Z axis.

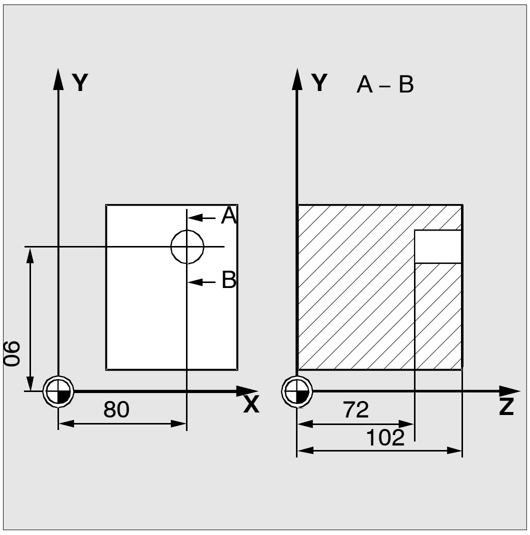


Fig. 9-25

DEF REAL RFP, RTP, DP, DTB	Definition of the parameters
RFP=102 RTP=107 DP=72 DTB=3	Value assignments
N10 G90 G17 F100 S450 M4	Specification of the technological values

N20 G0 X80 Y90 Z107	Approach drilling position
N30 CYCLE89(RTP, RFP, 5, DP, , DTB)	Cycle call
N40 M02	End of program

9.5 Drilling pattern cycles

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

9.5.1 Preconditions

Drilling pattern cycles without drilling cycle call

Drilling pattern cycles can also be used for other applications without the first drilling cycle first being called because the drilling pattern cycles can be parameterized without reference to the drilling cycle used.

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

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

Behavior when quantity parameter is zero

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

Checks in case of limited ranges of input values

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

9.5.2 Row of holes – HOLES1

Programming

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

Parameters

Table 9-13 Parameters for HOLES1

SPCA	real	1st axis of the plane (abscissa) of a reference point on the straight line (absolute)
SPCO	real	2nd axis of the plane (ordinate) of this reference point (absolute)
STA1	real	Angle to the 1st axis of the plane (abscissa) −180<STA1<=180 degrees
FDIS	real	Distance from the first hole to the reference point (enter without sign)
DBH	real	Distance between the holes (enter without sign)
NUM	int	Number of holes

Function

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

Sequence

To avoid unnecessary travel, the cycle calculates whether the row of holes is machined starting from the first hole or the last hole from the actual position of the plane axes and the geometry of the row of holes. The drilling positions are then approached one after the other at rapid traverse.

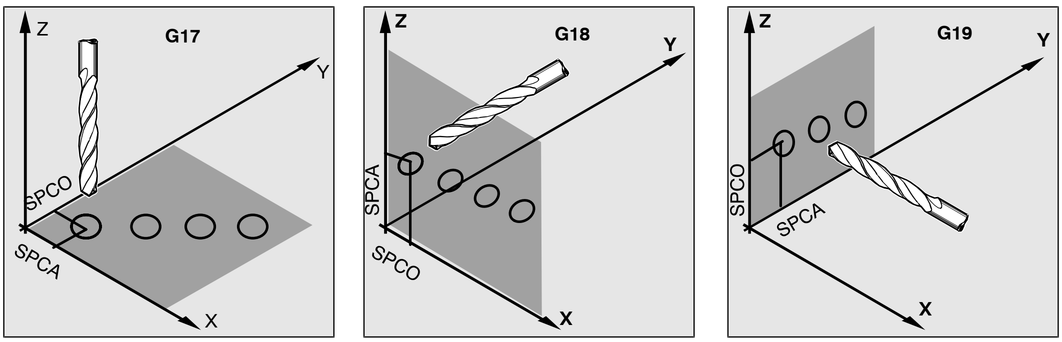


Fig. 9-26

Explanation of the parameters

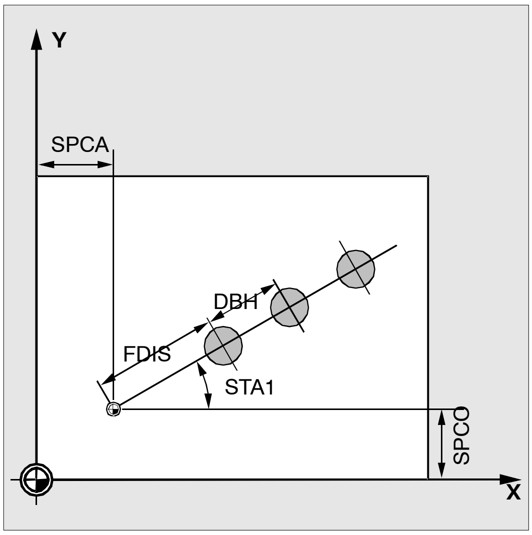


Fig. 9-27

SPCA and SPCO (reference point of 1st axis of the plane and 2nd axis of the plane)

One point along the straight line of the row of holes is defined as the reference point for determining the distances between the holes. The distance to the first hole FDIS is defined from this point.

STA1 (angle)

The straight line can be in any position in the plane. It is specified both by the point defined by SPCA and SPCO and by the angle contained by the straight line with the 1st axis of the plane of the workpiece coordinate system that is active when the cycle is called. The angle is entered under STA1 in degrees.

FDIS and DBH (distance)

The distance of the first hole and the reference point defined under SPCA and SPCO is programmed with FDIS. The parameter DBH contains the distance between any two holes.

NUM (number)

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

Programming example: Row of holes

Use this program to machine a row of holes consisting of 5 threaded holes arranged parallel to the Z axis of the ZX plane and which have a distance of 20 mm one to another. The starting point of the row of holes is at Z20 and X30 whereby the first hole has a distance of 10 mm from this point. The geometry of the row of holes is described by the cycle HOLES1. First, drilling is carried out using CYCLE82, and then tapping is performed using CYCLE84 (tapping without compensating chuck). The holes are 80 mm in depth (difference between reference plane and final drilling depth).

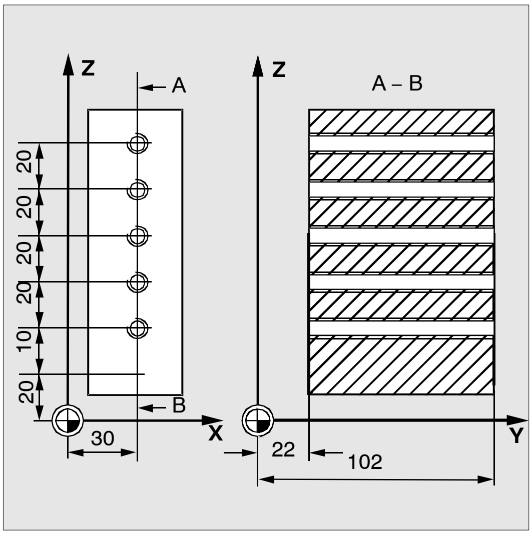


Fig. 9-28

N10 G90 F30 S500 M3 T10 D1	Specification of the technological values for the machining step
N20 G17 G90 X20 Z105 Y30	Approach starting position
N30 MCALL CYCLE82(105, 102, 2, 22, 0, 1)	Modal call of the drilling cycle
N40 HOLES1(20, 30, 0, 10, 20, 5)	Call of row-of-holes cycle; the cycle starts with the first hole; only the drill positions are approached in this cycle
N50 MCALL	Deselect modal call
...	Tool change
N60 G90 G0 X30 Z110 Y105	Traverse to position next to the 5th hole
N70 MCALL CYCLE84(105, 102, 2, 22, 0, , 3, , 4.2, ,300,)	Modal call of the tapping cycle
N80 HOLES1(20, 30, 0, 10, 20, 5)	call of the row–of–holes cycle started with the 5th hole in the row
N90 MCALL	Deselect modal call
N100 M02	End of program

Programming example: Grid of holes

Use this program to machine a grid of holes consisting of 5 rows with 5 holes each, which are arranged in the XY plane, with a distance of 10 mm between them. The starting point of the grid is at X30 Y20.

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

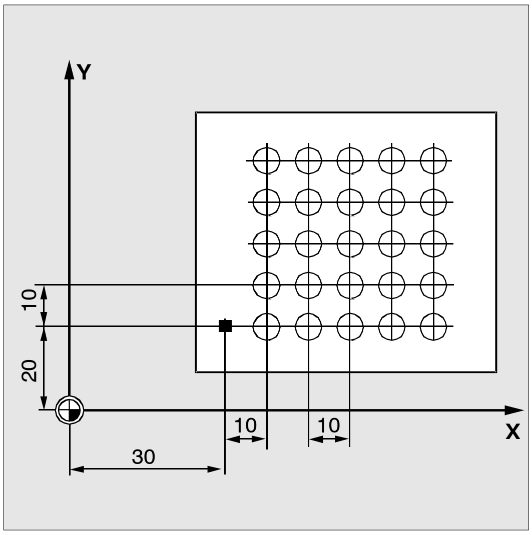


Fig. 9-29

R10=102	**Reference plane**
R11=105	Retraction plane
R12=2	Safety clearance
R13=75	Drilling depth
R14=30	Reference point: Row of holes of the 1st axis of the plane
R15=20	Reference point: Row of holes of the 2nd axis of the plane
R16=0	Starting angle
R17=10	Distance of the 1st hole to the reference point
R18=10	Distance between the holes
R19=5	Number of holes per row
R20=5	Number of rows
R21=0	Count of rows
R22=10	Distance between the rows

N10 G90 F300 S500 M3 T10 D1	Specification of the technological values
N20 G17 G0 X=R14 Y=R15 Z105	Approach starting position
N30 MCALL CYCLE82(R11, R10, R12, R13, 0, 1)	Modal call of the drilling cycle
N40 LABEL1:	Call of the row–of–holes cycle
N41 HOLES1(R14, R15, R16, R17, R18, R19)	
N50 R15=R15+R22	Calculate y value for the next line
N60 R21=R21+1	Increment line counter
N70 IF R21<R20 GOTOB LABEL1	Return to LABEL1 if the condition is fulfilled
N80 MCALL	Deselect modal call

N90 G90 G0 X30 Y20 Z105	Approach starting position
N100 M02	End of program

9.5.3 Circle of holes – HOLES2

Programming

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

Parameters

Table 9-14 Parameters for HOLES2

CPA	real	Center point of circle of holes (absolute), 1st axis of the plane
CPO	real	Center point of circle of holes (absolute), 2nd axis of the plane
RAD	real	Radius of circle of holes (enter without sign)
STA1	real	Starting angle Range of values: –180<STA1<=180 degrees
INDA	real	Incrementing angle
NUM	int	Number of holes

Function

Use this circle to machine a circle of holes. The machining plane must be defined before the cycle is called.

The type of hole is determined by the drilling hole cycle that has already been called modally.

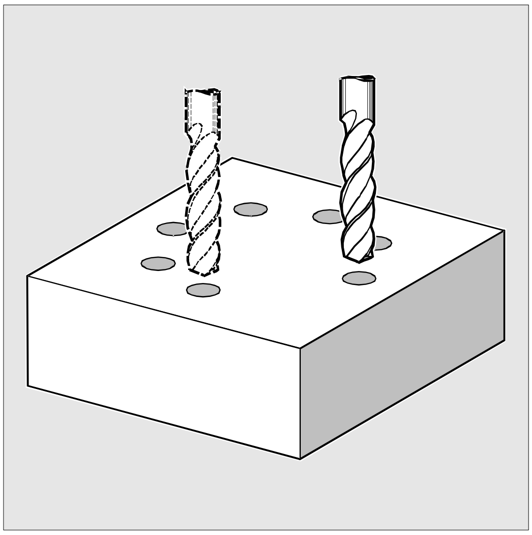


Fig. 9-30

Sequence

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

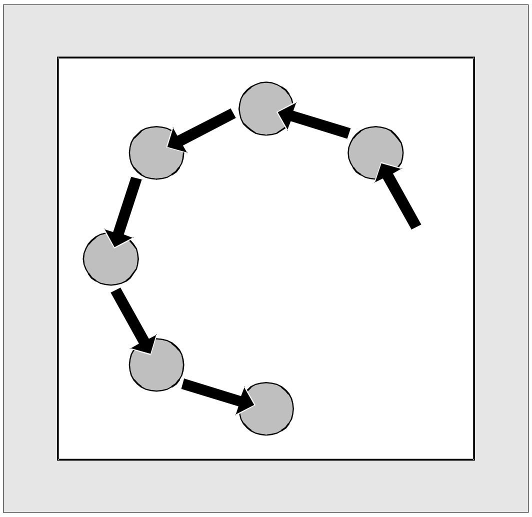


Fig. 9-31

Explanation of the parameters

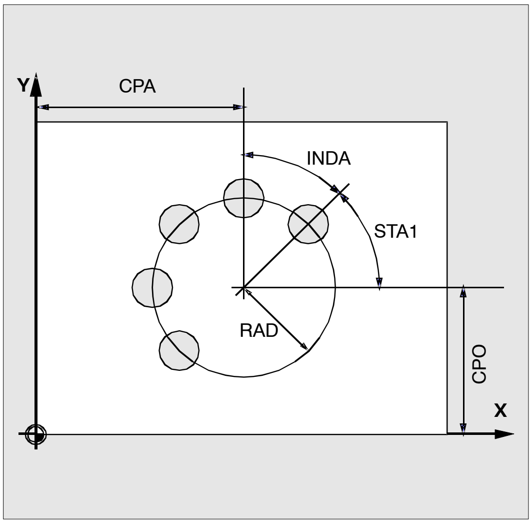


Fig. 9-32

CPA, CPO and RAD (center point position and radius)

The position of the circle of holes in the machining plane is defined via center point (parameters CPA and CPO) and radius (parameter RAD). Only positive values are permitted for the radius.

STA1 and INDA (starting and incremental angle)

These parameters define the arrangement of the holes on the circle of holes.

Parameter STA1 specifies the angle of rotation between the positive direction of the 1st axis (abscissa) in the workpiece coordinate system active before the cycle was called and the first hole. Parameter INDA contains the angle of rotation from one hole to the next.

If parameter INDA is assigned the value zero, the indexing angle is calculated internally from the number of holes which are positioned equally in a circle.

NUM (number)

Parameter NUM defines the number of holes.

Programming example: Circle of holes

The program uses CYCLE82 to produce 4 holes having a depth of 30 mm. The final drilling depth is specified as a relative value to the reference plane. The circle is defined by the center point X70 Y60 and the radius 42 mm in the XY plane. The starting angle is 33 degrees. The safety clearance along the drilling axis Z is 2 mm.

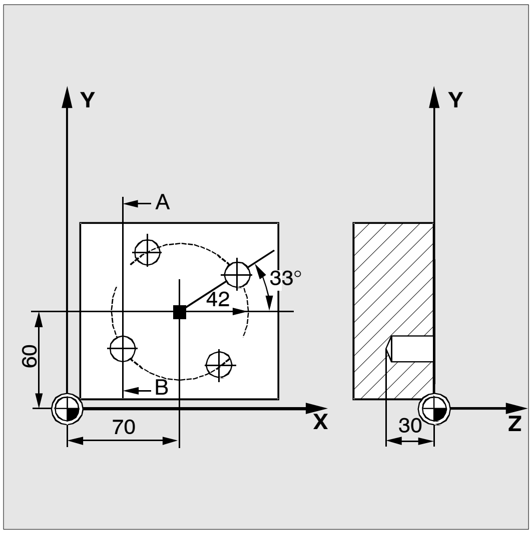


Fig. 9-33

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

9.6 Milling cycles

9.6.1 Preconditions

Call and return conditions

Milling cycles are programmed independently of the particular axis name.

Before you call the milling cycles, a tool compensation must be activated.

The appropriate values for feedrate, spindle speed and direction of rotation of spindle must be programmed in the part program if the appropriate parameters are not provided in the milling cycle.

The center point coordinates for the milling pattern or the pocket to e machined are programmed in a rectangular coordinate system.

The G functions active prior to the cycle call and the current programmable frame remain active beyond the cycle.

