Operation and Programming 08/2003 Edition

sinumerik

SINUMERIK 802S base line
SINUMERIK 802C base line
Turning

SIEMENS

SIEMENS

SINUMERIK 802S base line
SINUMERIK 802C base line

Operation and Programming
Turning

Valid for

Control system	*Software version*
SINUMERIK 802S base line	4
SINUMERIK 802C base line	4

2003.08 Edition

SINUMERIK® Documentation

Key to editions

The editions listed below have been published prior to the current edition.

The column headed “Note” lists the amended sections, with reference to the previous edition.

Marking of edition in the “Note” column:

A ... ... New documentation.
B ... ... Unchanged reprint with new order number.
C ... ... Revised edition of new issue.

Edition	Order No.	Note
1999.02	6FC5598-2AA00-0BP1	**A**
2000.04	6FC5598-3AA00-0BP1	**A**
2002.01	6FC5598-3AA00-0BP2	**C**
2003.08	6FC5598-4AA01-0BP0	**A**

Trademarks

SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIMODRIVE®, SINUMERIK®, and SIMOTION® are registered trademarks of SIEMENS AG.

Other names in this publication might be trademarks whose use by a third party for his own purposes may violate the registered holder.

Exclusion of liability

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

Safety Guidelines

This Manual contains notices intended to ensure your personal safety , as well as to protect products and connected equipment against damage. Safety notices are highlighted by a warning triangle and presented in the following categories depending on the degree of risk involved:

Danger

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

Warning

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

Caution

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

Caution

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

Notice

Indicates important information relating to the product or highlights part of the documentation for special attention.

Qualified person

The unit may only be started up and operated by qualified person or persons. Qualified personnel as referred to in the safety notices provided in this document are those who are authorized to start up, earth and label units, systems and circuits in accordance with relevant safety standards.

Proper use

Please observe the following:

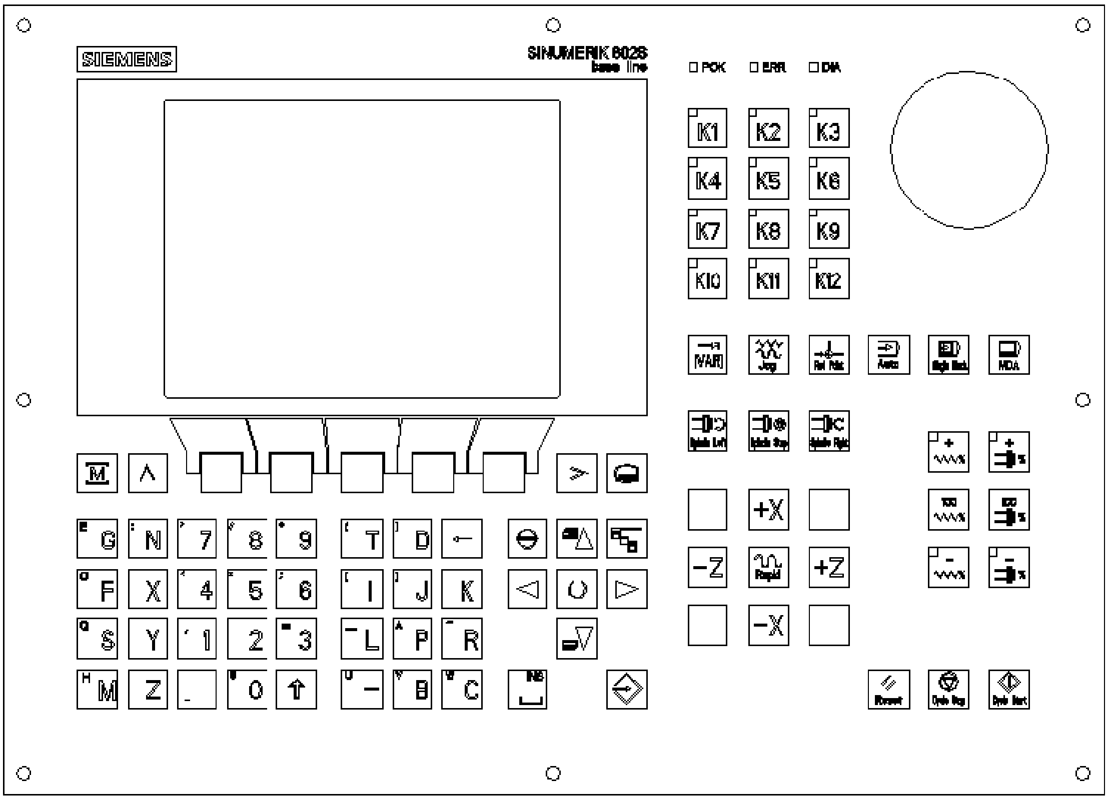
Warning

The unit may be used only for the applications described in the catalog or the technical description, and only in combination with the equipment, components and devices of other manufacturers as far as this is recommended or permitted by Siemens.

This product must be transported, stored and installed as intended, and maintained and operated with care to ensure that it functions correctly and safely.

Contents

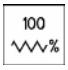
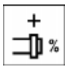
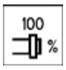
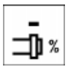
SINUMERIK 802S/C base line Operator Panel OP



NC keyboard area (left side):

	Softkey		Vertical menu
M	Machine area key		Acknowledge alarm
Λ	Recall key		Selection key/toggle key
>	ETC key		ENTER / input key
	Area switchover key	⇧	Shift key
	Cursor UP with shift: page up		Cursor DOWN with shift: page down
	Cursor LEFT		Cursor RIGHT
←	Delete key (backspace)	INS	SPACE (INSERT)
$ 0 + 9	Numerical keys shift for alternative assignment	U – Z	Alphanumeric keys shift for alternative assignment

<u>M</u>achine <u>C</u>ontrol <u>P</u>anel area (right side):

	RESET
	NC STOP
	NC START
K1 ... K12	User-defined key with LED
	User-defined key without LED
	INCREMENT
	JOG
	REFERENCE POINT
	AUTOMATIC
	SINGLE BLOCK
	MANUAL DATA
	SPINDLE START LEFT Counterclockwise direction
	SPINDLE START RIGHT Clockwise direction
	SPINDLE STOP
	RAPID TRAVERSE OVERLAY
+X −X	X axis
+Z −Z	Z axis
	Feedrate override plus with LED
	Feedrate override 100% without LED
	Feedrate override minus with LED
	Spindle speed override plus with LED
	Spindle speed override 100% without LED
	Spindle speed override minus with LED

Introduction 1

1.1 Screen layout

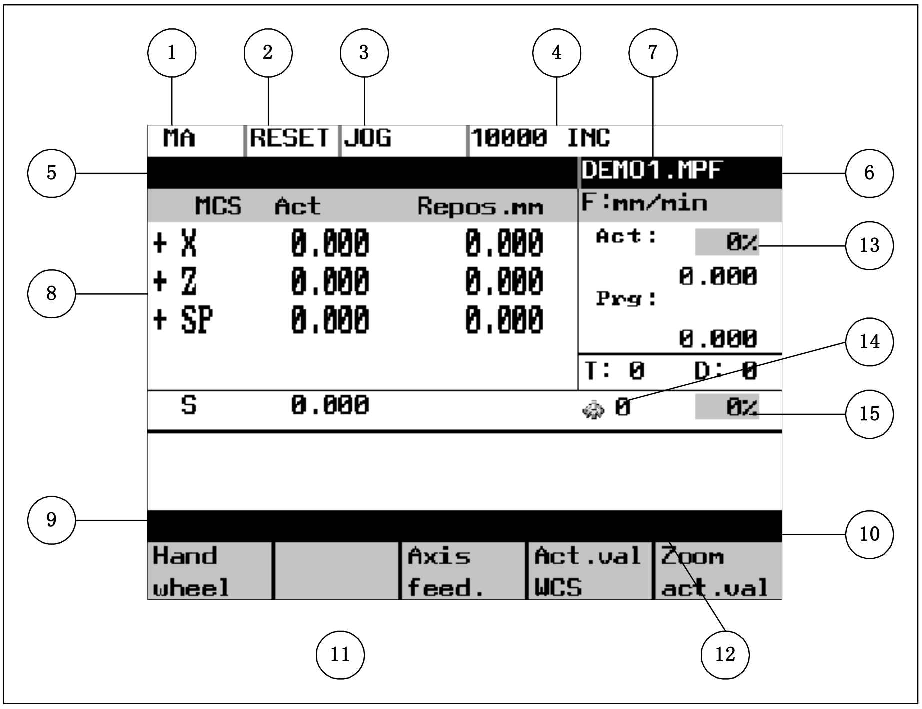


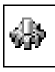
Fig.1-1 Screen layout

The abbreviations on the screen stand for the following:

Table 1–1 Explanation of display elements

Display Element	Abbreviation	Meaning
1 Active operating area	MA	Machine
	PA	Parameter
	PR	Programming
	DI	Services
	DG	Diagnosis
2 Program status	STOP	Programm stopped
	RUN	Program running
	RESET	Program aborted
3 Operating mode	Jog	Manual traverse
	MDA	Manual input with automatic function
	Auto	Automatic

Display Element	Abbreviation	Meaning
(4) Status display	SKP	Skip block Program blocks marked by a slash in front of the block number are ignored during program execution.
	DRY	Dry run feed Traversing movements are executed at the feed specified in the Dry Run Feed setting data.
	ROV	Rapid traverse override The feed override also applies to rapid feed mode.
	SBL	Single block with stop after each block When this function is active, the part program blocks are processed separately in the following manner: Each block is decoded separately, the program is stopped at the end of each block. The only exception are thread blocks without dry run feed. In this case, the program is stopped only when the end of the current thread block is reached. SBL can only be selected in the RESET state.
	M1	Programmed stop When this function is active, the program is stopped at each block in which the miscellaneous function M01 is programmed. In this case, the message “5 stop M00/M01 active“ appears on the screen.
	PRT	Program test
	1…1000 INC	Incremental mode If the control is in the Jog mode, incremental dimension is displayed instead of the active program control function.
(5) Operational message	1	Stop: No NC Ready
	2	
	3	Stop: EMERGENCY STOP active
	4	Stop: Alarm active with stop
	5	Stop: M0/M01 sctive
	6	Stop: Block ended in SBL mode
	7	Stop: NC STOP active
	8	Wait: Read-in enable missing
	9	Wait: Feed enable missing
	10	Wait: Dwell time active
	11	Wait: Auxiliary function acknowl. missing
	12	Wait: Axis enable missing
	13	Wait: Exact stop not reached
	14	
	15	Wait: For spindle
	16	
	17	Wait: Feed override to 0%
	18	Stop: NC block incorrect
	19	
	20	
	21	Wait: Block search active
	22	Wait: No spindle enable
	23	Wait: Axis feed value 0
(6) Program name		

Display Element	Abbreviation	Meaning
7 Alarm line		The alarm line is only displayed if an NC or PLC alarm is active. The alarm line contains the alarm number and reset criterion of the most recent alarm.
8 Working window		Working window and NC display
9 Recall symbol	∧	This symbol is displayed above the softkey bar when the operator is in a lower-level menu. When the Recall key is pressed, you can return to the next-higher menu without saving data.
10 Menu extension	>	ETC is possible If this symbol appears above the softkey bar, further menu functions are provided. These functions can be activated by the ETC key.
11 Softkey bar		
12 Vertical menu		If this symbol is displayed above the softkey bar, further menu functions are provided. When the VM key is pressed, these functions appear on the screen and can be selected by Cursor UP and Cursor DOWN.
13 Feedrate override	0%	Here the current actual feedrate override is shown.
14 Gear box	 0	Here the current spindle gear stage 1…5 is shown.
15 Spindel speed override	0%	Here the current spindel speed override is shown.

1.2 Operating areas

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

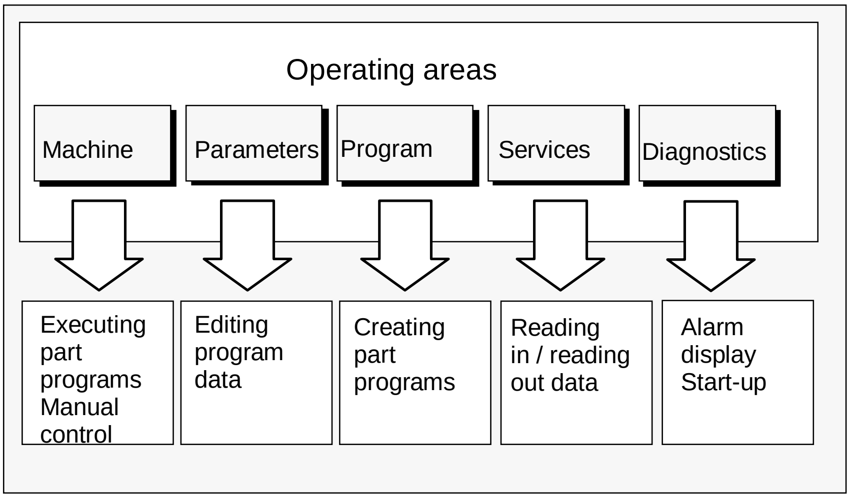


Fig.1-2 SINUMERIK 802S/C base line operating areas

Switching between the operating

Press the “Machine” area key for direct access to the “Machine” operating area.

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

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

After turning on the control system, the Machine operating area will appear by default.

Protection levels

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

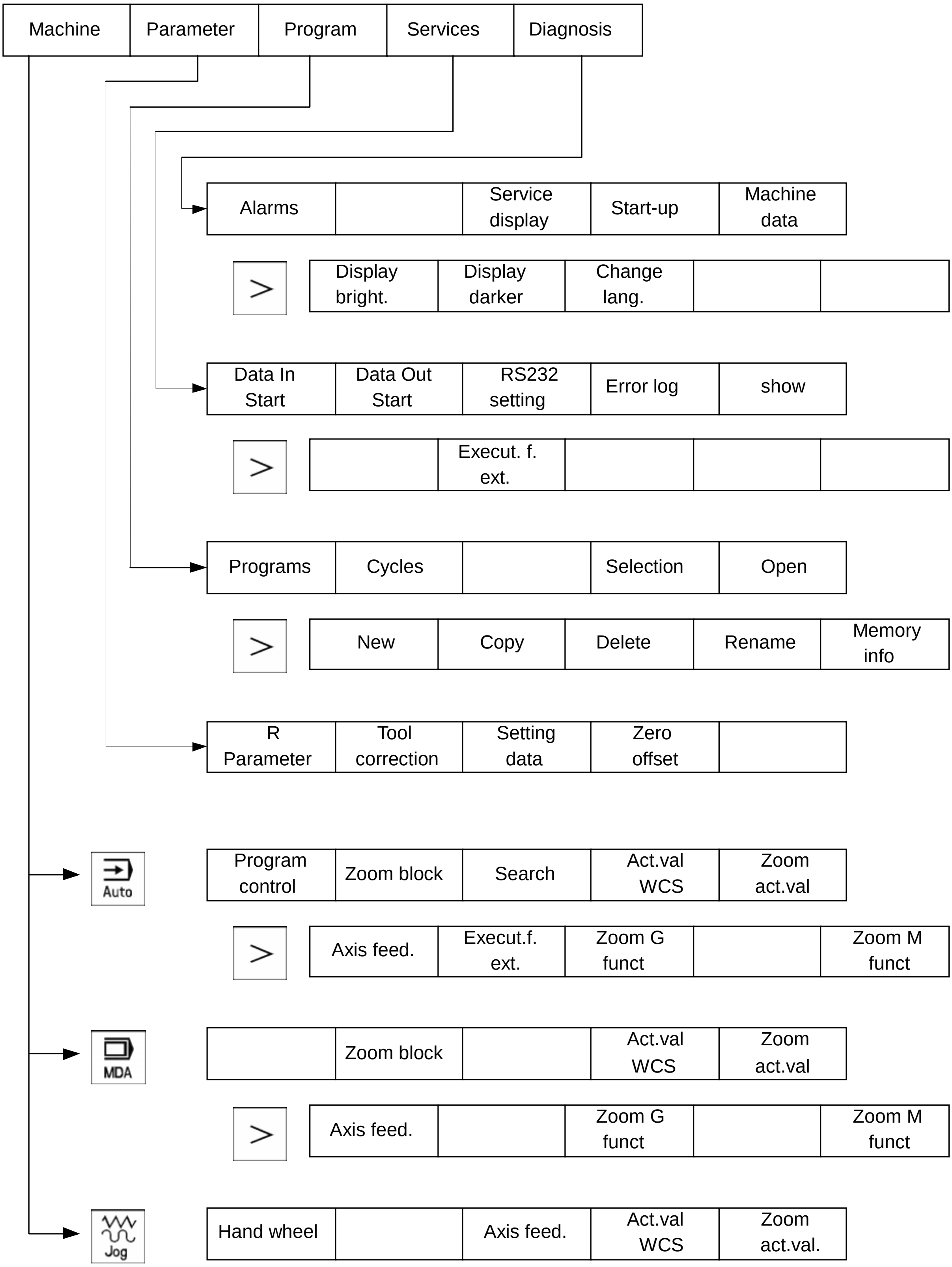
However, the operator can alter the protection levels in the “Machine Data” display menu in the “Diagnostics” operating area.