Plane definition

The milling cycles assume that the current coordinate system is reached by selecting a plane G17, G18 or G19 and activation of a programmable frame (if required). The infeed axis is always the 3rd axis of this coordinate system

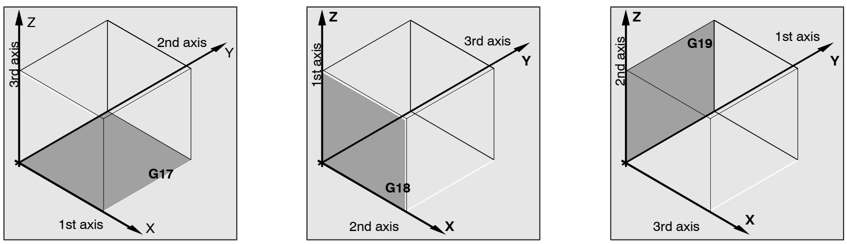


Fig. 9-34 Plane and axis assignment

Messages with regard to the machining state

During the execution of the milling cycles, various messages that refer to the machining status are displayed on the screen. The following messages are possible:

- ”Elongated hole <no.>First figure is being machined”
- ”Groove <no.>Another figure is being machined”
- ”Circumferential groove <no.>Last figure is being machined”

no.> in the message text always stands for the number of the figure currently machined.

These message do not interrupt the program execution and continue to be displayed until the next message is displayed or the cycle is completed.

9.6.2 Face milling – CYCLE71

Programming

CYCLE71(_RTP, _RFP, _SDIS, _DP, _PA, _PO, _LENG, _WID, _STA,_MID, _MIDA, _FDP, _FALD, _FFP1, _VARI, _FDP1)

Parameters

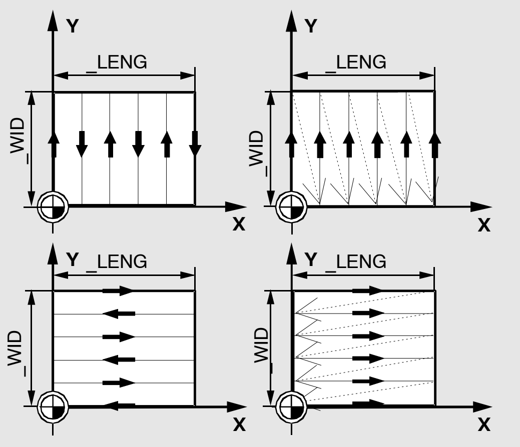
Table 9-15 Parameters for CYCLE71

_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (to be added to the reference plane; enter without sign)
_DP	real	Depth (absolute)
_PA	real	Starting point (absolute), 1st axis of the plane
_PO	real	Starting point (absolute), 2nd axis of the plane
_LENG	real	Length of the rectangle in the 1st axis, incremental. The corner from which the dimension starts results from the sign.
_WID	real	Length of the rectangle in the 2nd axis, incremental. The corner from which the dimension starts results from the sign.
_STA	real	Angle between longitudinal axis and the 1st axis of the plane (enter without sign) Range of values: $0° \le$ _STA $< 180°$
_MID	real	Maximum infeed depth (enter without sign)
_MIDA	real	Maximum infeed width during solid machining in the plane as a value (enter without sign)
_FDP	real	Retraction travel in the finishing direction (incremental, enter without sign)
_FALD	real	Finishing dimension in the depth (incremental, enter without sign)
_FFP1	real	Feedrate for surface machining
_VARI	integer	Machining type (enter without sign) UNIT DIGIT Values: 1 roughing 2 finishing TENS DIGIT Values: 1 parallel to the 1st axis of the plane, in one direction 2 parallel to the 2nd axis of the plane, in one direction 3 parallel to the 1st axis of the plane 4 parallel to the 2nd axis of the plane, with alternating direction
_FDP1	real	Overrun travel in the direction of the plane infeed (incremental, enter without sign)

Function

Use CYCLE71 to face mill any rectangular surface. The cycle distinguishes between roughing (solid machining of the surface in several steps up to finishing dimension) and finishing (end milling the surface in one step). The maximum infeed in width and depth can be specified.

The cycle operates without cutter radius compensation. The depth infeed is carried out in the open.



Possible solid machining strategies for face milling

Fig. 9-35

Sequence

Position reached prior to cycle start:

Starting position is any position from which the infeed point can be approached at the height of the retraction plane without collision.

The cycle creates the following sequence of motions:

- With G0, the infeed point is approached at the height of the current position, and then, from this position, with G0, too, the reference plane brought forward by the safety clearance is approached; then, also with G0, feeding to the machining plane. G0 is possible since infeed in the open is possible.
  Various solid machining strategies are provided (paraxially in one direction or oscillating to and fro).
- Sequence of motions when roughing:

  According to the programmed values _DP, _MID and _FALD, face milling can be carried out in several planes. Machining is carried out from the top downwards, i.e. one plane each is removed and then the next depth infeed is carried out in the open (parameters _FDP). The traversing paths for solid machining in the plane depend on the values of the parameters _LENG, _WID, _MIDA, _FDP, _FDP1 and the cutter radius of the active tool.

The first path to be milled is always traversed such that the infeed depth exactly corresponds to _MIDA so that no width infeed larger than the maximum possible width infeed results. The tool center point thus does not always travel exactly on the edge (only if _MIDA = cutter radius). The dimension by which the tool traverses outside the edge is always equal to cutter diameter – _MIDA even if only 1 surface cut is performed, i. e. surface width + overrun less _MIDA. The other paths for width infeed are calculated internally so that as to produce a uniform path width (<=_MIDA).

- Sequence of motions when finishing:

  When finishing, the surface is milled once in the plane. This means that the finishing allowance when roughing has to be selected also such that the residual depth can be removed with the finishing tool in one step.

  After each surface milling pass in the plane, the tool will retract. The retraction travel is programmed under the parameter _FDP.
  When machining in one direction, the tool will retract in one direction by finishing allowance + safety clearance, and the next starting point is approached at rapid traverse.

  When roughing in one direction, the tool will retract by the calculated infeed depth + safety clearance. The depth infeed is performed at the same point as in roughing.
  After completion of finishing, the tool will retract to the retraction plane _RTP at the position last reached.

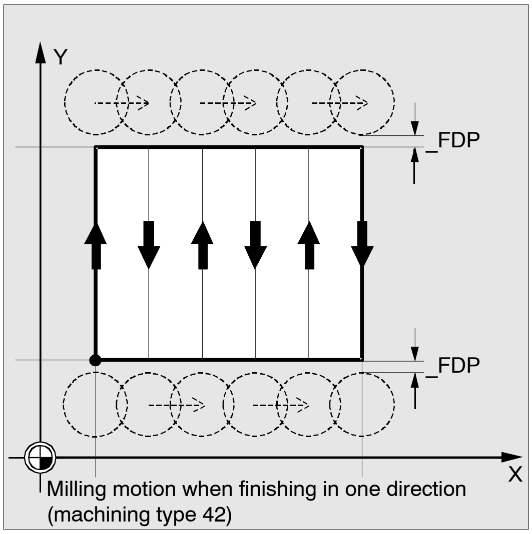


Fig. 9-36

Explanation of the parameters

For the parameters _RTP, _RFP, _SDIS, see CYCLE81.

For the parameters _STA, _MID, _FFP1, see POCKET3.

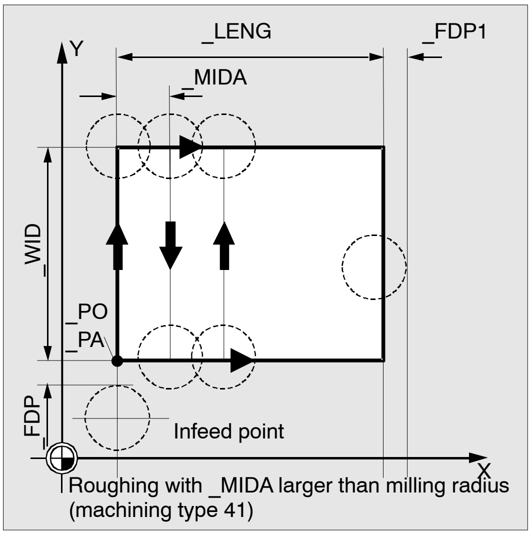


Fig. 9-37

_DP (depth)

The depth can be specified as an absolute value (_DP) to the reference plane.

_PA, _PO (starting point)

Use the parameters _PA and _PO to define the starting point of the area in the axes of the plane.

_LENG, _WID (length)

Use the parameters _LENG and WID to define the length and width of a rectangle in the plane. The position of the rectangle referred to _PA and _PO results from the sign.

_MIDA (max. infeed width)

Use this parameter to define the maximum infeed width when solid machining in a plane. Analogously to the known calculation method for the infeed depth (equal distribution of the total depth with the maximum possible value), the width is distributed equally, maximally with the value programmed under _MIDA.

If this parameter is not programmed or has value 0, the cycle will internally use 80% of the milling tool diameter as the maximum infeed depth.

_FDP (retraction travel)

Use this parameter to define the dimension for the retraction travel in the plane. This parameter should reasonably always have a value greater than zero.

_FDP1 (overrun travel)

Use this parameter to specify an overrun travel in the direction of the plane infeed (_MIDA). Thus, it is possible to compensate the difference between the current cutter radius and the tool nose radius (e.g. cutter radius or cutting tips arranged at an angle). The last cutter center point path thus always results as _LENG (or _WID) + _FDP1 − tool radius (from the compensation table).

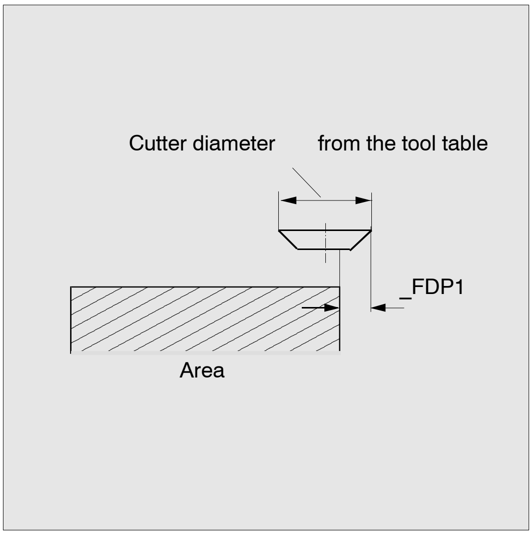


Fig. 9-38

_FALD (finishing allowance)

In roughing, a finishing allowance in the depth is taken into account which is programmed under this parameter.

The residual material remained as the finishing allowance must always be specified for finishing so to ensure that the tool can be retracted and then fed to the starting point of the next cut without collision.

_VARI (machining type)

Use the parameter _VARI to define the machining type.

Possible values are:

Units digit:

1=roughing to finishing allowance

2=finishing

Tens digit:

1=parallel to the 1st axis of the plane; in one direction

2=parallel to the 2nd axis of the plane; in one direction

3=parallel to the 1st axis of the plane; with alternating direction

4=parallel to the 2nd axis of the plane; with alternating direction

If a different value is programmed for the parameter _VARI, the cycle is aborted after output of alarm 61002 ”Machining type defined incorrectly”.

Further notes

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 ”No tool compensation active” is output.

Programming example: Face milling

Parameters for the cycle call:

• Retraction plane:	10 mm
• Reference plane:	0 mm
• Safety clearance:	2 mm
• Milling depth:	–11 mm
• Starting point of the rectangle	X = 100 mm Y = 100 mm
• Rectangle size	X = +60 mm Y = +40 mm
• Angle of rotation in the plane	10 degrees
• Max. infeed depth	6 mm
• Max. infeed depth	10 mm
• Retraction travel at the end of the milling path:	5 mm
• No finishing allowance	–
• Feedrate for surface machining:	4,000 mm/min
• Machining type: Roughing parallel to the X axis with alternating direction	
• Overrun at the last cut due to the cutting edge geometry	2 mm

A milling cutter with 10 mm radius is used.

N10 T2 D2	
N20 G17 G0 G90 G54 G94 F2000 X0 Y0 Z20	Approach starting position
N30 CYCLE71(10, 0, 2, –11, 100, 100, 60, 40, 10, 6, 10, 5, 0, 4000, 31, 2)	Cycle call
N40 G0 G90 X0 Y0	
N50 M02	End of program

9.6.3 Contour milling – CYCLE72

Programming

CYCLE72 (_KNAME, _RTP, _RFP, _SDIS, _DP, _MID, _FAL, _FALD, _FFP1, _FFD, _VARI, _RL, _AS1, _LP1, _FF3, _AS2, _LP2)

Parameters

Table 9-16 Parameters for CYCLE72

_KNAME	string	Name of contour subroutine
_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (to be added to the reference plane; enter without sign)
_DP	real	Depth (absolute)
_MID	real	Maximum infeed depth (incremental; enter without sign)
_FAL	real	Finishing allowance at the edge contour (enter without sign)
_FALD	real	Finishing allowance at the base (incremental, enter without sign)
_FFP1	real	Feedrate for surface machining
_FFD	real	Feedrate for depth infeed (enter without sign)
_VARI	integer	Machining type (enter without sign) UNIT DIGIT Values: 1 roughing 2 finishing TENS DIGIT Values: 0 Intermediate path with G0 1 Intermediate path with G1 HUNDREDS DIGIT Values: 0 Retraction at contour end to _RTP 1 Retraction at contour end to _RFP + _SDIS 2 Retraction at contour end by _SDIS 3 No retraction at the contour end
_RL	integer	Traveling around the contour either centrally, to the right or to the left (with G40, G41 or G42; enter without sign) Values: 40...G40 (approach and retraction – only one straight line) 41...G41 42...G42

Table 9-16 Parameters for CYCLE72, cont'd

_AS1	integer	Specification of the approach direction/approach path: (enter without sign) Units digit: Values: 1...Straight line tangential 2...Quadrant 3...Semicircle Tens digit: Values: 0...Approach of contour in the plane 1...Approach of contour along a spatial path
_LP1	real	Length of the approach travel (with straight–line) or radius of the approach arc (with circle) (enter without sign)

The other parameters can be selected as options.

_FF3	real	Retraction feedrate and feedrate for intermediate positions in the plane (in the open)
_AS2	integer	Specification of the retraction direction/retraction path: (enter without sign) Units digit: Values: 1...Straight line tangential 2...Quadrant 3...Semicircle Tens digit: Values: 0...Retraction from contour in the plane 1...Retraction from contour along a spatial path
_LP2	real	Length of the retraction travel (with straight–line) or radius of the retraction arc (with circle) (enter without sign)

Function

Use CYCLE72 to mill along any contour defined in any contour defined in a subroutine. The cycle operates with or without cutter radius compensation.

It is not imperative that the contour is closed. Internal or external machining is defined via the position of the cutter radius compensation (centrally, left or right to the contour).

The contour must be programmed in the direction as it is to be milled and must consist of a minimum of 2 contour blocks (starting and end point), since the contour subroutine is called directly internally in the cycle.

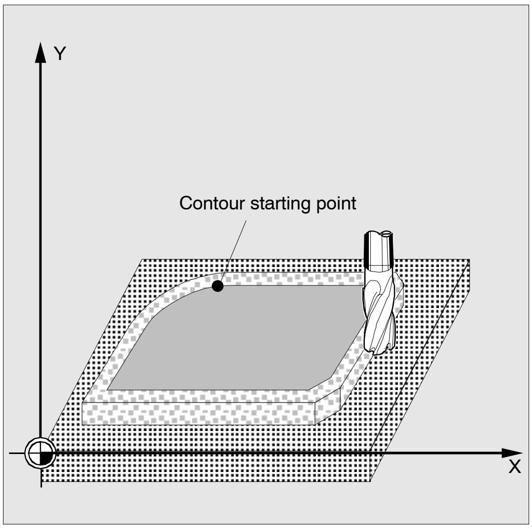


Fig. 9-39

Functions of the cycle:

- Selection of roughing (single–pass traversing parallel to contour, taking into account a finishing allowance, if necessary at several depth until the finishing allowance is reached) and finishing (single–pass traversing along the final contour if necessary at several depths)
- Smooth approach to and retraction from the contour either tangentially or radially (quadrant or semicircle)
- Programmable depth infeeds

Intermediate motions either at rapid traverse rate or at feedrate

Sequence

Position reached prior to cycle start:

Starting position is any position from which the contour starting point can be approached at the height of the retraction plane without collision.

The cycle generates the following sequence of motions when roughing:

The depth infeeds are distributed equally with the maximum possible value of the specified parameters.

- Traversing to the starting point for first milling with G0/G1 (and _FF3). This point is calculated internally in the control system and depends on
  - the contour starting point (first point in the subroutine),
  - the direction of the contour at the starting point,
  - the approach mode and its parameters and
  - the tool radius.
    In this block, the cutter radius compensation is activated.

- Depth infeed to the first or next machining depth plus programmed safety clearance with G0/G1. The first machining depth results from
  - the total depth,
  - the finishing allowance and
  - the maximum possible depth infeed.
- Approach of the contour vertically with depth infeed and then in the plane at the programmed feedrate or 3D with the feedrate programmed under _FAD according to the programming for smooth approach.
- Milling along the contour with G40/G41/G42.
- Smooth retraction from the contour with G1 and always still feedrate for the surface machining by the retraction amount.
- Retraction with G0 /G1 (and feedrate for intermediate paths _FF3), depending on the programming.
- Retraction to the depth infeed point with G0/G1 (and _FF3).
- This sequence is repeated on the next machining plane up to finishing allowance in the depth.

A completion of roughing, the tool stands above the point (calculated internally in the control) of retraction from the contour at the height of the retraction plane.

The cycle generates the following sequence of motions when finishing:

During finishing, milling is performed at the relevant infeed along the base of the contour until the final dimension is reached.

Smooth approach and retraction of the contour is carried out according to the existing parameters. The appropriate path is calculated internally in the control system.

At the end of the cycle, the tool is positioned at the contour retraction point at the height of the retraction level.

Further notes:
Contour programming

When programming the contour, please observe the following:

- No programmable offset must be selected in the subroutine prior to the first programmed position.
- The first block of the contour subroutine is a straight line block containing G90 / G0 or G90 / G1 and defines the start of the contour.
- The starting condition of the contour is the first position in the machining plane which is programmed in the contour subroutine.
- The cutter radius compensation is selected / deselected by the higher-level cycle; therefore, no G40, G41, G42 is programmed in the contour subroutine.

Explanation of the parameters

For the parameters _RTP, _RFP, _SDIS, see CYCLE81.

For the parameters _MID, _FAL, _FALD, _FFP1, _FFD, _DP, see POCKET3.

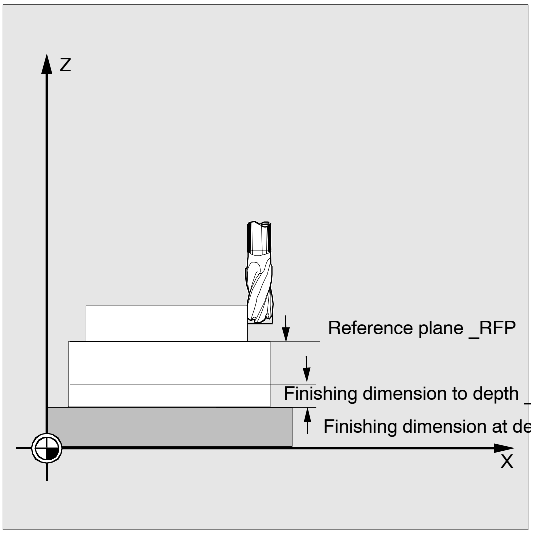


Fig. 9-40

_KNAME (name)

The contour to be milled is programmed completely in a subroutine. _KNAME defines the name of the contour subroutine.

1. The contour can be defined as a subroutine:
   _KNAME=name of subroutine

   The name of the contour subroutine is subject to all name conventions described in the Programming Guide.

   Input:

   – The subroutine already exists --> enter name, continue
   – The subroutine does not yet exist --> enter name and press softkey "**new file**". A program (main program) with the entered name is created and the program will jump to the contour editor.

   Use the "**Technol. mask**" softkey to confirm your input and return to the cycle help screenform.

2. The contour can also be a section of the calling program:
   _KNAME=name of starting label: name of end label

   Input:

   – Contour is already described --> name of starting label: Enter name of end label
   – Contour is not yet described --> enter name of starting label and press softkey "**contour append**".
     Starting and end label are automatically created from the name you have entered; then the program will jump to the contour editor.

   Use the "**Technol. mask**" softkey to confirm your input and return to the cycle help screenform.

Examples:

_KNAME=”KONTUR_1”	The milling contour is the complete program Kontur_1.
_KNAME=”ANFANG:ENDE”	The milling contour is defined as a section in the calling program, which starts from the block containing label ANFANG to the block containing label ENDE.

_LP1, _LP2 (length, radius)

Use parameter _LP1 to program the approach travel or approach radius (distance from the tool external edge to the contour starting point), and _LP2 to program the retraction travel or retraction radius (distance from the tool external edge to the contour end point).

The value _LP1, _LP2 must be >0. In the case of zero, error 61116 ”Approach or retraction path=0” is output.

Note

When using G40, the approach or retraction travel is the distance from the tool center point to the starting or end point of the contour.

_VARI (machining type)

Use the parameter _VARI to define the machining type. Possible values are:

UNIT DIGIT
Values: 1 roughing
2 finishing

TENS DIGIT
Values: 0 Intermediate path with G0
1 Intermediate path with G1

HUNDREDS DIGIT
Values: 0 Retraction at contour end to _RTP
1 Retraction at contour end to _RFP + _SDIS
2 Retraction at contour end by _SDIS
3 No retraction at the contour end

If a different value is programmed for the parameter _VARI, the cycle is aborted after output of alarm 61002 ”Machining type defined incorrectly”.

_RL (traveling around the contour)

With the parameter _RL, you will program traveling around the contour centrally, to the right or to the left with G40, G41 or G42. For possible values, see at ”Parameters for CYCLE72”.

_AS1, _AS2 (approach direction/path, retraction direction/retraction path)

Use the parameter _AS1 to program the specification of the approach path and _AS2 to program that of the retraction path. For possible values, see at ”Parameters for CYCLE72”. If _AS2 is not programmed, then the behavior of the retraction path is analogously to that of the approach path.

Smooth approach of the contour along a spatial path (helix or straight line) should only be programmed if the tool is not yet in mesh or is suitable for this type of approach.

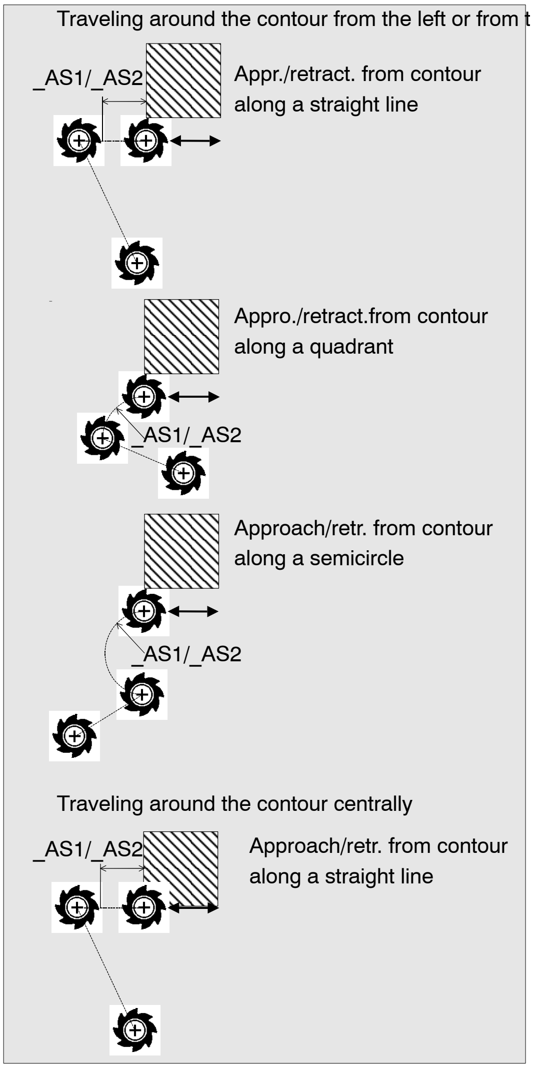


Fig. 9-41

In the case of central (G40) approach and retraction, only possible along a straight line.

_FF3 (retraction feedrate)

Use parameter _FF3 to define a retraction feedrate for intermediate positions in the plane (in the open) if the intermediate motions are to be carried out with feedrate (G01). If no feedrate value is programmed, the intermediate motions with G01 are carried out at surface feedrate.

Further notes

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 ”No tool compensation active” is output.

Programming example 1: Milling around a closed contour externally

This program is used to mill the contour shown in the diagram below.

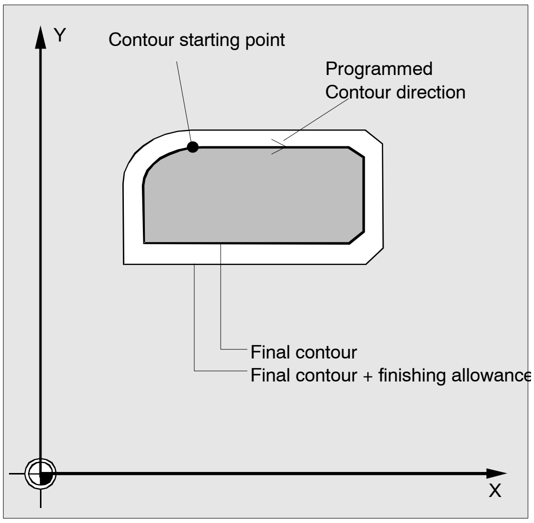


Fig. 9-42

Parameters for the cycle call:

- Retraction plane 250 mm
- Reference plane 200
- Safety clearance 3 mm
- Depth 175 mm
- Maximum depth infeed 10 mm
- Finishing allowance in the depth 1.5 mm
- Feedrate for depth infeed 400 mm/min
- Finishing allowance in the plane 1 mm
- Feedrate in the plane 800 mm/min
- Machining: Roughing up to finishing allowance; intermediate paths with G1, for intermediate paths retraction in Z to _RFP + _SDIS

Parameters for approach:

- G41 – left of the contour, i.e. external machining
- Approach and retraction along a quadrant in the plane 20 mm radius
- Retraction feedrate 1,000 mm/min

N10 T3 D1	T3: Milling cutter with radius 7
N20 S500 M3 F3000	Program feedrate, speed
N30 G17 G0 G90 X100 Y200 Z250 G94	Approach starting position
N40 CYCLE72("EX72CONTOUR", 250, 200, 3, 175, 10,1, 1.5, 800, 400, 111, 41, 2, 20, 1000, 2, 20)	Cycle call

N50 X100 Y200	
N60 M2	End of program

%_N_EX72CONTOUR_SPF	Subroutine for milling contour (for example)
N100 G1 G90 X150 Y160	Starting point of contour
N110 X230 CHF=10	
N120 Y80 CHF=10	
N130 X125	
N140 Y135	
N150 G2 X150 Y160 CR=25	
N160 M2	
N170 M02	

Programming example 2

Milling around a closed contour externally, as in programming example 1, with contour programming in the calling program

N10 T3 D1	T3: Milling cutter with radius 7
N20 S500 M3 F3000	Program feedrate, speed
N30 G17 G0 G90 X100 Y200 Z250 G94	Approach starting position
N40 CYCLE72 ("PIECE_245:PIECE_245_E", 250, 200, 3, 175, 10,1, 1.5, 800, 400, 11, 41, 2, 20, 1000, 2, 20)	Cycle call
N50 X100 Y200	
N60 M2	
N70 PIECE_245:	Contour
N80 G1 G90 X150 Y160	
N90 X230 CHF=10	
N100 Y80 CHF=10	
N110 X125	
N120 Y135	
N130 G2 X150 Y160 CR=25	
N140 PIECE_245_E:	End of contour
N150 M2	

9.6.4 Rectangular spigot milling – CYCLE76

Programming

CYCLE76 (_RTP, _RFP, _SDIS, _DP, _DPR, _LENG, _WID, _CRAD, _PA, _PO, _STA, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _AP1, _AP2)

Parameters

Table 9-17 Parameters for CYCLE76

_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (enter without sign)
_DP	real	Final drilling depth (absolute)
_DPR	real	Final drilling depth relative to the reference plane (enter without sign)
_LENG	real	Spigot length (enter without sign)
_WID	real	Spigot length (enter without sign)
_CARD	real	Spigot corner radius (enter without sign)
_PA	real	Reference point of spigot, abscissa (absolute)
_PO	real	Reference point of spigot, ordinate (absolute)
_STA	real	Angle between longitudinal axis and 1st axis of the plane
_MID	real	Maximum depth infeed (incremental; enter without sign)
_FAL	real	Final machining allowance at the margin contour (incremental)
_FALD	real	Finishing allowance at the base (incremental, enter without sign)
_FFP1	real	Feedrate at the contour
_FFD	real	Feedrate for depth infeed
_CDIR	integer	Milling direction (enter without sign) Values: 0 Climb milling 1 opposed milling 2 with G2 (independent of spindle direction) 3 with G3
_VARI	integer	Technology Values: 1 Roughing up to final machining allowance 2 Finishing (allowance X/Y/Z=0)
_AP1	real	Length of blank spigot

Function

Use this cycle to machine rectangular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always carried out in the position upstream of the semi-circle style approach to the contour.

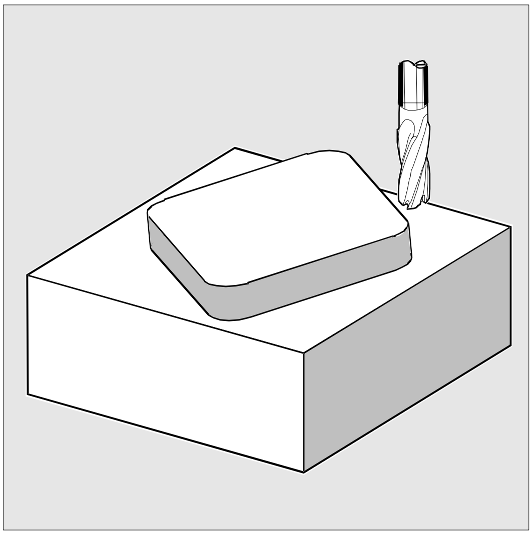


Fig. 9-43

Sequence

Position reached prior to cycle start:

The starting point is a position in the positive range of the abscissa with the approach semicircle and the programmed raw dimension on the abscissa end taken into account.

Sequence of motions when roughing (_VARI=1)
Approach to and retraction from the contour:

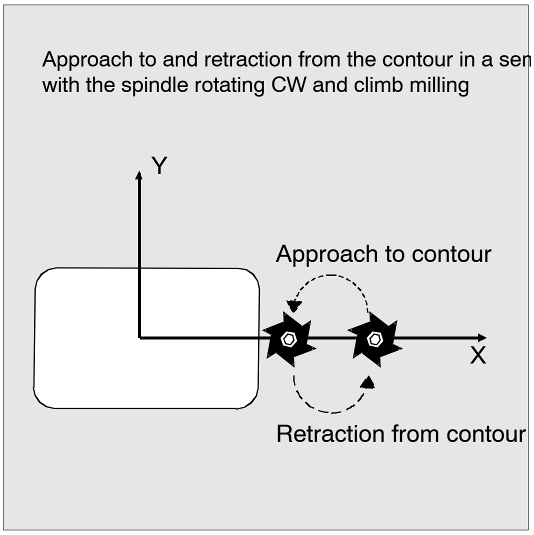


Fig. 9-44

The retraction plane (_RTP) is approached at rapid traverse rate to be able to position then at this height to the starting point in the machining plane. The starting point is defined with reference to 0 deg. of the abscissa.

The tool is feed to the safety clearance (_SDIS) at rapid traverse with subsequent traversing to the machining depth at feedrate. To approach the spigot contour, the tool will travel along a semicircle path.

The mill direction can be determined either as climb milling or as opposed (conventional) milling with reference to the spindle direction.

If the spigot is traveled around once, the contour is left along a semicircle in the plane, and the tool is fed to the next machining depth.
Then, the contour is approached along a semicircle again, and the spigot is traveled around once. This process is repeated until the programmed spigot depth is reached. Then, the retraction plane (_RTP) is approached at rapid traverse rate.

Depth infeed:

- Feeding to the safety clearance
- Insertion to machining depth

The first machining depth is calculated from:

- the total depth,
- the finishing allowance and
- the maximum possible depth infeed.

Sequence of motions when finishing (_VARI=2)

According to the set parameters _FAL and _FALD, either finishing is carried out at the surface contour or at the base or both together. The approach strategy corresponds to the motions in the plane as with roughing.

Explanation of the parameters

For the parameters _RTP, _RFP, _SDIS, _DP, _DPR, see CYCLE81.

For the parameters _MID, _FAL, _FALD, _FFP1, _FFD, see POCKET3.