Default: Protection level 3.

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

- tool offsets
- zero offsets
- setting data
- RS232 settings

1.3 Overview of the most important softkey functions



1.4 Pocket calculator

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

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

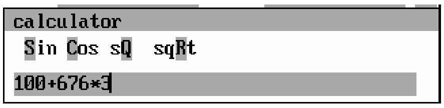


Fig. 1-3 Pocket calculator

Permissible characters

The following characters are permitted for input:

+ Value X plus value Y

- Value X minus value Y

* Value X multiplied with value Y

/ Value X divided by value Y

S Sine function
The value X in front of the input cursor is replaced by the value sin(X).

C Cosine function
The value X in front of the input cursor is replaced by the value cos(X).

Q Square function
The value X in front of the input cursor is replaced by the value X^{2}.

R Square root function
The value X in front of the input cursor is replaced by the value $\sqrt{X}$.

Calculation examples

Task	Input
100 + (67*3)	100+67*3
sin(45°)	45 S -> 0.707107
cos(45°)	45 C -> 0.707107
4^2	4 Q -> 16
$\sqrt{4}$	4 R -> 2

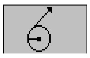
The calculation is carried out by pressing the Input key. The softkey function OK will accept the result into the input field, quitting the calculator automatically.

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

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

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

Softkeys



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

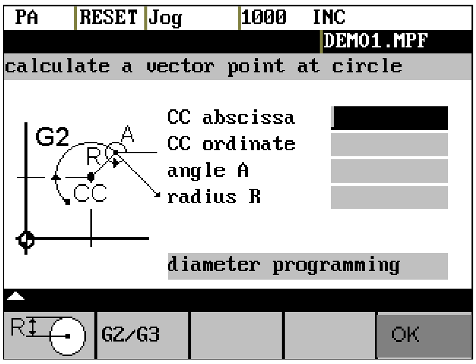
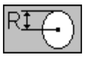


Fig.1-4 Calculation of a point on a circle



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

The function switches the screen form from diameter programming to radius programming.

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

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

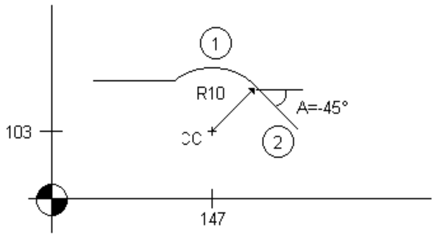
If plane G18 is active, the abscissa is the Z axis, and the ordinate is the X axis.

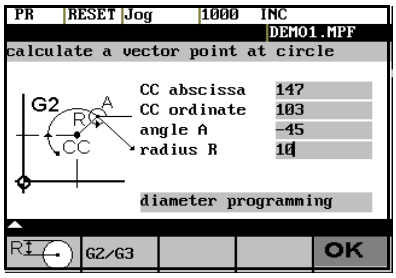
The value of the abscissa is copied into that input field from which the pocket calculator function has been called, and the ordinate value into the next following input field.

Example

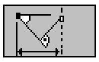
Calculating the intersection point between the circle sector 1 and the straight line 2.

Given:	Radius: 10
Circle center point:	Z 147 X103
Ongoing angle of the straight line:	-45°





Result: Z = 154.071
X = 117.142



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

The following values of the straight line are known:

Straight line 1: Start point and rise angle

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

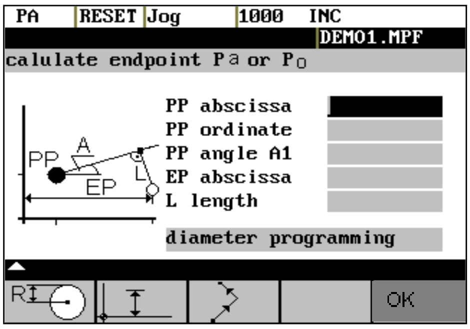
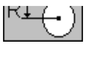
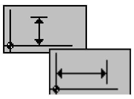


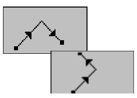
Fig.1-5



The function switches the screenform from diameter programming to radius programming.



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



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

The function chooses the appropriate setting.

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

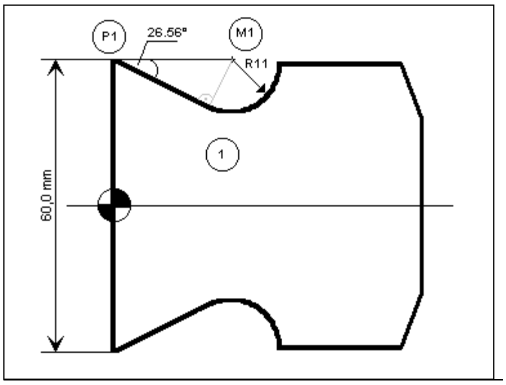
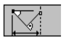
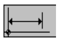


Fig.1-6

The drawing above must be added by the value of the circle center point to be able to calculate the intersection point between the circle sector of the straight line. The missing coordinate of the center point is calculated by means of the pocket calculator function  , since the radius in the tangential transition stands vertical on the straight line.

Calculating M1 in section 1:

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

Use the softkeys  and  to select the given constellation .