_LENG, _WID and _CRAD (spigot length, spigot width and corner radius)

Use the parameters _LENG, _WID and _CRAD to define the form of a spigot in the plane.

The spigot is always dimensioned from the center. The amount of the length (_LENG) always refers to the abscissa (with a plane angle of 0 deg.).

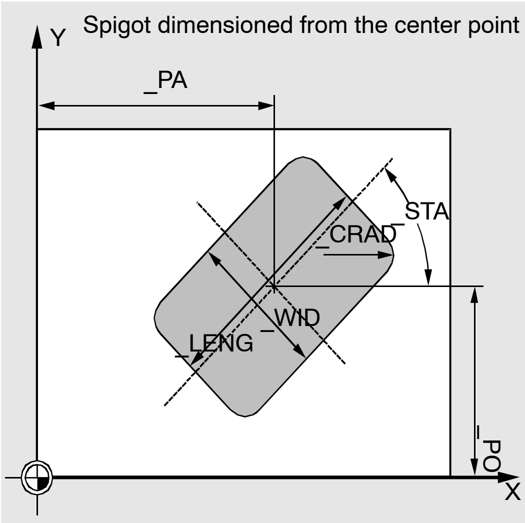


Fig. 9-45

_PA, _PO (reference point)

Use the parameters _PA and _PO to define the reference point of the spigot along the abscissa and the ordinate.

This is the spigot center point.

_STA (angle)

_STA specifies the angle between the 1st axis of the plane (abscissa) and the longitudinal axis of the spigot.

_CDIR (milling direction)

Use this parameter to specify the machining direction for the spigot.
Using the parameter _CDIR, the milling direction

- can be programmed directly with ”2 for G2” and ”3 for G3” or,
- alternatively, ”Synchronized operation” or ”Reverse rotation”.

synchronized operation or reverse rotation are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

Synchronized operation	Reverse rotation
M3 → G3	M3 → G2
M4 → G2	M4 → G3

_VARI (machining type)

Use the parameter _VARI to define the machining type.

Possible values are:

- 1=roughing
- 2=finishing

_AP1, _AP2 (blank dimensions)

When machining the spigot, it is possible to take into account blank dimensions (e.g. when machining precast parts).
The blank dimensions for length and width (_AP1 and _AP2) are programmed without sign and are placed by the cycle symmetrically around the pocket center point via calculation. The internally calculated radius of the approach semicircle depends on this dimension.

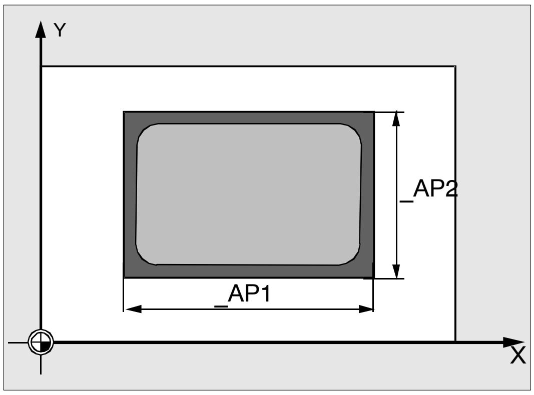


Fig. 9-46

Further notes

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61009 ”Active tool number=0” is output.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original c ordinate system is active again.

Programming example for a spigot

Use this program to machine a spigot in the XY plane which is 60 mm in length, 40 mm in width, and which has a corner radius of 15 mm and is mm in depth. The spigot has an angle of 10 degrees relative to the X axis and is premanufactured with a length allowance of 80 mm and a width allowance of 50 mm.

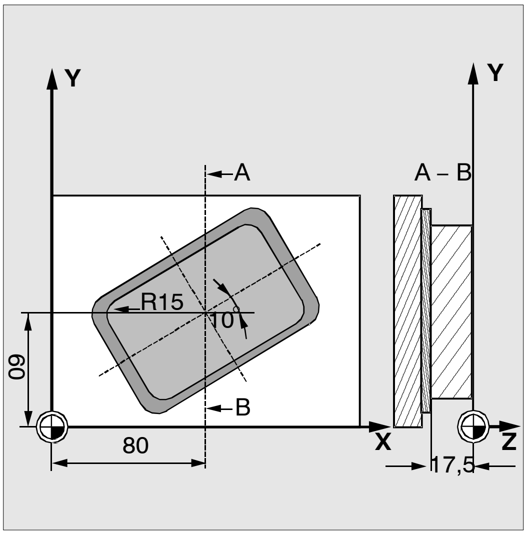


Fig. 9-47

N10 G90 G0 G17 X100 Y100 T20 D1 S3000 M3	Specification of the technological values
N11 M6	
N30 CYCLE76 (10, 0, 2, –17.5, , –60, –40, 15, 80, 60, 10, 11, , , 900, 800, 0, 1, 80, 50)	Cycle call
N40 M30	End of program

9.6.5 Circular spigot milling – CYCLE77

Programming

CYCLE77 (_RTP, _RFP, _SDIS, _DP, _DPR, _PRAD, _PA, _PO, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _AP1)

Parameters

The following input parameters are always required:

Table 9-18 Parameters for CYCLE77

_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (enter without sign)
_DP	real	Depth (absolute)
_DPR	real	Depth relative to the reference plane (enter without sign)

Table 9-18 Parameters for CYCLE77

_PRAD	real	Spigot diameter (enter without sign)
_PA	real	Center point of spigot, abscissa (absolute)
_PO	real	Center point of spigot, ordinate (absolute)
_MID	real	Maximum depth infeed (incremental; enter without sign)
_FAL	real	Final machining allowance at the margin contour (incremental)
_FALD	real	Finishing allowance at the base (incremental, enter without sign)
_FFP1	real	Feedrate at the contour
_FFD	real	Feedrate for depth infeed (or spatial infeed)
_CDIR	integer	Milling direction (enter without sign) Values: 0 Climb milling 1 opposed milling 2 with G2 (independent of spindle direction) 3 with G3
_VARI	integer	Technology Values: 1 Roughing up to final machining allowance 2 Finishing (allowance X/Y/Z=0)
_AP1	real	Length of blank spigot

Function

Use this cycle to machine circular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always carried out in the position upstream of the semicircle style approach to the contour.

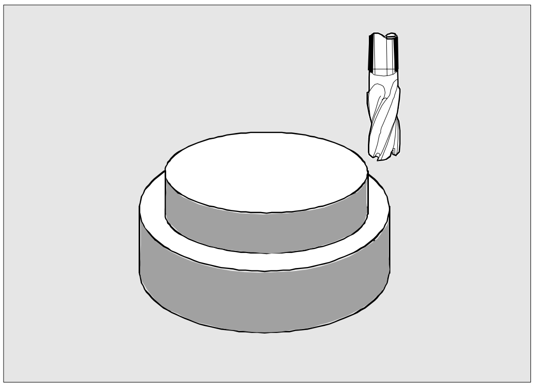


Fig. 9-48

Sequence

Position reached prior to cycle start:

The starting point is a position in the positive range of the abscissa with the approach semicircle and the programmed raw dimension taken into account.

Sequence of motions when roughing (_VARI=1)

Approach to and retraction from the contour:

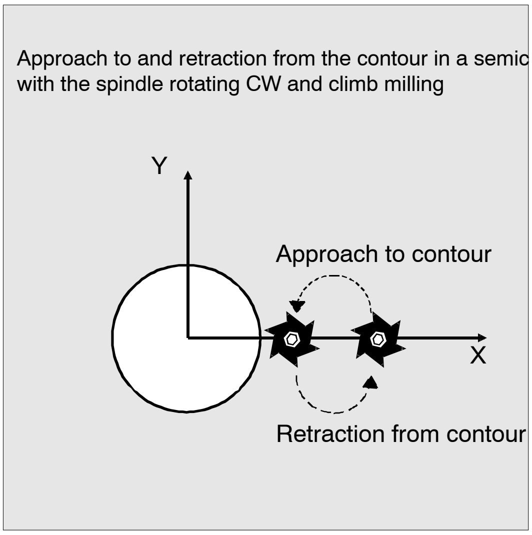


Fig. 9-49

The retraction plane (_RTP) is approached at rapid traverse rate to be able to position then at this height to the starting point in the machining plane. The starting point is defined with reference to 0 deg. of the axis of the abscissa.

The tool is feed to the safety clearance (_SDIS) at rapid traverse with subsequent traversing to the machining depth at feedrate. To approach the spigot contour, the tool is approached along a semicircle path, taking into account the programmed blank spigot.

The mill direction can be determined either as climb milling or as opposed (conventional) milling with reference to the spindle direction.

If the spigot is traveled around once, the contour is left along a semicircle in the plane, and the tool is fed to the next machining depth.
Then, the contour is approached along a semicircle again, and the spigot is traveled around once. This process is repeated until the programmed spigot depth is reached.

Then, the retraction plane (_RTP) is approached at rapid traverse rate.

Depth infeed:

- Feeding to the safety clearance
- Insertion to machining depth

The first machining depth is calculated from:

- the total depth,
- the finishing allowance and
- the maximum possible depth infeed.

Sequence of motions when finishing (_VARI=2)

According to the set parameters _FAL and _FALD, either finishing is carried out at the surface contour or at the base or both together. The approach strategy corresponds to the motions in the plane as with roughing.

Explanation of the parameters

For the parameters _RTP, _RFP, _SDIS, _DP, _DPR, see CYCLE81.

For the parameters _MID, _FAL, _FALD, _FFP1, _FFD, see POCKET3.

_PRAD (spigot diameter)

Enter the diameter without sign.

_PA, _PO (spigot center)

Use the parameters _PA and _PO to define the reference point of the spigot.

_CDIR (milling direction)

Use this parameter to specify the machining direction for the spigot. By using the parameter _CDIR, the milling direction

- can be programmed directly with ”2 for G2” and ”3 for G3” or,
- alternatively, ”Synchronized operation” or ”Reverse rotation”.

Synchronized operation or reverse rotation are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

Synchronized operation	Reverse rotation
M3 → G3	M3 → G2
M4 → G2	M4 → G3

_VARI (machining type)

Use the parameter _VARI to define the machining type. Possible values are:

- 1=roughing
- 2=finishing

_AP1 (diameter of the blank spigot)

Use this parameter to define the blank dimension of the spigot (without sign). The internally calculated radius of the approach semicircle depends on this dimension.

Further notes

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61009 ”Active tool number=0” is output.
Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original coordinate system is active again.

Programming example for a circular spigot

Machining a spigot from a blank with a diameter of 55 mm and a maximum infeed of 10 mm per cut; specification of a final machining allowance for subsequent finishing of the spigot surface. The whole machining is performed with reverse rotation.

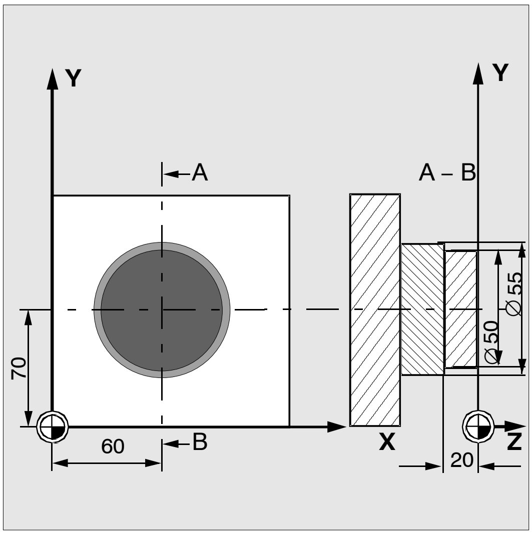


Fig. 9-50

N10 G90 G17 G0 S1800 M3 D1 T1	Specification of the technological values
N11 M6	
N20 CYCLE77 (10, 0, 3, –20, ,50, 60, 70, 10, 0.5, 0, 900, 800, 1, 1, 55)	Calling the roughing cycle
N30 D1 T2 M6	Tool change
N40 S2400 M3	Specification of the technological values
N50 CYCLE77 (10, 0, 3, –20, , 50, 60, 70, 10, 0, 0, 800, 800, 1, 2, 55)	Calling the finishing cycle
N40 M30	End of program

9.6.6 Slots on a circle – LONGHOLE

Programming

LONGHOLE (RTP, RFP, SDIS, DP, DPR, NUM, LENG, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID)

Parameters

Table 9-19 LONGHOLE parameters

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

Table 9-19 LONGHOLE parameters, cont'd

NUM	integer	Number of slots
LENG	real	Slot length (enter without sign)
CPA	real	Center point of the circle (absolute), 1st axis of the plane
CPO	real	Center point of the circle (absolute), 2nd axis of the plane
RAD	real	Radius of the circle (enter without sign)
STA1	real	Starting angle
INDA	real	Incrementing angle
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for one infeed (enter without sign)

Important

The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

Function

Use this cycle to machine slots arranged on a circle. The longitudinal axis of the slots is aligned radially.

Contrary to the groove, the width of the slot is determined by the tool diameter.

Internally in the cycle, an optimum traversing path of the tool is determined, ruling out unnecessary idle passes. If several depth infeeds are required to machine a slot, the infeed is carried out alternately at the end points. The path to be traversed along the longitudinal axis of the slot will change its direction after each infeed. The cycle will search for the shortest path when changing to the next slot.

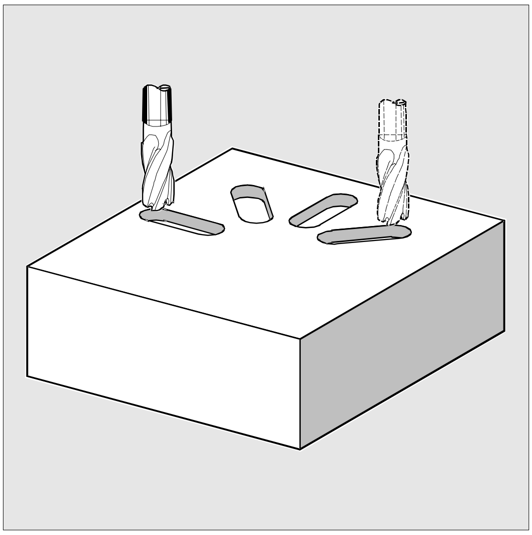


Fig. 9-51

Sequence

Position reached prior to cycle start:

Initial position is any position from which each of the elongated holes can be approached without collision.

The cycle creates the following sequence of motions:

- Using G0, the starting position for the cycle is approached. In both axes of the current plane, the next end point of the first slot to be machined is approached at the height of the retraction plane in this applicate, and then the applicate is lowered to the reference plane brought forward by the safety clearance.
- Each slot is milled in a reciprocating motion. The machining in the plane is performed using G1 and the feedrate programmed under FFP1. The infeed to the next machining depth calculated using G1 internally in the cycle and using feedrate is performed at each reversal point until the final depth is reached.
- Retraction to the retraction plane using G0 and approach to the next slot on the shortest path.
- After the last slot has been machined, the tool is moved with G0 to the position in the machining plane, which was reached last and which is specified in the diagram below, and the cycle is ended.

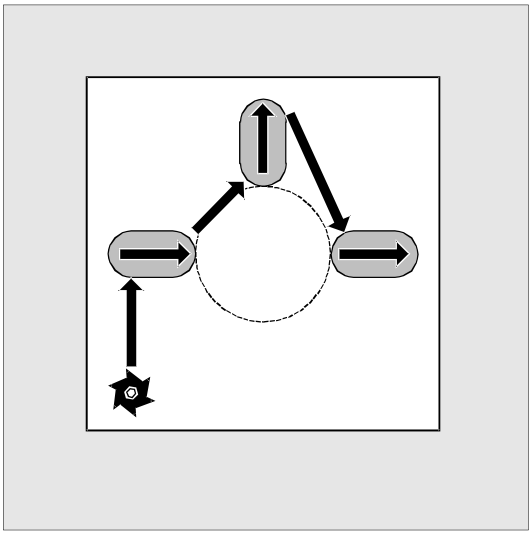


Fig. 9-52

Explanation of the parameters

For the parameters RTP, RFP, SDIS, see CYCLE81.

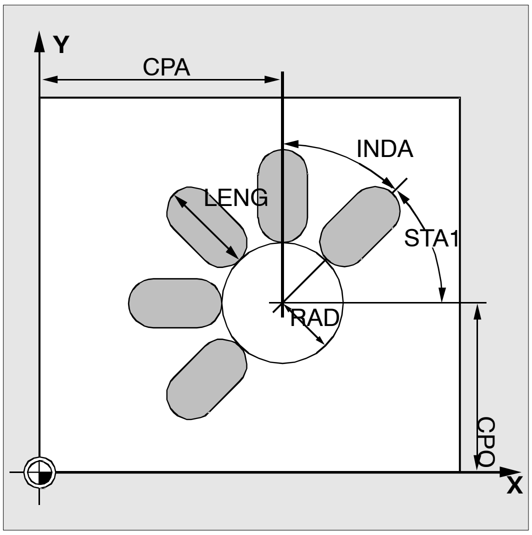


Fig. 9-53

DP and DPR (elongated hole final drilling depth)

The slot depth can be specified either absolute (DP) or relative (DPR) to the reference plane.

With relative specification, the cycle will calculate the resulting depth automatically using the positions of reference and retraction planes.

NUM (number)

Use the parameter NUM to specify the number of slots.

LENG (elongated hole final drilling depth)

The length of the slot is programmed under LENG.

If the cycle detects that this length is less than the milling diameter, the cycle is aborted with alarm 61105 "Cutter radius too large".

MID (infeed depth)

Use this parameter to define the maximum infeed depth.

The depth infeed is performed by the cycle in equally sized infeed steps.

Using MID and the total depth, the cycle automatically calculates this infeed which lies between 0.5 x maximum infeed depth and the maximum infeed depth. The minimum possible number of infeed steps is used as the basis. MID=0 means that the cut to pocket depth is made with one feed.

The depth infeed starts from the reference plane brought forward by the safety clearance (depending on _ZSD[1]).

FFD and FFP1 (feedrate for depth and surface)

The feedrate FFP1 is active for all movements in the plane traversed at feedrate when roughing. FFD acts for infeeds vertically to this plane.

CPA, CPO and RAD (center point and radius)

The position of the circle of holes in the machining plane is defined by the center point (CPA, CPO) and the radius (RAD). Only positive values are permitted for the radius.

STA1 and INDA (starting and incremental angle)

The arrangement of the slots on the circle is defined by these parameters.

If INDA=0, the incrementing angle is calculated from the number of slots so that they are arranged equally around the circle.

Further notes

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

If mutual contour violations of the slots result from incorrect values of the parameters that determine the arrangement and the size of the slots, the cycle will not start the machining. The cycle is aborted and the error message
61104 "Contour violation of grooves/slots" is output.

During the cycle, the workpiece coordinate system is offset and rotated. The values in the workpiece coordinate system (WCS) displayed on the actual value display are such that the longitudinal axis of the slot that has just been machined corresponds to the 1st axis of the current machining plane.

After the cycle has been completed, the workpiece coordinate system is in the same position again as it was before the cycle was called.

Programming example: Machining slots

By using this program, you can machine 4 slots of the length 30 mm and the relative depth 23 mm (difference between the reference plane and the slot root), which are arranged on a circle with the center point Z45 Y40 and the radius 20 mm in the YZ plane. The starting angle is 45 degrees, the advancing angle is 90 degrees. The maximum infeed depth is 6 mm, the safety clearance 1 mm.

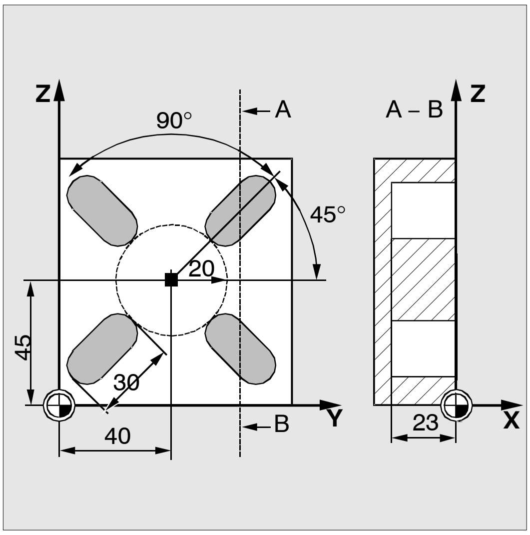


Fig. 9-54

N10 G19 G90 D9 T10 S600 M3	Specification of the technological values
N20 G0 Y50 Z25 X5	Approach starting point
N30 LONGHOLE (5, 0, 1, , 23, 4, 30, 40, 45, 20, 45, 90, 100 , 320, 6)	Cycle call
N40 M02	End of program

9.6.7 Slots on a circle – SLOT1

Programming

SLOT1(RTP, RFP, SDIS, DP, DPR, NUM, LENG, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF)

Parameters

Table 9-20 Parameters for SLOT1

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)

Table 9-20 Parameters for SLOT1, cont'd

DP	real	Groove depth (absolute)
DPR	real	Slot depth relative to the reference plane (enter without sign)
NUM	integer	Number of slots
LENG	real	Slot length (enter without sign)
WID	real	Slot width (enter without sign)
CPA	real	Center point of the circle (absolute), 1st axis of the plane
CPO	real	Center point of the circle (absolute), 2nd axis of the plane
RAD	real	Radius of the circle (enter without sign)
STA1	real	Starting angle
INDA	real	Incrementing angle
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for one infeed (enter without sign)
CDIR	integer	Milling direction for machining the slot Values: 2 (for G2) 3 (for G3)
FAL	real	Finishing allowance at the slot edge (enter without sign)
VARI	integer	Machining type Values: 0=complete machining 1=roughing 2=finishing
MIDF	real	Maximum infeed depth for finishing
FFP2	real	Feedrate for finishing
SSF	real	Speed when finishing

Note

The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

Function

The cycle SLOT1 is a combined roughing–finishing cycle.

Use this cycle to machine slots arranged on a circle. The longitudinal axis of the slots is aligned radially. Unlike the elongated hole, a value is defined for the slot width.

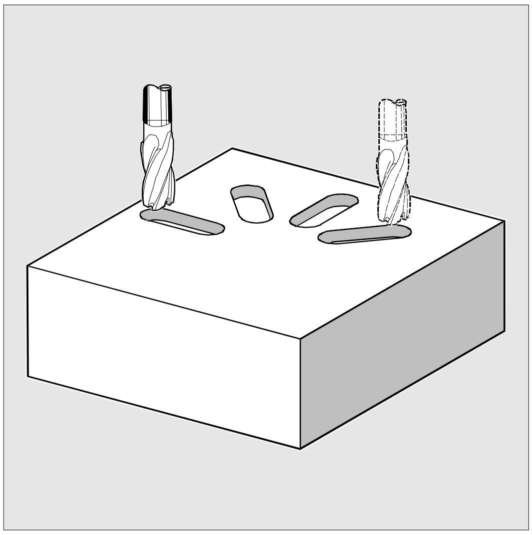


Fig. 9-55

Sequence

Position reached prior to cycle start:

The starting position can be any position from which each of the slots can be approached without collision.

The cycle creates the following sequence of motions:

- Approach of the position specified in the diagram 9-56 on the right with G0 at the beginning of the cycle
- Complete machining of a slot is carried out in the following steps:
  - Approach of the reference plane brought forward by the safety clearance by using G0
  - Infeed to the next machining depth with G1 and with feedrate value FFD
  - Solid machining of the slot to the finishing allowance at the slot edge with feedrate value FFP1. Then finishing with feedrate value FFP2 and spindle speed SSF along the contour according to the machining direction programmed under CDIR.
  - The depth infeed is always carried out at the same position in the machining plane until the end depth of the slot is reached.
- Retract tool to the retraction plane and move to the next slot with G0.
- After the last slot has been machined, the tool is moved with G0 to the end position in the machining plane, which is specified in the diagram below, and the cycle is ended.

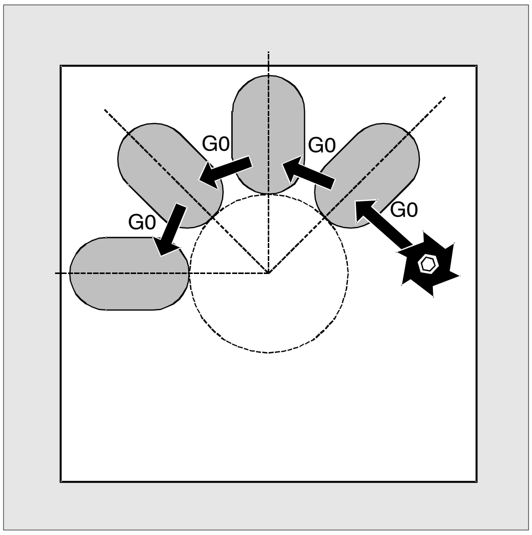


Fig. 9-56

Explanation of the parameters

For the parameters RTP, RFP, SDIS, see CYCLE81.

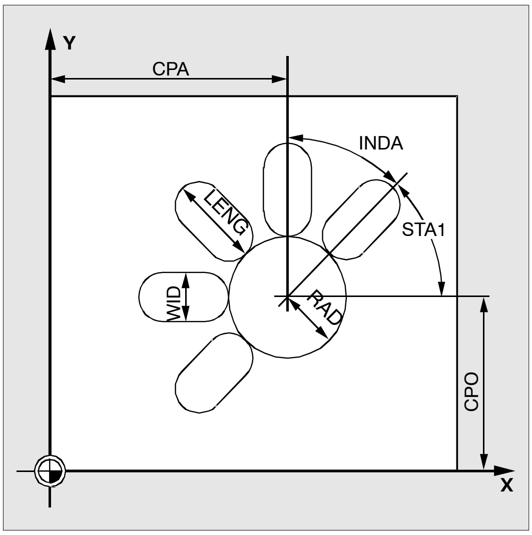


Fig. 9-57

DP and DPR (slot depth)

The slot depth can be specified either absolute (DP) or relative (DPR) to the reference plane.

With relative specification, the cycle will calculate the resulting depth automatically using the positions of reference and retraction planes.

NUM (number)

Use the parameter NUM to specify the number of slots.

LENG and WID (slot length and slot width)

Use the parameters LENG and WID to define the form of a slot in the plane. The milling cutter diameter must be smaller than the slot width. Otherwise, alarm 61105 ”Cutter radius too large” will be activated and the cycle aborted.

The milling cutter diameter must not be smaller than half of the slot width. This is not checked.

CPA, CPO and RAD (center point and radius)

The position of the circle of holes in the machining plane is defined by the center point (CPA, CPO) and the radius (RAD). Only positive values are permitted for the radius.

STA1 and INDA (starting and incremental angle)

The arrangement of the slot on the circle is defined by these parameters.

STA1 specifies the angle between the positive direction of the 1st axis (abscissa) in the workpiece coordinate system active before the cycle was called and the first slot. Parameter INDA contains the angle from one slot to the next.

If INDA=0, the incrementing angle is calculated from the number of slots so that they are arranged equally around the circle.

FFD and FFP1 (feedrate for depth and surface)

The feedrate FFD is active for all infeed movements perpendicular to the machining plane.

The feedrate FFP1 is active for all movements in the plane traversed at feedrate when roughing.

MID (infeed depth)

Use this parameter to define the maximum infeed depth.

The depth infeed is performed by the cycle in equally sized infeed steps.

Using MID and the total depth, the cycle automatically calculates this infeed which lies between 0.5 x maximum infeed depth and the maximum infeed depth. The minimum possible number of infeed steps is used as the basis. MID=0 means that the cut to slot depth is made with one feed.

The depth infeed commences at the reference plane moved forward by the safety clearance.

CDIR (milling direction)

Use this parameter to specify the machining direction for the slot. Possible values are:

- ”2” for G2
- ”3” for G3

If the parameter is set to an illegal value, then the message ”Wrong milling direction, G3 will be generated” will be displayed in the dialog line. In this case, the cycle is continued and G3 is automatically generated.

FAL (finishing allowance)

Use this parameter to program a finishing allowance at the slot edge. FAL does not influence the depth infeed.

If the value of FAL is greater than allowed for the specified width and the milling cutter used, FAL is automatically reduced to the maximum possible value. In the case of roughing, milling is performed with a reciprocating movement and depth infeed at both end points of the slot.

VARI, MIDF, FFP2 and SSF (machining type, infeed depth, feedrate and speed)

Use the parameter VARI to define the machining type.

Possible values are:

- 0=complete machining in two parts
  - Solid machining of the slot (SLOT1, SLOT2) to the finishing allowance is performed at the spindle speed programmed before the cycle was called and with feedrate FFP1. Depth infeed is defined with MID.
  - Solid machining of the remaining machining allowance is carried out at the spindle speed defined via SSF and the feedrate FFP2. Depth infeed is defined with MIDF.
If MIDF=0, the infeed is equal to the final depth.
  - If FFP2 is not programmed, feedrate FFP1 is active. This also applies analogously if SSF is not specified, i.e. the speed programmed prior to the cycle call will apply.
- 1=roughing
The slot (SLOT1, SLOT2) is solid–machined up to the finishing allowance at the speed programmed before the cycle call and at the feedrate FFP1. The depth infeed is programmed via MID.
- 2=finishing
The cycle requires that the slot (SLOT1, SLOT2) is already machined to a residual finishing allowance and that it is only necessary to machine the final finishing allowance. If FFP2 and SSF are not programmed, the feedrate FFP1 or the speed programmed before the cycle call is active. Depth infeed is defined with MIDF.

If a different value is programmed for the parameter VARI, the cycle is aborted after output of alarm 61102 ”Machining type defined incorrectly”.

Further notes

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 ”No tool compensation active” is output.

If incorrect values are assigned to the parameters that determine the arrangement and size of the slots and thus cause mutual contour violation of the slots, the cycle is not started. The cycle is canceled, and the error message 61104 ”Contour violation of slots/elongated holes” is output.

During the cycle, the workpiece coordinate system is offset and rotated. The values in the workpiece coordinate system displayed on the actual value display are such that the longitudinal axis of the slot that has just been machined corresponds to the 1st axis of the current machining plane.

After the cycle has been completed, the workpiece coordinate system is in the same position again as it was before the cycle was called.

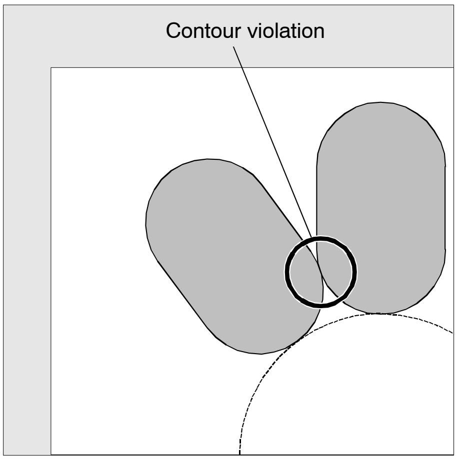


Fig. 9-58

Programming example: Slots

4 slots are milled.

The slots have the following dimensions: Length 30 mm, width 15 mm and depth 23 mm. The safety clearance is 1 mm, the finishing allowance 0.5 mm, the milling direction is G2, and the maximum infeed in the depth is 6 mm.

The slot is to be machined completely. infeed during finishing is to be performed directly to the pocket depth and the same feedrate and speed are to be used.

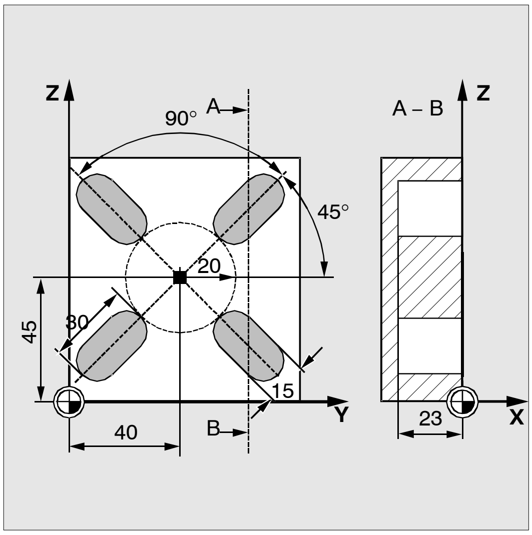


Fig. 9-59

N10 G17 G90 T1 D1 S600 M3	Specification of the technological values
N20 G0 X20 Y50 Z5	Approach starting position
N30 SLOT1(5, 0, 1, −23, , 4, 30, 15, 40, 45, 20, 45, 90, 100, 320, 6, 2, 0.5, 0, , 0,)	Cycle call, parameters VARI, MIDF, FFP2 and SSF omitted
N40 M02	End of program

9.6.8 Circumferential slot – SLOT2

Programming

SLOT2(RTP, RFP, SDIS, DP, DPR, NUM, AFSL, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF)

Parameters

Table 9-21 Parameters for SLOT2

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Groove depth (absolute)
DPR	real	Slot depth relative to the reference plane (enter without sign)
NUM	integer	Number of slots
AFSL	real	Angle for the slot length (enter without sign)
WID	real	Circumferential slot width (enter without sign)
CPA	real	Center point of the circle (absolute), 1st axis of the plane
CPO	real	Center point of the circle (absolute), 2nd axis of the plane
RAD	real	Radius of the circle (enter without sign)
STA1	real	Starting angle
INDA	real	Incrementing angle
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for one infeed (enter without sign)
CDIR	integer	Milling direction for machining the circumferential slot Values: 2 (for G2) 3 (for G3)
FAL	real	Finishing allowance at the slot edge (enter without sign)
VARI	integer	Machining type Values: 0=complete machining 1=roughing 2=finishing
MIDF	real	Maximum infeed depth for finishing
FFP2	real	Feedrate for finishing
SSF	real	Speed when finishing

Note

The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

Function

The cycle SLOT2 is a combined roughing–finishing cycle.