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

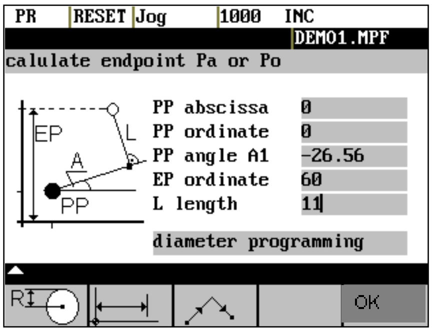


Fig.1-7

Result: Z = 24.601
X = 60

1.5 Coordinate systems

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

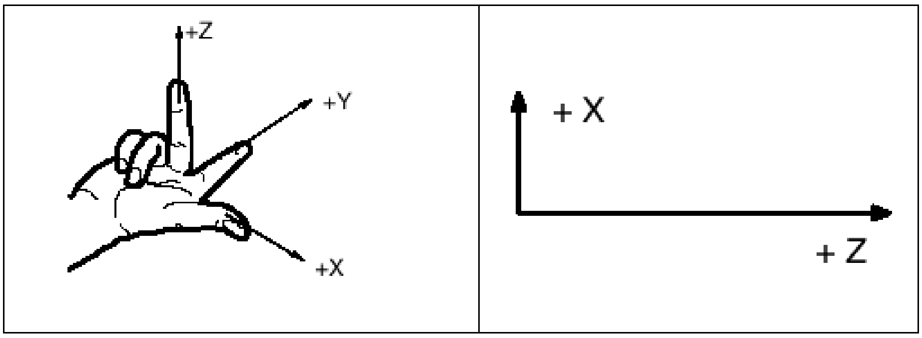


Fig.1-8 Specification of the axis directions to one another; coordinate system when programming for turning

Machine coordinate system (MCS)

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

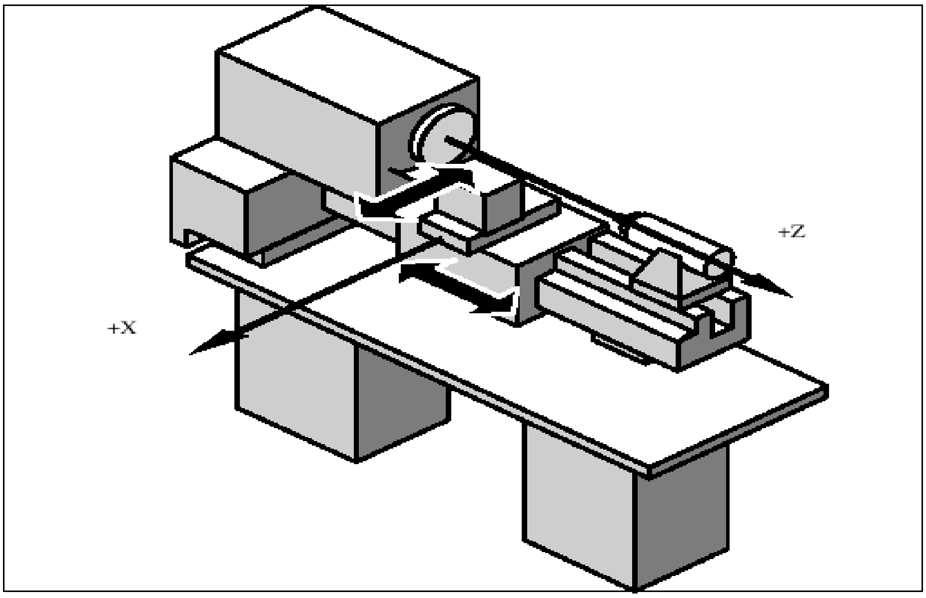


Fig. 1-9 Machine coordinates/axes on a turning machine

The origin of this coordinate system is the machine zero.

All axes are in the zero position at this point. This point is merely a reference point determined by the machine manufacturer. It does not need to be approachable.

The traversing range of the machine axes can be negative.

Workpiece coordinate system (WCS)

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

The workpiece zero can be freely selected in the Z axis by the programmer. In the Z axis, the zero point corresponds to the turning center.

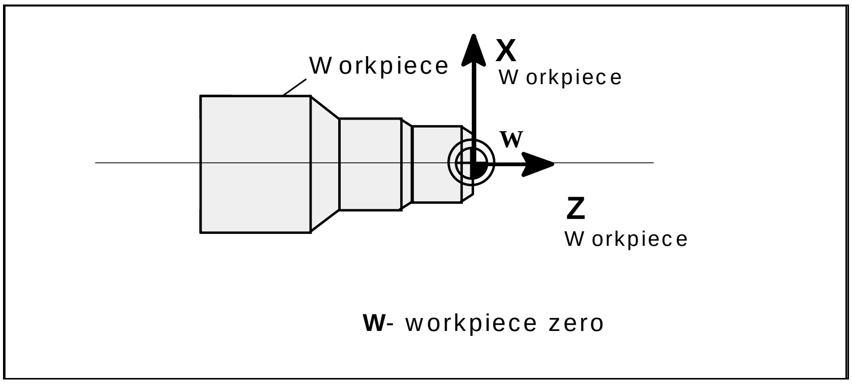


Fig.1-10 Workpiece coordinate system

Workpiece clamping

To machine the workpiece, it is clamped in the machine. The workpiece must be aligned such that the axes of the workpiece coordinate system are in parallel with the machine axes. Any resultant offset of the machine zero to the workpiece zero is determined in the Z axis and entered in a specially provided data area for the settable zero offset. This offset is activated during the NC program execution by means, for example, of a programmable G54 (see Section “Workpiece Clamping - Settable Zero Offset ...”).

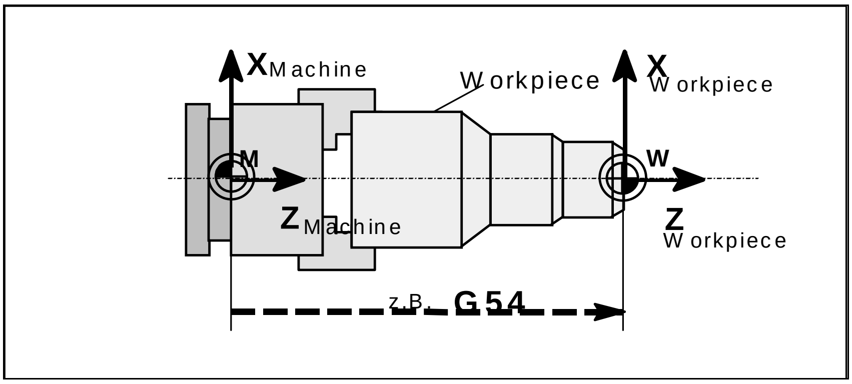


Fig.1-11 Workpiece on the machine

Current workpiece coordinate system

An offset in relation to the workpiece coordinate system can be generated by means coordinate system of the programmable zero offset G158. The result is the current workpiece (see Section “Programmable Zero Offset: G158”).

Turning On and Reference Point Approach 2

Notice

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

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

The Reference point approach window is active.

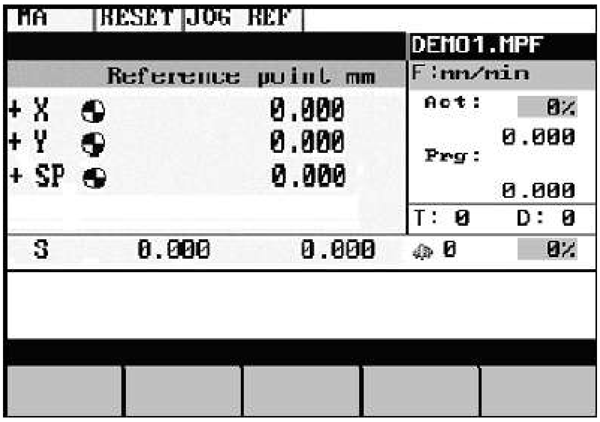
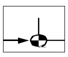


Fig.2-1 Jog Ref basic screen

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



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

In the “Reference point approach” window (Fig. NO TAG), it is displayed whether or not the axes have to be referenced.

○ Axis has to be referenced

◕ Axis has reached the reference point

+X … −Z

Press the direction keys.

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

Approach the reference point in each axis successively.

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

Set Up

3

Preliminary remarks Before you can use the CNC, set up the machine, tools, etc. on the CNC by:

- entering the tools and tool offsets
- entering/modifying the zero offset
- entering the setting data

3.1 Entering tools and tool offsets

Functionality

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

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

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

See also Section 8.6 “Tool and Tool Offset“.

Operating sequences

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

Parameter

	Tool Corr.