Use this cycle to machine circumferential slots arranged on a circle.

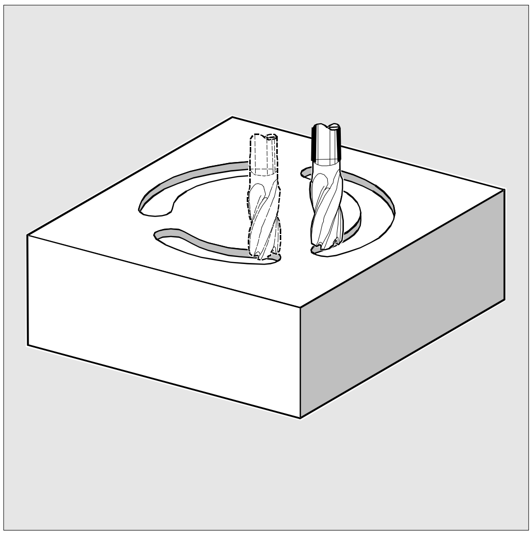


Fig. 9-60

Sequence

Position reached prior to cycle start:

The starting position can be any position from which each of the slots can be approached without collision.

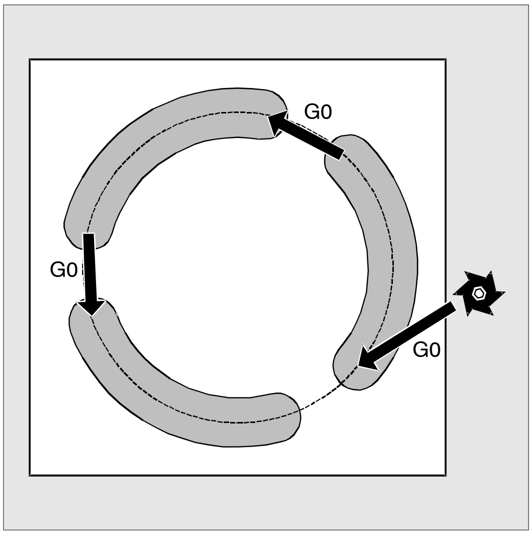


Fig. 9-61

The cycle creates the following sequence of motions:

- G0 is used to approach the position specified in the diagram below at cycle start.
- The steps when machining a circumferential slot are the same as when machining an elongated hole.
- After a circumferential slot is machine completely, the tool is retracted to the retraction plane and the next slot is machined with G0.
- After the last slot has been machined, the tool is moved with G0 to the end position in the machining plane, which is specified in the diagram below, and the cycle is ended.

Explanation of the parameters

For the parameters RTP, RFP, SDIS, see CYCLE81.

For the parameters DP, DPR, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF, see SLOT1.

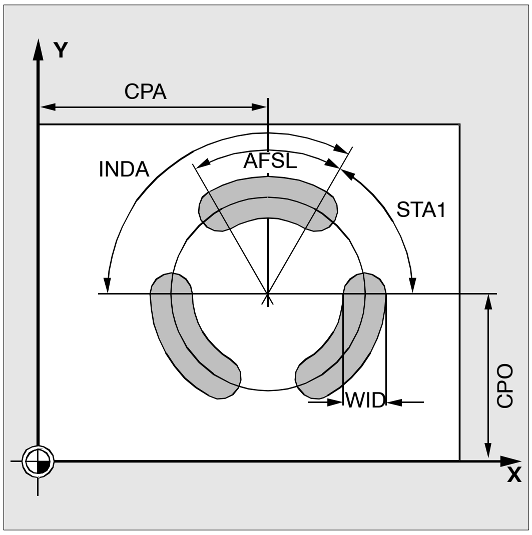


Fig. 9-62

NUM (number)

Use the parameter NUM to specify the number of slots.

AFSL and WID (angle and circumferential slot width)

Use the parameters AFSL and WID to define the form of a slot in the plane. The cycle checks whether the slot width is violated with the active tool. Otherwise, alarm 61105 ”Cutter radius too large” will be activated and the cycle aborted.

CPA, CPO and RAD (center point and radius)

The position of the circle of holes in the machining plane is defined by the center point (CPA, CPO) and the radius (RAD). Only positive values are permitted for the radius.

STA1 and INDA (starting and incremental angle)

The arrangement of the circumferential slots on the circle is defined by these parameters.

STA1 specifies the angle between the positive direction of the 1st axis of the plane in the workpiece coordinate system active before the cycle was called and the first circumferential slot.

Parameter INDA contains the angle from one circumferential slot to the next.

If INDA=0, the incrementing angle is calculated from the number of circumferential slots so that they are arranged equally around the circle.

Further notes

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 ”No tool compensation active” is output.

If incorrect values are assigned to the parameters that determine the arrangement and size of the slots and thus cause mutual contour violation of the slots, the cycle is not started.

The cycle is canceled, and the error message 61104 ”Contour violation of slots/elongated holes” is output.

During the cycle, the workpiece coordinate system is offset and rotated. The values in the workpiece coordinate system displayed on the actual value display are always such that the circumferential slot just machined starts at the 1st axis of the current machining plane and the zero point of the WCS is located in the center point of the circle.

After the cycle has been completed, the workpiece coordinate system is in the same position again as it was before the cycle was called.

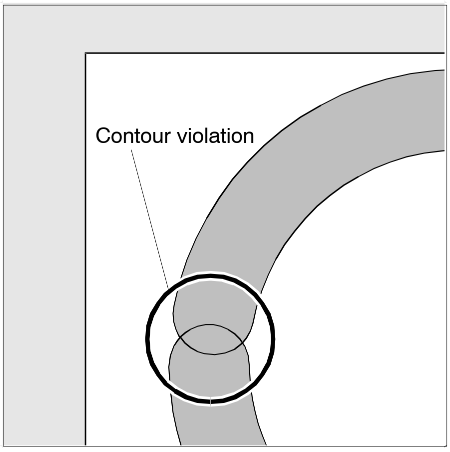


Fig. 9-63

Programming example: Slots2

Use this program to machine 3 circumferential slots arranged at a circle with center point X60 Y60 and radius 42 mm in the XY plane. The circumferential slots have the following dimensions: Width 15 mm, angle for slot length 70 degrees, depth 23 mm. The starting angle is 0 degrees, and the incrementing angle is 120 degrees. A finishing allowance of 0.5 mm is taken into account at the contour of the slots, the safety clearance in the infeed axis Z is 2 mm, and the maximum depth infeed is 6 mm. The slots are to be machined completely. Speed and feedrate are to be the same when finishing. The infeed when finishing is to be performed to slot depth.

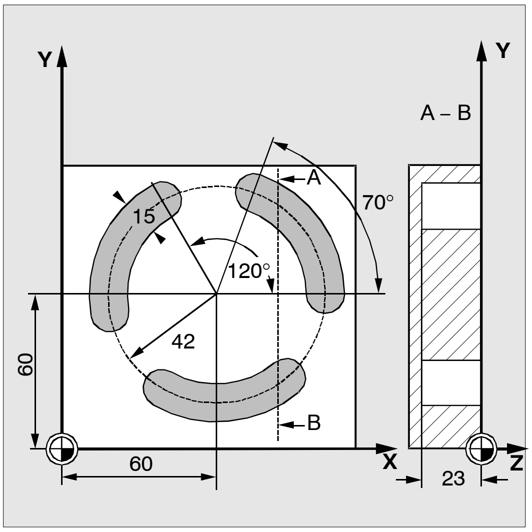


Fig. 9-64

N10 G17 G90 T1 D1 S600 M3	Specification of the technological values
N20 G0 X60 Y60 Z5	Approach starting position
N30 SLOT2(2, 0, 2, –23, , 3, 70, 15, 60, 60, 42, , 120, 100, 300, 6, 2, 0.5, 0, , 0,)	Cycle call Reference plane+SDIS=retraction plane means: Lowering in the infeed axis with G0 to reference plane+SDIS no longer applicable, parameters VAR, MIDF, FFP2 and SSF omitted
N40 M02	End of program

9.6.9 Milling a rectangular pocket – POCKET3

Programming

POCKET3(_RTP, _RFP, _SDIS, _DP, _LENG, _WID, _CRAD, _PA, _PO, _STA, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _MIDA, _AP1, _AP2, _AD, _RAD1, _DP1)

Parameters

Table 9-22 Parameters for POCKET3

_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (enter without sign)
_DP	real	Pocket depth (absolute)
_LENG	real	Pocket length, for dimensioning from the corner with sign
_WID	real	Pocket width, for dimensioning from the corner with sign
_CRAD	real	Pocket corner radius (enter without sign)
_PA	real	Pocket reference point (absolute), 1st axis of the plane
_PO	real	Pocket reference point (absolute), 2nd axis of the plane
_STA	real	Angle between the pocket longitudinal axis and the 1st axis of the plane (enter without sign) Range of values: 0° ≤ _STA < 180°
_MID	real	Maximum infeed depth (enter without sign)
_FAL	real	Finishing allowance at the pocket edge (enter without sign)
_FALD	real	Finishing allowance at the base (enter without sign)
_FFP1	real	Feedrate for surface machining
_FFD	real	Feedrate for depth infeed
_CDIR	integer	Milling direction: (enter without sign) Values: 0 climb milling (as spindle direction) 1 opposed milling 2 with G2 (independent of spindle direction) 3 with G3
_VARI	integer	Machining type UNITS DIGIT Values: 1 roughing 2 finishing TENS DIGIT Values: 0 perpendicular to pocket center with G0 1 perpendicular to pocket center with G1 2 along a helix 3 oscillating along the pocket longitudinal axis

The other parameters can be selected as options. Specify the plunge–cut strategy and the overlap for solid machining (to be entered without sign):

_MIDA	real	Maximum infeed width as a value in solid machining in the plane
_AP1	real	Blank dimension of pocket length
_AP2	real	Blank dimension of pocket width
_AD	real	Blank pocket depth dimension from reference plane
_RAD1	real	Radius of the helical path on insertion (relative to the tool center point path) or maximum insertion angle for oscillating motion
_DP1	real	Insertion depth per 360° revolution on insertion along helical path

Function

The cycle can be sued for roughing and finishing. For finishing, a face cutter is required.

The depth infeed is always started from the pocket center point and performed there vertically; thus predrilling can be suitably performed at this position.

- The milling direction can be determined either using a G command (G2/G3) or from the spindle direction as synchronous or opposed milling.
- For solid machining, the maximum infeed width in the plane can be programmed.
- Finishing allowance also for the pocket base
- There are three different insertion strategies:
  - vertically to the pocket center
  - along a helical path around the pocket center
  - oscillating at the pocket central axis
- Shorter approach paths in the plane for finishing
- Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).

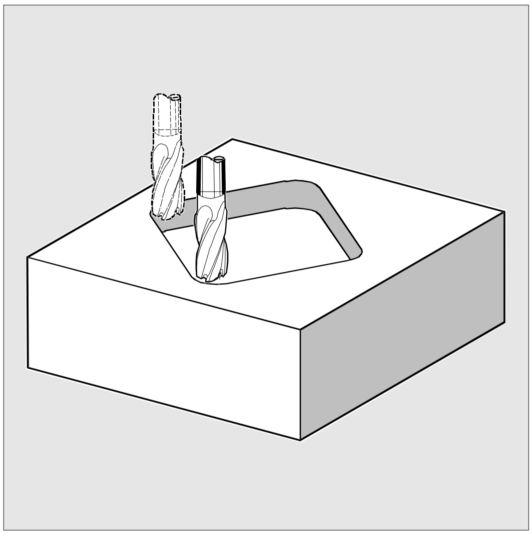


Fig. 9-65

Sequence

Position reached prior to cycle start:

Starting position is any position from which the pocket center point can be approached at the height of the retraction plane without collision.

Sequence of motions when roughing:

With G0, the pocket center point is approached at the retraction level, and then, from this position, with G0, too, the reference plane brought forward by the safety clearance is approached. The machining of the pocket is then carried out according to the selected insertion strategy, taking into account the programmed blank dimensions.

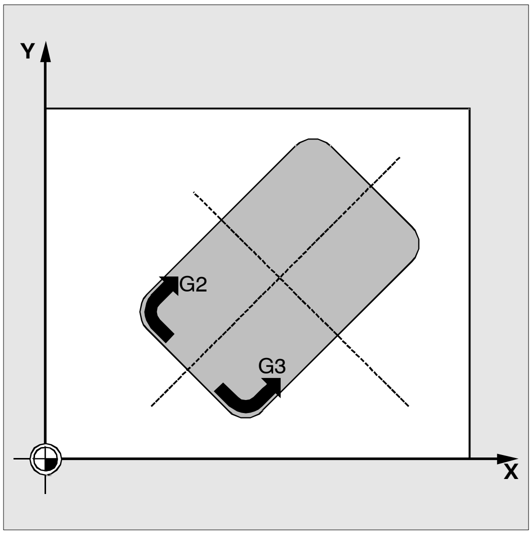


Fig. 9-66

Sequence of motions when finishing

Finishing is performed in the order from the edge until the finishing allowance on the base is reached, and then the base is finished. If one of the finishing allowances is equal to zero, this part of the finishing process is skipped.

- Finishing on the edge

  While finishing on the edge, the tool will traverse around the pocket contour only once.

  For finishing on the edge, the path includes one quadrant reaching the corner radius. The radius of this path is normally 2 mm or, if "less space" is provided, equals to the difference between the corner radius and the mill radius.
  If the finishing allowance on the edge is larger than 2 mm, the approach radius is increased accordingly.

  The depth infeed is performed with G0 in the open towards the pocket center, and the starting point of the approach path is also reached with G0.

- Finishing on the base

  During finishing on the base, the machine performs G0 towards the pocket center until reaching a distance equal to pocket depth + finishing allowance + safety clearance. From this point onwards, the tool is always fed in **vertically** at the depth (since a tool with a front cutting edge is used for base finishing).

  The base surface of the pocket is machined once.

Insertion strategies:

- Inserting vertically to the pocket center means that the current infeed depth calculated internally in the cycle (≤ maximum infeed depth programmed under _MID) is executed in a block containing G0 or G1.

- Insertion at a helical path means that the cutter center point traverses along the helical path determined by the radius _RAD1 and the depth per revolution _DP1. The feedrate is also programmed under _FFD. The direction of rotation of this helical path corresponds to the direction of rotation with which the pocket will be machined.

  The insertion depth programmed under _DP1 is taken into account as the maximum depth and is always calculated as an integer number of revolutions of the helical path.

  If the current depth required for an infeed (this can be several revolutions on the helical path) is reached, a full circle is still executed to eliminate the inclined path of insertion. Then solid machining of the pocket starts on this plane until finishing allowance.

  The starting point of the described helical path is at the longitudinal axis of the pocket in "plus direction" and is approached with G1.

- Insertion with oscillation to the central axis of the pocket means that the cutter center point inserts oscillating an a straight line to and fro until it has reached the next current depth. The maximum insertion angle is programmed under _RAD1, and the length of the oscillation travel is calculated in the cycle. If the current depth is reached, the travel is executed once more without depth infeed in order to eliminate the inclined insertion path. The feedrate is programmed under _FFD.

Taking into account the blank dimensions

During solid machining of the pockets, it is possible to take into account blank dimensions (e.g. when machining precast parts).