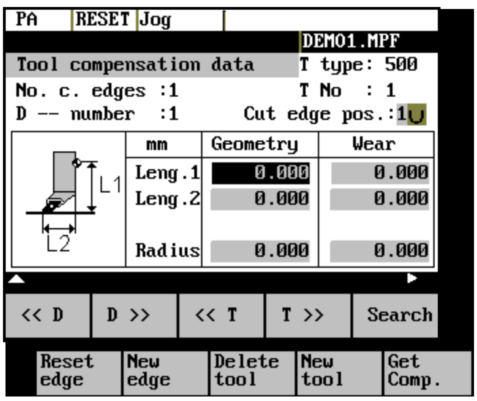


Fig.3-1 Tool compensation data window

Softkeys

<< D / D >>	Select next lower or next higher edge number.
<< T / T >>	Select next lower or next higher tool.
Get Comp.	Determine length compensation values.
>	Use the ETC key to extend the softkey functions.
Reset edge	All edge compensation values are reset to zero.
New edge	Creates a new edge and loads it with the appropriate parameters. The new edge is created for the currently displayed tool; it is automatically assigned the next higher edge number (D1 – D9). Max. 30 edges (in total) can be stored in the memory.
Delete tool	Deletes the tool compensation data of all edges of the selected tool.
New tool	Creates new tool compensation data for a new tool. Note: Max. 15 tools can be created.
Search	Pressing this softkey opens the dialog box and the overview of the tool numbers assigned. Enter the tool number you search for in the input window and start search with OK. If the searched tool exists, the search function opens the tool offset data box.

3.1.1 Creating a new tool

Operating sequence

New tool

Press this softkey to create a new tool.

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

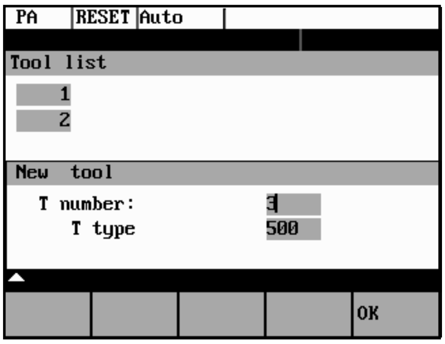


Fig 3-2 New Tool window

0 … 9

Enter the new T number (maximal only three digits) and specify the tool type.

OK

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

3.1.2 Tool compensation data

The tool compensation data are divided into length and radius compensation data.

The list is structured according to the tool type.

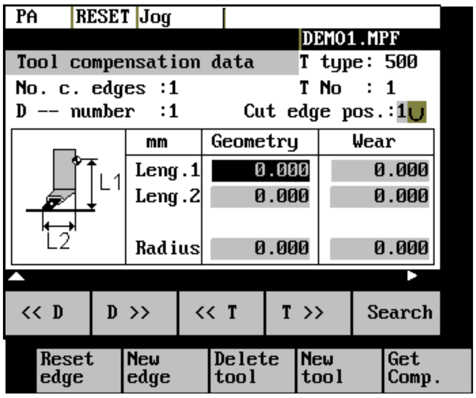
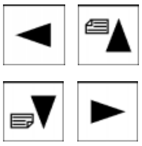
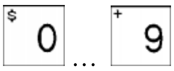


Fig.3-3 Tool compensation data window

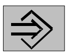
Operating sequence Enter the offsets by



positioning the cursor on the input field to be modified,



entering value(s)



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

3.1.3 Determining the tool offsets

Functionality

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

Prerequisite

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

Procedure

Enter the offset value into the intended Offset field. Then select the required zero offset (e.g. G54) or G500 if no zero offset is to be calculated. These entries must be made for each selected axis (see Fig. 3-6).

Please note the following: The assignment of length 1 or 2 to the axis depends on the type of tool (turning tool, drill)..
For the turning tool, the offset value for the X axis is a diameter dimension.

Using the actual position of point F (machine coordinate), the offset entry and the selected zero offset Gxx (position of the edge), the control system can calculate the assigned compensation value of length 1 or length 2 for the preselected axis.

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

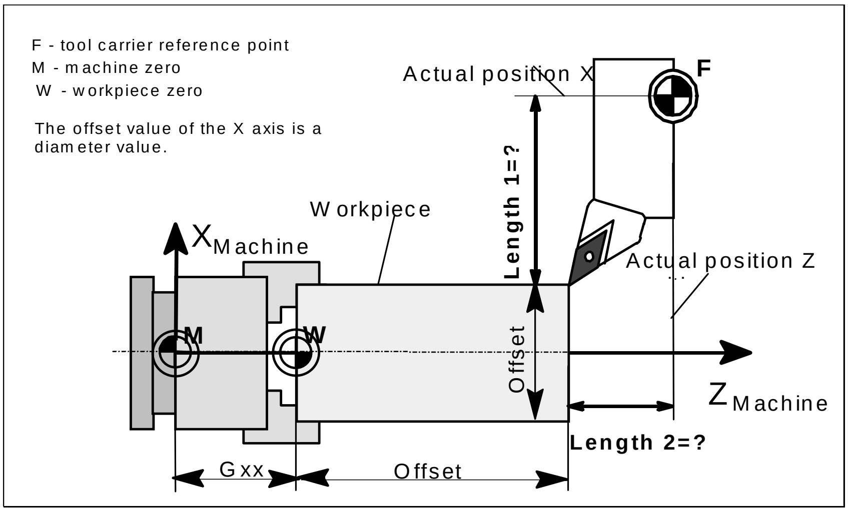


Fig.3-4 Determination of the length compensation values using the example of a cutting tool

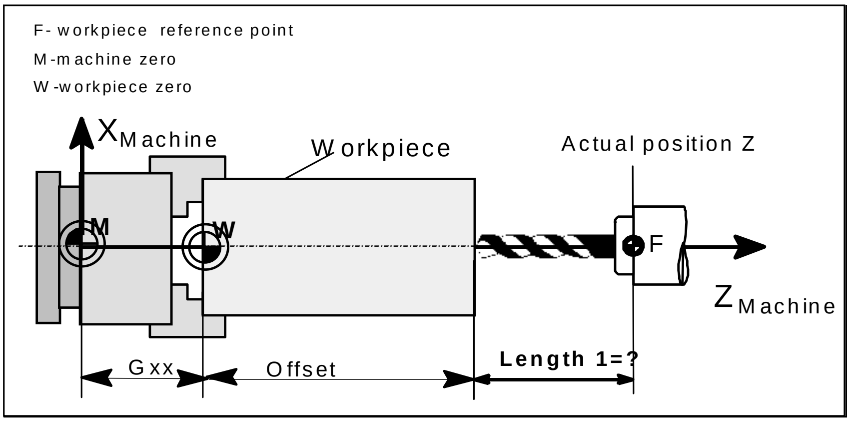


Fig.3-5 Determination of length compensation value using the example of a drill: Length 1/Z axis

Operating sequence

Get Comp.

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

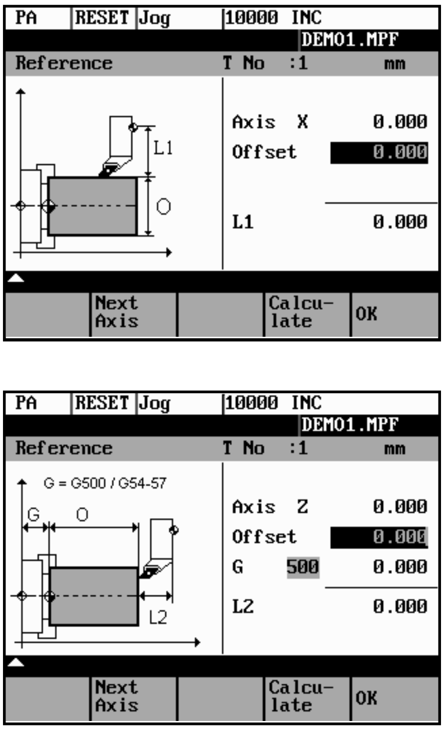


Fig.3-6 Compensation values window

- Enter offset if the tool edge cannot approach the zero point Gxx. If you work without zero offset, select G500 and enter offset.
- When the softkey Calculate is pressed, the control system determines the searched geometry length 1 or 2 depending on the preselected axis. This geometry is calculated on the basis of the approached actual position, the selected Gxx function and the entered offset value.

  The determined compensation value is stored.

3.2 Entering/modifying the zero offset

Functionality

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

Operating sequences

Parameter

Use the Parameter and Zero Offset softkeys to select the zero offset.

An overview of settable zero offsets appears on the screen .

Zero offset

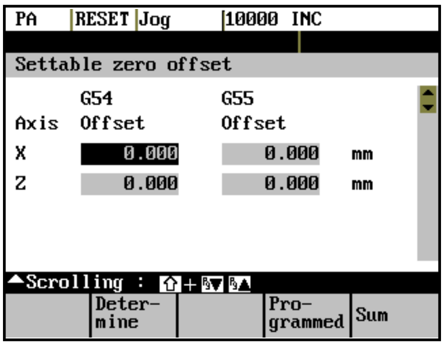


Fig.3-7 Zero offset window

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

0 ... 9

enter value(s).

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

Λ

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

Softkeys

Deter-mine

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

Pro-
grammed

A window with the programmed zero offset is displayed. The values in the window cannot be edited.

Sum

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

3.2.1 Determining the zero offset

Prerequisite

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

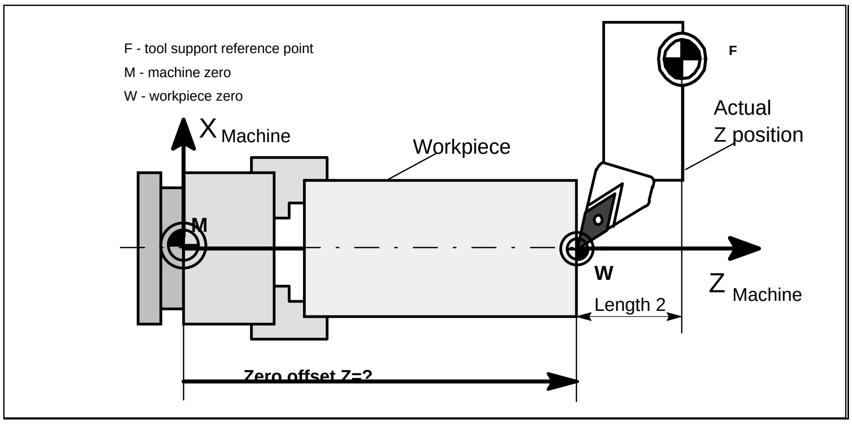


Fig.3-8 Determining the zero offset for the Z axis

Approach

- A zero offset can only be determined with a known tool. Enter the active tool in the dialog box. Press OK to take over the tool; the *Determine* window is then opened.
- The selected axis appears in the Axis area.

  The actual position of the tool support reference point (MCS) associated to the axis is displayed in the adjacent field.
- D number 1 is displayed for the tool edge.

  If you have entered the valid offsets for the used tool under a D number other than D1, enter that D number here.
- The stored tool type is displayed automatically.
- The effective length compensation value (geometry) is displayed.
- Select the sign (-, +) for calculating the length offset, or select “without” taking the length offset into account.

  A negative sign subtracts the length offset value from the actual position. The zero offset in the selected axis is the result.
- Offset

  If the tool does not reach zero, an offset can be entered to specify an additional offset to a point which can be approched by the tool.

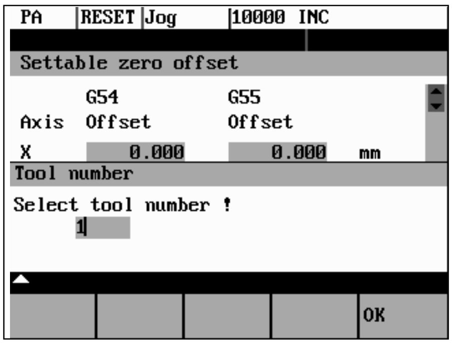


Fig.3-9 Select Tool screen form

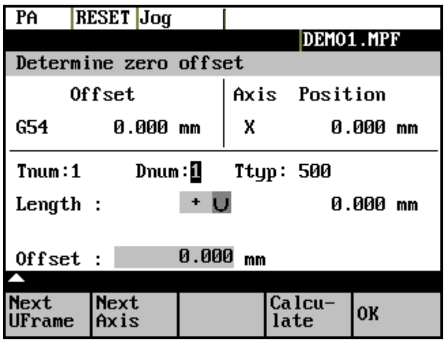


Fig.3-10 Determine zero offset form

Next UFrame	Softkey can be used to select the zero offsets G54 to G57. The selected zero offset is displayed on the selected softkey.
Next Axis	Selects the next axis.
Calcu-late	Pressing the Calculate softkey calculates the zero offset.
OK	Press the OK softkey to quit the window.

3.3 Programming the setting data - “Parameters” operating area

Functionality

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

Operating sequences

Parameter

Sett. data

Use the Parameter and Setting Data softkeys to select Setting Data.

The Setting Data softkey branches to another menu level in which various control options can be set.

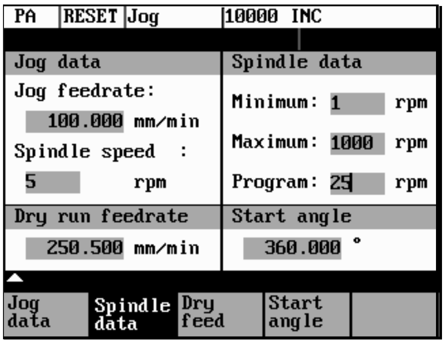
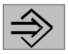


Fig.3-11 Setting data main screen

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

$ 0 … + 9

Enter the new value in the input fields.



Use Input or the cursor keys to confirm.

Softkeys

Jog data

This function can be used to change the following settings:

Jog feed

Feed value in Jog mode

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

Spindle

Spindle speed

Direction of rotation of the spindle

Spindle data

Minimum / Maximum

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

Programmed (LIMS)

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

Dry feed

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

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

Start angle

Start angle for thread cutting (SF)

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

3.4 R parameters – “Parameters” operating area

Functionality

All R parameters (arithmetic parameters) that exist in the control system are displayed on the R Parameters main screen as a list (see also Section 8.8 “Arithmetic Parameters /R Parameters”). These can be modified if necessary.

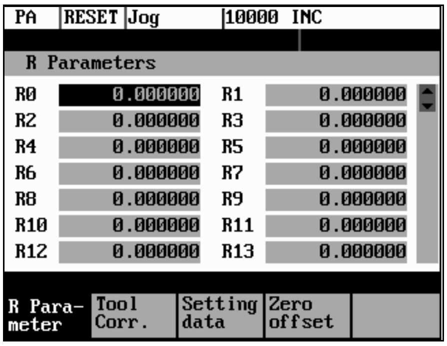


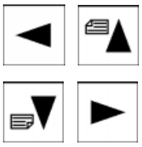
Fig.3-12 R Parameters window

Operating sequence

Parameters

R Para- meters

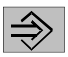
Use the Parameter and R Parameter softkeys



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

$ 0 … + 9

Enter value(s).



Press Input or use the cursor keys to confirm.

Manually Operated Mode 4

Preliminary remarks

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

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

4.1 Jog mode – “Machine” operating area

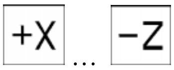
Functionality

In Jog mode, you can

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

Operating sequences

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



Press the appropriate key for the X or Z axis to traverse the desired axis.

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

If necessary use the override button key to set the traversing speed.

It can be adjusted by settable increments:

0%, 1%, 2%, 4%, 8%, 10%, 20%, 30%, 40%, 50%, 60%, 75%, 80%, 85%, 90%, 95%, 100%, 105%, 110%, 115%, 120%.

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

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

The Jog main screen displays position, feed and spindle values, including the feedrate override and spindle override, gear stage status as well as the current tool.