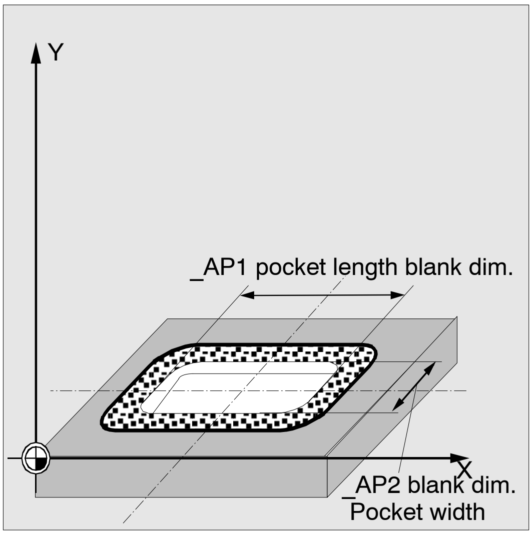


Fig. 9-67

The blank dimensions for length and width (_AP1 and _AP2) are programmed without sign and are placed by the cycle symmetrically around the pocket center point via calculation. You will define the part of the pocket which is no longer to be machined by solid machining. The blank dimension for the depth (_AD) is also programmed without sign and taken into account by the reference plane in the direction of the pocket depth.

The depth infeed when taking into account blank dimensions is carried out according to the programmed type (helical path, oscillating, vertically). If the cycle detects that there is space enough in the pocket center because of the given blank contour and the radius of the active tool, the infeed is carried out vertically to the pocket center point as long as it is possible in order not to traverse extensive insertion paths in the open.

Solid machining of the pocket is carried out starting from the top downwards.

Explanation of the parameters

For the parameters _RTP, _RFP, _SDIS, see CYCLE81.

For the parameter _DP, see LONGHOLE.

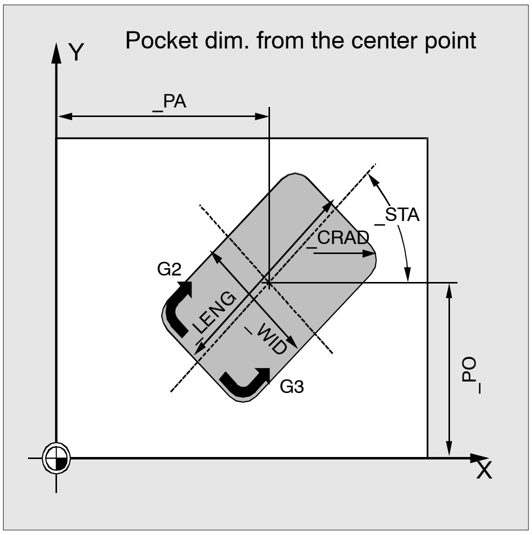


Fig. 9-68

_LENG, _WID and _CRAD (pocket length, pocket width and corner radius)

Use the parameters _LENG, _WID and _CRAD to define the form of a pocket in the plane.

If you cannot traverse the programmed corner radius with the active tool since its radius is larger, then the corner radius of the machine pocket corresponds to the tool radius.

If the milling tool radius is larger than half of the length or width of the pocket, then the cycle will be aborted and alarm 61105 "Cutter radius too large" is output.

_PA, _PO (reference point)

Use the parameters _PA and _PO to define the reference point of the pocket in the axes of the plane.
This is the pocket center point.

_STA (angle)

_STA specifies the angle between the 1st axis of the plane (abscissa) and the longitudinal axis of the pocket.

_MID (infeed depth)

Use this parameter to define the maximum infeed depth in roughing.

The depth infeed is performed by the cycle in equally sized infeed steps.

Using _MID and the entire depth, the cycle will calculate this infeed automatically. The minimum possible number of infeed steps is used as the basis.

_MID=0 means that the cut to pocket depth is made with one feed.

_FAL (finishing allowance on the edge)

The finishing allowance affects the machining of the pocket in the plane only on the edge.

If finishing allowance ≥ tool diameter, complete solid machining of then pocket is not guaranteed. The message "Caution: Finishing allowance ≥ tool diameter" appears; the cycle, however, is continued.

_FALD (finishing allowance at the base)

In roughing, a separate finishing allowance is taken into account at the base.

_FFD and _FFP1 (feedrate for depth and surface)

The feedrate _FFD is effective when inserting into the material.

The feedrate FFP1 is active for all movements in the plane which are traversed at feedrate when roughing.

_CDIR (milling direction)

Use this parameter to specify the machining direction for the pocket.

By using the parameter _CDIR, the milling direction

- can be programmed directly with "2 for G2" and "3 for G3" or,
- alternatively, "Synchronized operation" or "Reverse rotation".

Synchronized operation or reverse rotation are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

Synchronized operation	Reverse rotation
M3 → G3	M3 → G2
M4 → G2	M4 → G3

_VARI (machining type)

Use the parameter _VARI to define the machining type.

Possible values are:

Units digit:

- 1=roughing
- 2=finishing

Tens digit (infeed):

- 0=vertically to pocket center with G0
- 1=vertically to pocket center with G1
- 2=along a helical path
- 3=oscillating to pocket length axis

If a different value is programmed for the parameter _VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

_MIDA (max. infeed width)

Use this parameter to define the maximum infeed width when solid machining in a plane. Analogously to the known calculation method for the infeed depth (equal distribution of the total depth with the maximum possible value), the width is distributed equally, maximally with the value programmed under _MIDA.

If this parameter is not programmed or has value 0, the cycle will internally use 80% of the milling tool diameter as the maximum infeed depth.

Further notes

applies if the calculated width infeed from edge machining is recalculated when reaching the full pocket in the depth; otherwise the width infeed calculated at the beginning is kept for the whole cycle.

_AP1, _AP2, _AD (blank dimension)

Use the parameters _AP1, _AP2 and _AD to define the blank dimension (incremental) of the pocket in the plane and in the depth.

_RAD1 (radius)

Use the parameter _RAD1 to define the radius of the helical path (referred to the tool center point path) or the max. insertion angle for the oscillating motion.

_DP1 (insertion depth)

Use the parameter _DP1 to define the infeed depth when inserting to the helical path.

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 ”No tool compensation active” is output.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point. At the end of the cycle, the original coordinate system is active again.

Programming example: Pocket

Use this program to machine a pocket in the XY plane which is 60 mm in length, 40 mm in width, and which has a corner radius of 8 mm and is 17.5 mm in depth. The pocket has an angle of 0 degrees to the X axis. The finishing allowance for the pocket edges is 0.75 mm, at the base 0.2 mm, and the safety clearance in the Z axis added to the reference plane is 0.5 mm. The pocket center point is at X60 and Y40, the maximum depth infeed is 4 mm.

The machining direction results from the direction of rotation of the spindle in the case of climb milling. A milling cutter with 5 mm radius is used.

Merely a rough machining operation is to be carried out.

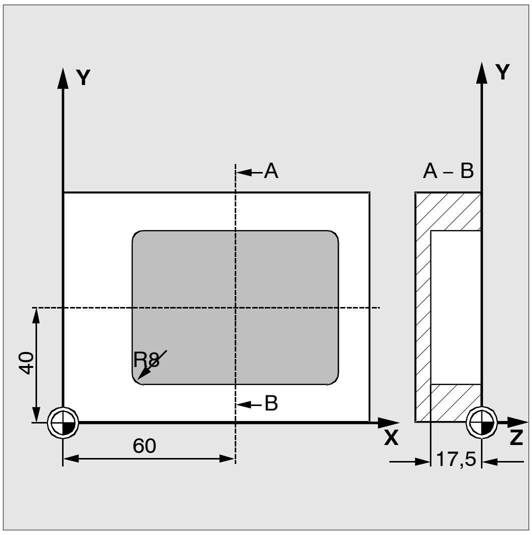


Fig. 9-69

N10 G90 T1 D1 S600 M4	Specification of the technological values
N20 G17 G0 X60 Y40 Z5	Approach starting position
N30 POCKET3(5, 0, 0.5, –17.5, 60, 40, 8, 60, 40, 0, 4, 0.75, 0.2, 1000, 750, 0, 11, 5, , , , ,)	Cycle call
N40 M02	End of program

9.6.10 Milling a circular pocket – POCKET4

Programming

POCKET4(_RTP, _RFP, _SDIS, _DP, _PRAD, _PA, _PO, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _MIDA, _AP1, _AD, _RAD1, _DP1)

Parameters

Table 9-23 Parameters for POCKET4

_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (to be added to the reference plane; enter without sign)
_DP	real	Pocket depth (absolute)
_PRAD	real	Pocket radius
_PA	real	Pocket center point (absolute), 1st axis of the plane
_PO	real	Pocket center point (absolute), 2nd axis of the plane
_MID	real	Maximum infeed depth (enter without sign)
_FAL	real	Finishing allowance at the pocket edge (enter without sign)
_FALD	real	Finishing allowance at the base (enter without sign)
_FFP1	real	Feedrate for surface machining
_FFD	real	Feedrate for depth infeed
_CDIR	integer	Milling direction: (enter without sign) Values: 0 climb milling (as spindle direction) 1 opposed milling 2 with G2 (independent of spindle direction) 3 with G3
_VARI	integer	Machining type UNITS DIGIT Values: 1 roughing 2 finishing TENS DIGIT Values: 0 vertically to pocket center with G0 1 vertically to pocket center with G1 2 along helical path

The other parameters can be selected as options. Specify the plunge–cut strategy and the overlap for solid machining (to be entered without sign):

_MIDA	real	Maximum infeed width as a value in solid machining in the plane
_AP1	real	Pocket radius blank dimension
_AD	real	Blank pocket depth dimension from reference plane

_RAD1	real	Radius of the helical path during insertion (referred to the tool center point path)
_DP1	real	Insertion depth per 360° revolution on insertion along helical path

Function

Use this cycle to machine circular pockets in the machining plane. For finishing, a face cutter is required.

The depth infeed is always started from the pocket center point and performed there vertically; thus predrilling can be suitably performed at this position.

- The milling direction can be determined either using a G command (G2/G3) or from the spindle direction as synchronous or opposed milling.
- For solid machining, the maximum infeed width in the plane can be programmed.
- Finishing allowance also for the pocket base
- Two different insertion strategies:
  – vertically to pocket center
  – along a helical path around the pocket center
- Shorter approach paths in the plane for finishing
- Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).
- _MIDA is recalculated during edge machining.

Sequence

Position reached prior to cycle start:

Starting position is any position from which the pocket center point can be approached at the height of the retraction plane without collision.

Sequence of motions when roughing (VARI=X1):

With G0, the pocket center point is approached at the retraction level, and then, from this position, with G0, too, the reference plane brought forward by the safety clearance is approached. The machining of the pocket is then carried out according to the selected insertion strategy, taking into account the programmed blank dimensions.

Sequence of motions when finishing:

Finishing is performed in the order from the edge until the finishing allowance on the base is reached, and then the base is finished. If one of the finishing allowances is equal to zero, this part of the finishing process is skipped.

- Finishing on the edge

  While finishing on the edge, the tool will traverse around the pocket contour only once.

  For finishing on the edge, the path includes one quadrant reaching the pocket radius. The radius of this path is 2 mm as the maximum or, if ”less space” is provided, equals to the difference between the pocket radius and the mill radius.

  The depth infeed is performed with G0 in the open towards the pocket center, and the starting point of the approach path is also reached with G0.

- Finishing on the base

  During finishing on the base, the machine performs G0 towards the pocket center until reaching a distance equal to pocket depth + finishing allowance + safety clearance. From this point onwards, the tool is always fed in **vertically** at the depth (since a tool with a front cutting edge is used for base finishing).

  The base surface of the pocket is machined once.

Insertion strategies:

see section ”POCKET3”

Taking into account the blank dimensions

During solid machining of the pockets, it is possible to take into account blank dimensions (e.g. when machining precast parts).

With circular pockets, the blank dimension _AP1 is also a circle (with a smaller radius than the pocket radius).

For further explanations, see POCKET3.

Explanation of the parameters

For the parameters _RTP, _RFP, _SDIS, see CYCLE81.

For the parameters _DP, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _MIDA, _AP1, _AD, _RAD1, _DP1, see POCKET3.

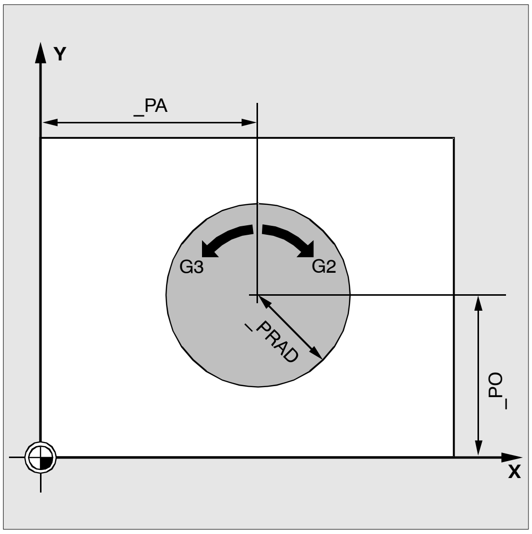


Fig. 9-70

_PRAD (pocket radius)

The form of the circular pocket is determined solely by its radius.

If this is smaller than the tool radius of the active tool, then the cycle is aborted and alarm 61105 ”Cutter radius too large” is output.

_PA, _PO (pocket center point)

Use the parameters _PA and _PO to define the pocket center point. Circular pockets are always dimensioned across the center.

_VARI (machining type)

Use the parameter _VARI to define the machining type.

Possible values are:

Units digit:

- 1=roughing
- 2=finishing

Tens digit (infeed):

- 0=vertically to pocket center with G0
- 1=vertically to pocket center with G1
- 2=along a helical path

If a different value is programmed for the parameter _VARI, the cycle is aborted after output of alarm 61002 ”Machining type defined incorrectly”.

Further notes

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 ”No tool compensation active” is output.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original coordinate system is active again.

Programming example: Circular pocket

With this program, you can machine a circular pocket in the YZ plane. The center point is determined by Y50 Z50. The infeed axis for the depth infeed is the X axis. Neither finishing dimension, nor safety clearance are specified. The pocket is machined with conventional (opposed) milling. Infeed is performed along a helical path.

A milling cutter with 10 mm radius is used.

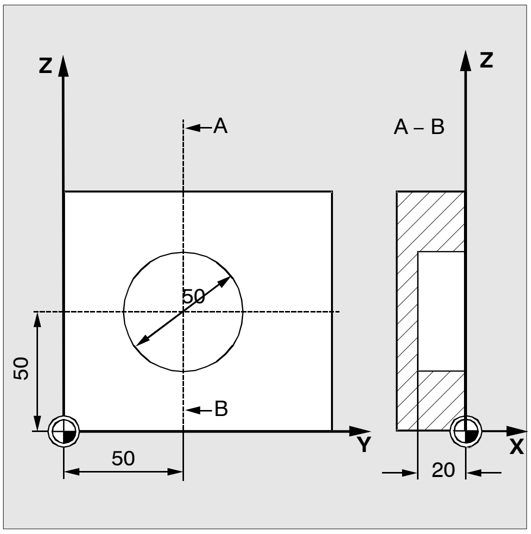


Fig. 9-71

N10 G17 G90 G0 S650 M3 T1 D1	Specification of the technological values
N20 X50 Y50	Approach starting position
N30 POCKET4(3, 0, 0, –20, 25, 50, 60, 6, 0, 0, 200, 100, 1, 21, 0, 0, 0, 2, 3)	Cycle call The parameters _FAL, _FALD are omitted
N40 M02	End of program

9.6.11 Thread milling – CYCLE90

Programming

CYCLE90 (RTP, RFP, SDIS, DP, DPR, DIATH, KDIAM, PIT, FFR, CDIR, TYPTH, CPA, CPO)

Parameters

Table 9-24 Parameters for CYCLE90

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DIATH	real	Nominal diameter, outer diameter of the thread
KDIAM	real	Core diameter, internal diameter of the thread

Table 9-24 Parameters for CYCLE90

PIT	real	Thread lead; Range of values: 0.001 ... 2000.000 mm
FFR	real	Feedrate for thread milling (enter without sign)
CDIR	int	Direction of rotation for thread milling Values: 2 (for thread milling using G2) 3 (for thread milling using G3)
TYPTH	int	Thread type Values: 0=internal thread 1=external thread
CPA	real	Center point of circle, abscissa (absolute)
CPO	real	Center point of circle, ordinate (absolute)

Function

Using the cycle CYCLE90, you can produce internal or external threads. The path when milling threads is based on a helix interpolation. All three geometry axes of the current plane, which you will define before calling the cycle, are involved in this motion.