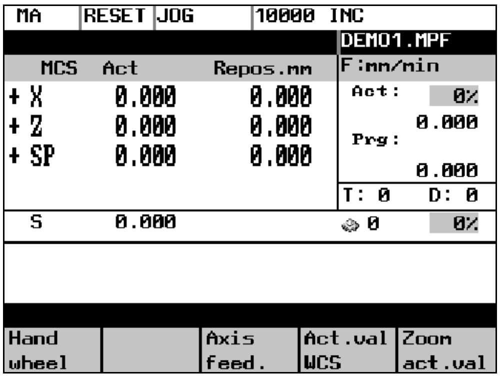


Fig.4-1 Jog main screen

Parameters

Table 4–1 Description of parameters in the *Jog* main screen

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

Softkeys

Hand-
wheel

Call the Handwheel window.

Axis feed

Interp./ feed

Call the Axis Feed or Interp. Feed window.

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

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

Act. val. WCS

Act.val. MCS

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

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

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

Zoom act.val.

Enlarged view of actual values.

Λ

Pressing Recall key , return to the next-higher menu level.

4.1.1 Assigning handwheels

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

Operating Sequence

Jog

In Jog mode, call the Handwheel window.

Hand–wheel

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

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

The symbol ☑ appears in the window.

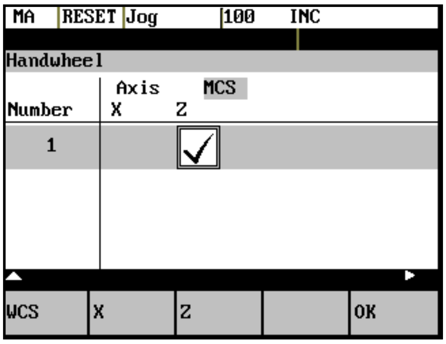


Fig.4-2 Handwheel window

WCS

MCS

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

OK

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

>

Menu extension

De-select

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

4.2 MDA mode (Manual Data Input) – “Machine” operating area

Functionality

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

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

Caution

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

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

Before NC-start of an input NC-program in the mode MDA is to wait till the message “Block store active” displays on the screen.

Operating sequences

Use the MDA key in the machine control panel area to select the MDA mode.

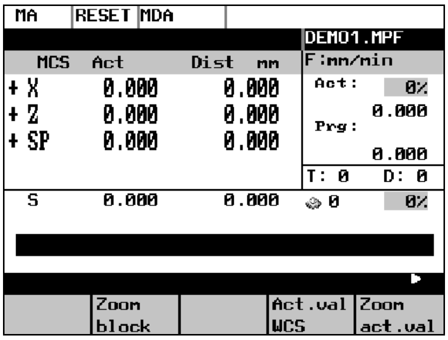
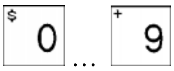


Fig.4-3 MDA main screen



Enter a block using the control keyboard.

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

Parameters

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

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

Softkeys

Act.val.
WCS

Act.val.
MCS

The actual values for the MDA mode are displayed as a function of the selected coordinate system.

There are two different coordinate systems, i.e. the machine coordinate system (MCS) and the workpiece coordinate system (WCS).

Zoom
act.val.

Enlarged view of the actual values

>

Menu extension

Axis
feed

Interp.
feed

Display of Axis Feed or Interp. Feed window

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

Zoom
G funct.

The G function window contains all active G functions whereby each G function is assigned a group and has its own fixed positon in the window.

Further G functions can be displayed using the Page Up or Page Down keys together with Shift key. Select Recall to quit the window.

Zoom block	The window shows the currently edited block full length.
Zoom M funct.	Opens the M function window to display all active M functions of the block.

Automatic Mode 5

Functionality

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

Preconditions

The preconditions for executing part programs are:

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

Operating sequence

Use the Automatic key to select the Automatic mode.

The Automatic main screen appears that displays the position, feed, spindle, override and tool values, the gear stage status as well as the current block.

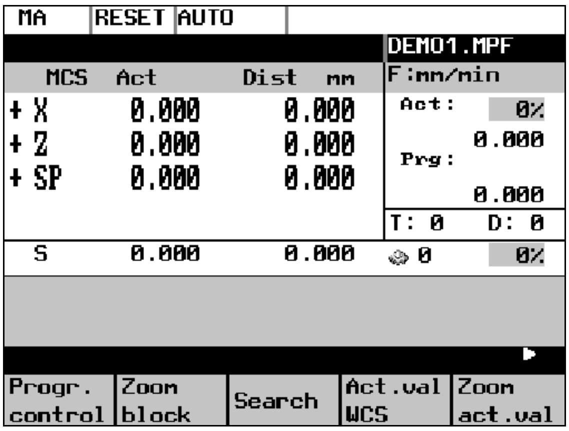


Fig.5-1 Automatic main screen

Parameters

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

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

Softkeys

Progr. control

The window to select Program Control (e.g. skip block, program test) appears on the screen.

Zoom block

The window shows the previous, current and next block full length. In addition, the names of the current program or subroutine are displayed.

Search

Use the Block Search function to jump to the desired point in the program.

Search

The Search softkey provides the functions “Find line” and “Find text”.

Interr. point

The cursor is positioned to the main program block of the breakpoint (”interrupt point”). The search target is automatically set in the subroutine levels.

Contin. search

Continue Search

Start B search

The Start B Search softkey starts the search process in which the same calculations are carried out as in normal program mode, but without axis movements.

The block search can be canceled by NC Reset.

Act.val. WCS

Act.val. MCS

The values of the machine or workpiece coordinate system are selected. The softkey label changes to Act. val. WCS or Act. val. MCS.

Zoom act.val.

Enlarged view of actual values

>

Menu extension

Axis feed

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

Interp. feed

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

Execute f. ext.

An external program is transferred into the control system via the RS232 interface and executed immediately by pressing NC START.

Zoom G Funkt.

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

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

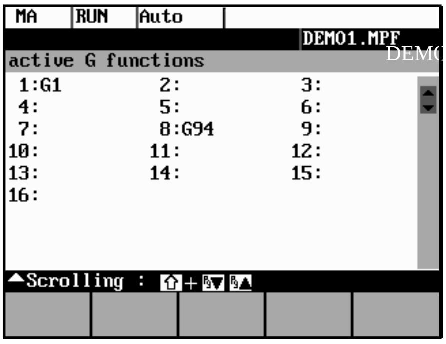


Fig.5-2 Active G functions window

Zoom M funct.

Opens the M Function window to display all active M functions.

5.1 Selecting/starting a part program – “Machine” operating area

Functionality

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

Operating sequence

Use the Automatic key to select the Automatic mode.

Programs

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

Position the cursor bar on the desired program.

Select

Use the Select softkey to select the program for execution. The selected program name appears in the Program Name screen line.

Progr. control

If necessary you can now make settings on program execution.

The following program control functions can be activated and deactivated:

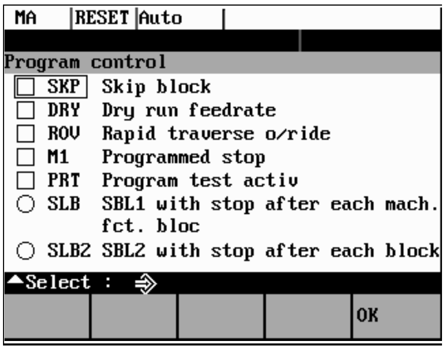


Fig.5-3 Program control window

The part program is executed when NC START is pressed.

5.2 Block search – “Machine” operating area

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

Search

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

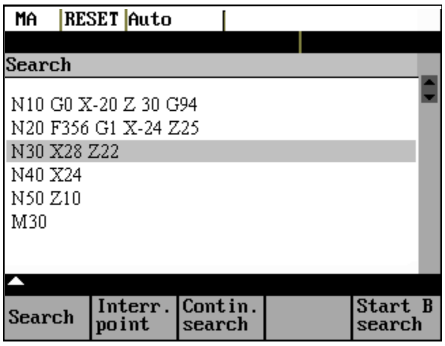


Fig.5-4 Block search window

Start B search

This function starts program advance and closes the Search window.