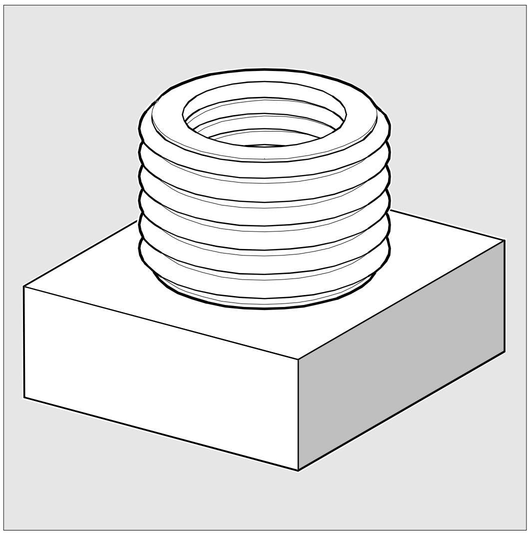


Fig. 9-72

Sequence when producing an external thread

Position reached prior to cycle start:

The starting position is any position from which the starting position at the outside diameter of the thread at the height of the retraction plane can be reached without collision.

When milling threads using G2, this starting position is between the positive abscissa and the positive ordinate in the current plane (i.e. not in the 1st quadrant of the coordinate system). When milling threads using G3, the starting position is between the positive abscissa and the negative ordinate (i.e. in the 4th quadrant of the coordinate system).

The distance from the thread diameter depends on the size of the thread and the tool radius used.

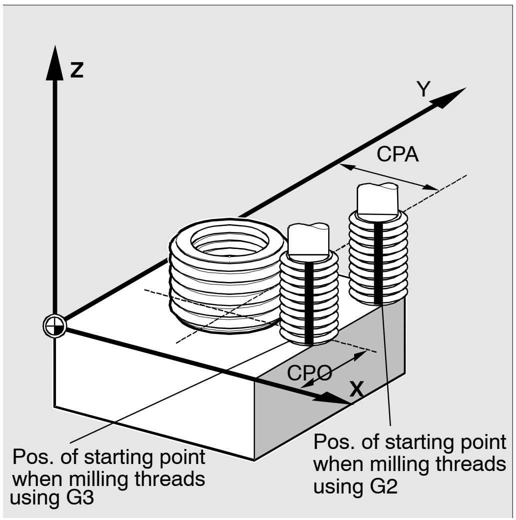


Fig. 9-73

The cycle creates the following sequence of motions:

- Positioning on the starting point using G0 at the height of the retraction plane in the applicate of the current plane
- Infeed to the reference plane brought forward by the safety clearance for swarf removal, using G0
- Approach motion to the thread diameter along a circle path opposite to the direction G2/G3programmed under CDIR
- Thread milling along a helix path using G2/G3 and the feedrate value FFR
- Retraction motion along a circle path in the opposite direction of rotation G2/G3 at the reduced feedrate FFR
- Retraction to the retraction plane along the applicate using G0

Sequence when producing an internal thread

Position reached prior to cycle start:

The starting position is any position from which the center point of the thread at the height of the retraction plane can be reached without collision.

The cycle creates the following sequence of motions:

- Positioning on the center point using G0 at the height of the retraction plane in the applicate of the current plane
- Infeed to the reference plane brought forward by the safety clearance for swarf removal, using G0
- Approach to an approach circle calculated internally in the cycle using G1 and the reduced feedrate FFR
- Approach motion to the thread diameter along a circle path according to the direction G2/G3programmed under CDIR
- Thread milling along a helix path using G2/G3 and the feedrate value FFR
- Retraction motion along a circle path in the same direction of rotation at the reduced feedrate FFR
- Retraction to the center point of the thread using G0
- Retraction to the retraction plane along the applicate using G0

Thread from the bottom to the top

For technological reasons, it can be reasonable to machine a thread also from the bottom to the top. In this case, the retraction plane RTP will be behind the thread depth DP.

This machining is possible, but the depth specifications must be programmed as absolute values and the retraction plane must be approached before calling the cycle or a position after the retraction plane must be approached.

Programming example (thread from the bottom to the top)

A thread with a lead of 3 mm is to start from −20 and to be milled to 0. The retraction plane is at 8.

N10 G17 X100 Y100 S300 M3 T1 D1 F1000
N20 Z8
N30 CYCLE90(8, −20, 0, −60, 0, 46, 40, 3, 800, 3, 0, 50, 50)
N40 M2

The drill hole must have a depth of at least −21.5 (a half lead more).

Overrun travels towards the thread length

The travel−in / travel−out movements when milling threads is executed in all three axes involved. This will result in an additional travel along the vertical axis at the thread run−out, which goes beyond the programmed thread depth.

The overrun travel is calculated as follows:

$$\Delta z = \frac{P}{4} * \frac{2*WR+RDIFF}{DIATH}$$

Δz	Overrun travel, internally
p	Lead/pitch of the thread
WR	Tool radius
DIATH	Outer diameter of the thread
RDIFF	Radius difference for the retraction circle

With internal threads, RDIFF = DIATH/2 – WR,
with external threads, RDIFF = DIATH/2 + WR.

Explanation of the parameters

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

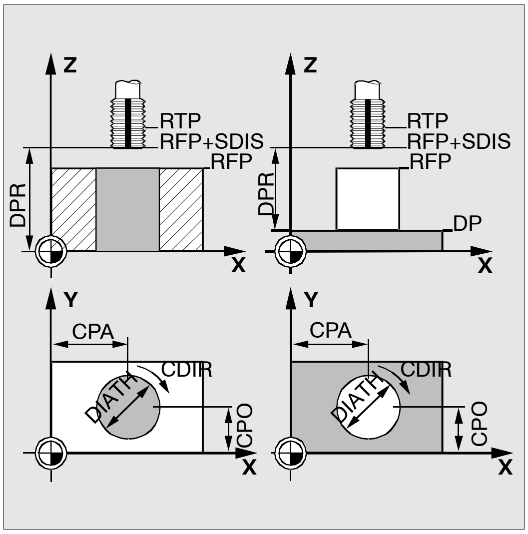


Fig. 9-74

DIATH, KDIAM and PIT (nominal/core diameters and thread lead/pitch)

These parameters are used to determine the thread data nominal diameter, core diameter and lead/pitch. The parameter DIATH is the external, and KDIAM is the internal diameter of the thread. The travel–in / travel–out movements are created internally in the cycle, based on these parameters.

FFR (feedrate)

The value of the parameter FFR is specified as the current feedrate value for thread milling. It is effective when thread milling on a helix path.

This value will be reduced in the cycle for the travel–in / travel–out movements. The retraction is performed outside the helix path using G0.

CDIR (direction of rotation)

This parameter is used to specify the value for the machining direction of the thread.

If the parameter has an illegal value, the following message will appear:

”Wrong milling direction; G3 is generated”.

In this case, the cycle is continued and G3 is automatically generated.

TYPTH (thread type)

The parameter TYPTH is used to define whether you want to machine an external or an internal thread.

CPA and CPO (center point)

These parameters are used to define the center point of the drill hole or of the spigot on which the thread will be produced.

Further notes

The cutter radius is calculated internally in the cycle. Therefore, a tool compensation must be programmed before calling the cycle. Otherwise, the alarm 61000 ”No tool compensation active” appears and the cycle is aborted.

If the tool radius=0 or negative, the cycle is also aborted and this alarm is issued.

With internal threads, the tool radius is monitored and alarm 61105 ”Cutter radius too large” is output, and the cycle is aborted.

Programming example: Internal thread

Using this program, you can mill an internal thread at point X60 Y50 of the G17 plane.

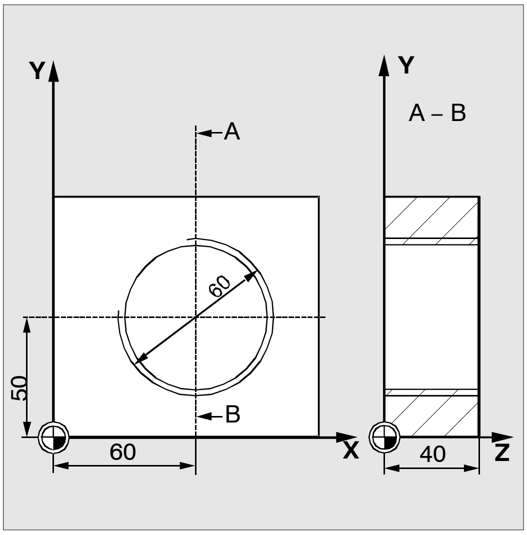


Fig. 9-75

DEF REAL RTP=48, RFP=40, SDIS=5, DPR=40, DIATH=60, KDIAM=50 **DEF REAL PIT=2, FFR=500, CPA=60,CPO=50** **DEF INT CDIR=2, TYPTH=0**	Definition of the variable with value assignments
N10 G90 G0 G17 X0 Y0 Z80 S200 M3	Approach starting position
N20 T5 D1	Specification of the technological values
N30 CYCLE90 (RTP, RFP, SDIS, DP, DPR, DIATH, KDIAM, PIT, FFR, CDIR, TYPTH, CPA CPO)	Cycle call
N40 G0 G90 Z100	Approach position after cycle
N50 M02	End of program

9.7 Error messages and error handling

9.7.1 General notes

If error conditions are detected in the cycles, an alarm is generated and the execution of the cycle is aborted.

Furthermore, the cycles display their messages in the message line of the control system. These message will not interrupt the program execution.

The errors with their reactions and the messages in the message line of the control system are described in conjunction with the individual cycles.

9.7.2 Error handling in the cycles

If error conditions are detected in the cycles, an alarm is generated and the machining is aborted.

Alarms with numbers between 61000 and 62999 generated in the cycles. This range of numbers, in turn, is divided again with regard to alarm responses and cancel criteria.

The error text that is displayed together with the alarm number gives you more detailed information on the error cause.

Table 9-25

Alarm Number	Clear Criterion	Alarm Response
61000 ... 61999	NC_RESET	Block preparation in the NC is aborted
62000 ... 62999	Clear key	The block preparation is interrupted; the cycle can be continued with NC START after the alarm has been cleared.

9.7.3 Overview of cycle alarms

The error numbers are classified as follows:

6	_	X	_	_

- X=0 General cycle alarms
- X=1 Alarms generated by the drilling, drilling pattern and milling cycles

The Table below includes a list of all errors occurring in the cycles with their location of occurrence and appropriate instructions for fault correction.

Table 9-26

Alarm Number	Alarm Text	Source	Explanation, Remedy
61000	"No tool compensation active"	SLOT1 SLOT2 POCKET3 POCKET4 CYCLE71 CYCLE72	D offset must be programmed prior to cycle call
61001	"Illegal thread pitch"	CYCLE84 CYCLE840	Check the parameters for the thread size or the specifications for the thread (are contradicting)
61002	"Machining type defined incorrectly"	SLOT1 SLOT2 POCKET3 POCKET4 CYCLE71 CYCLE72	The value of parameters VARI for the machining type is specified incorrectly and must be changed
61003	"No feedrate programmed in the cycle"	CYCLE71 CYCLE72	The parameter for the feed is incorrectly specified and must be changed.
61009	"Active tool number = 0"	CYCLE71 CYCLE72	No tool (T) is programmed prior to the cycle call.
61010	"Finishing allowance too large"	CYCLE72	The finishing allowance at the root is larger than the overall depth; it must be reduced.
61011	"Scaling not permitted"	CYCLE71 CYCLE72	A scaling factor is active what is not permissible for this cycle.
61101	"Reference plane defined incorrectly"	CYCLE71 CYCLE72 CYCLE81 to CYCLE89 CYCLE840 SLOT1 SLOT2 POCKET3 POCKET4	Either different values for reference and retraction plane must be selected in the case of relative specification of the depth or an absolute value must be specified for the depth.
61102	"No spindle direction programmed"	CYCLE86 CYCLE88 CYCLE840 POCKET3 POCKET4	The parameter SDIR (or SDR in CYCLE840) must be programmed
61103	"Number of holes is zero"	HOLES1 HOLES2	No value for the number of holes programmed
61104	"Contour violation of the slots / elongated holes"	SLOT1 SLOT2	Faulty parameterization of the milling pattern in the parameters defining the position of the slots/long holes on the circle and their form
61105	"Cutter radius too large"	SLOT1 SLOT2 POCKET3 POCKET4	The diameter of the cutter used is too large for the figure to be manufactured; either use a tool with smaller radius or change the contour
61106	"Number or distance of circle elements"	HOLES2 SLOT1 SLOT2	Parameters NUM or INDA parameterized incorrectly; the arrangement of the circle elements within a full circle is not possible

Table 9-26 , Fortsetzung

Alarm Number	Alarm Text	Source	Explanation, Remedy
61107	”First drilling depth defined incorrectly”	CYCLE83	First drilling depth is opposite to total drilling depth
61108	”No permissible values for the parameters _RAD1 and _DP1”	POCKET3 POCKET4	The parameters _RAD1 and _DP for defining the path for depth infeed were incorrectly specified.
61109	”Parameter _CDIR defined incorrectly”	POCKET3 POCKET4	The value of the _CDIR parameter for the milling direction was specified incorrectly and must be changed.
61110	”Finishing allowance at the base > depth infeed”	POCKET3 POCKET4	The finishing allowance on the root has been specified greater than the maximum depth infeed; either reduce the finishing allowance or increase the depth infeed.
61111	”Infeed width > tool diameter”	CYCLE71 POCKET3 POCKET4	The programmed infeed width is larger than the diameter of the active tool; it must be reduced.
61112	Tool radius negative”	CYCLE72	The radius of the active tool is negative; this is not permissible.
61113	”Parameter _CRAD for corner radius too large”	POCKET3	The parameter for the corner radius _CRAD was specified too large; it must be reduced.
61114	”Machining direction G41/G42 defined incorrectly”	CYCLE72	The machining direction of the cutter radius path compensation G41/G42 was selected incorrectly.
61115	”Approach or retraction mode (straight line/circle/plane/space) defined incorrectly”	CYCLE72	The approach / retraction mode for the contour was incorrectly defined; check parameter _AS1 or _AS2.
61116	”Approach or retraction path=0”	CYCLE72	The approach orretraction travel is specified with zero; it must be increased; check parameter _LP1 or _LP2.
61117	”Active tool radius <= 0”	CYCLE71 POCKET3 POCKET4	The radius of the active tool is negative or zero; this is not permissible!
61118	”Length or width = 0”	CYCLE71	The length or width of the milling face is not permissible; check the parameters _LENG and _WID.
61124	”No infeed width programmed”	CYCLE71	With the simulation active without tool, a value for the infeed width _MIDA must always be programmed.
62100	”No drilling cycle active”	HOLES1 HOLES2	No drilling cycle has been called modally before the drilling cycle has been called.

9.7.4 Messages in the cycles

The cycles display their messages in the message line of the control system. These message will not interrupt the program execution.

Messages provide information with regard to a certain behavior of the cycles and with regard to the progress of machining and are usually kept beyond a machining step or until the end of the cycle. The following messages are possible:

Table 9-27

Message Text	Source
"Depth: according to the value for the relative depth"	CYCLE81...CYCLE89, CYCLE840
"Slot is being processed"	SLOT1
"Circumferential slot is being processed"	SLOT2
"Wrong milling direction; G3 is generated".	SLOT1, SLOT2
"1st drilling depth: according to the value for the relative depth"	CYCLE83

no.> in the message text always stands for the number of the figure currently machined.

This sheet has been left empty for your notes

Index

M

N

O

P

R

S

T

V

W

To:
SIEMENS AG
A&D MC BMS
Postfach 3180

D–91050 Erlangen

(Tel. +49 (0) 180 5050 – 222 [hotline]
Fax +49 (0) 9131 98 – 63315 [documentation]
E–mail: motioncontrol.docu@siemens.com)

Suggestions

Corrections

For publication/manual:

SINUMERIK 802D

User Documentation

From:

Name

Company/Dept.

Address

Zip code: City:

Telephone: /

Telefax: /

Operation and Programming
Milling
Order No.: 6FC5698–2AA10–1BP5
08.05Edition

Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.

Suggestions and/or corrections

SINUMERIK 802D Document Structure