Result of the search The desired block is displayed in the Current Block window.

5.3 Stopping/aborting a part program – “Machine” operating area

Functionality Part programs can be stopped and aborted.

Operating Sequence

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

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

5.4 Repositioning after interruption – “Machine” operating area

Functionality

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

Operating sequence

Auto

Select the Automatic mode.

Search

Open the Block Search window to load the breakpoint.

Interr.
Point

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

Start B
search

A block search to the breakpoint is started.

Cycle Start

Continue execution of the program by NC START.

5.5 Program execution from external (RS232 interface)

Functionality

An external program is transferred to the control system via the RS232 interface and executed immediately by pressing NC START.

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

Operating sequence

Prerequisite: The control system is reset.
The RS232 interface is parameterized correctly (see Chapter 7) and not occupied by any other application (DataIn, DatatOut, STEP7).

Execute f. ext.

Press this softkey.

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

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

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

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

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

Note

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

5.6 Teach-in

Functionality

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

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

Operating sequence

Prerequisite:

— Teach-in option is set (display MD 278=4)

— The control system is either in the state Stop or Reset.

Programs

A list of all programs existing in the control system is displayed.

open

Pressing Open calls the editor for the selected program and opens the editor window.

>

Menu extension

Edit

Select

>

Menu extension

Teach In on

Select

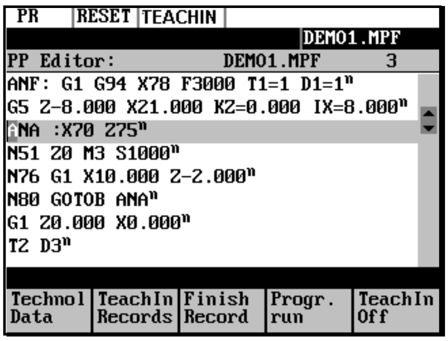


Fig.5-5 Teach in main screen

Softkeys

Technol . Data

Use this softkey to generate a block with technological data.

Use this screen form to enter

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

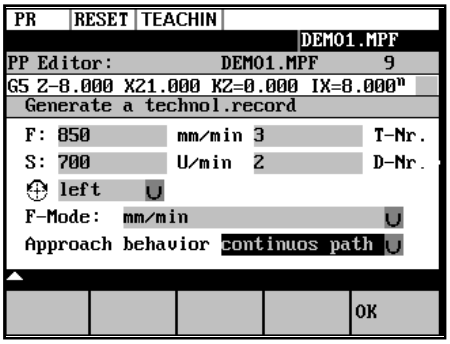


Fig.5-6 Input screen form for technological data

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

Teach In Records

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

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

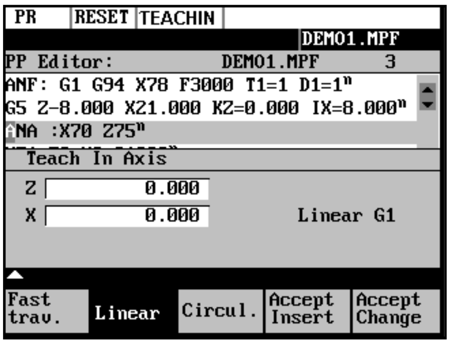


Fig.5-7 Teach in of NC blocks

Fast Trav.

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

Linear

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

Circul.

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

Accept Insert

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

Accept Change

Values are corrected in the block (accepted from the screen form) to which the cursor is positioned.

Use RECALL to return to the Teach In main screen. Any amendments you wish to make can be later inserted manually.

Finish Record

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

Progr. run

Use this softkey to traverse the programmed block.

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

Teach In Off

Use this softkey to turn off the submode Teach In.

Note

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

Example

Teaching a G5 block

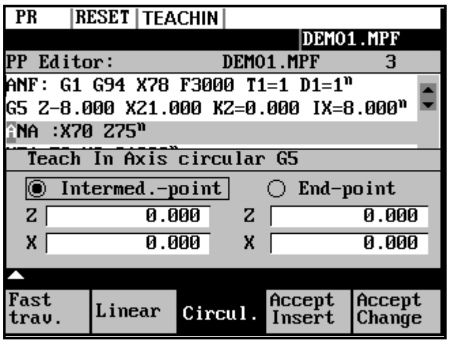


Fig.5-8 Teach in of a circular block

- The program block with G5 is selected by the cursor.
- Press the softkey Circul.

  The circle start point is the end point of the previous block.
- Approach to the intermediate point of the contour and press Accept Change.
- Approach to the end point of the contour and press Accept Change.

Part Programming 6

Functionality

This Section describes how to create a new part program.

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

Operating sequence

Pro- grams

You are in the main menu.

The Programming main screen appears.

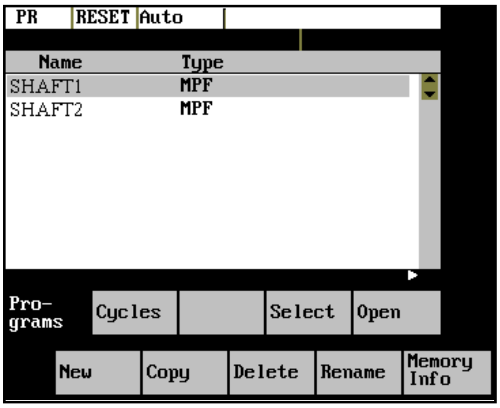


Fig.6-1 Programming main screen

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

Softkeys

Cycles

The Standard Cycles directory is displayed by pressing the Cycles softkey.

This softkey is only displayed if the operator has the appropriate access authorization.

Select

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

Open

Opens the files selected by the cursor for editing.

>

Menu extension

New

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

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

Copy

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

Delete

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

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

Rename

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

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

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

Memory Info

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

6.1 Entering a new program – "Program" operating area

Functionality

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

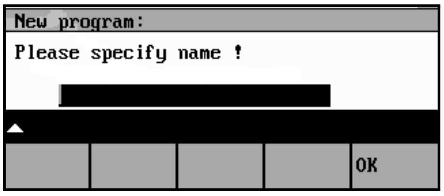


Fig.6-2 New program input screen form

Operating sequences

Program

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

New

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

U – … Z

Enter the new name.

OK

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

∧

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

6.2 Editing a part program – “Program” operating area

Functionality

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

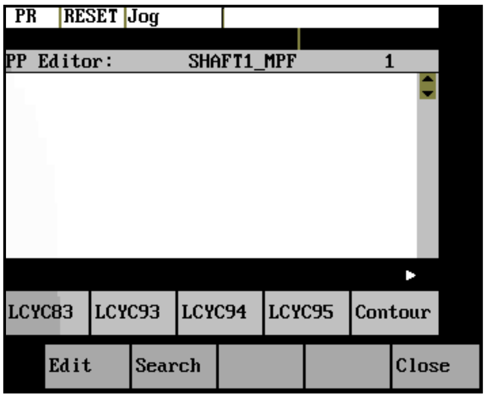


Fig. 6-3 Editor window

Operating sequence

Programs

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

Use the paging keys to select the program you wish to edit.

open

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

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

Softkeys

User-assignable softkeys

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

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

Contour

The contour functions are described in Section 6.3 ”Programming Support”.

>

Menu extension

Edit

Mark

This function selectes section of text up to the current cursor position.

Delete

This function deletes the selected text.

Copy

This function copies selected text to the clipboard.

Past

This function inserts text from the clipboard at the current cursor poisition.

Recomp. cycles

For re-compilation, the cursor must stand on the cycle call line in the program. The required parameters must be arranged directly in front of the cycle call and may not be separated by instruction or comment lines. The function decodes the cycle name and prepares the screen form with the respective parameters. If there are any parameters are outside the validity range, the function automatically uses standard values. When the screen form has been quitted, the original parameter block is automatically replaced by the corrected one.

Note: Only automatically generated blocks can be recompiled.

Note

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

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

>

Menu extension

Assign SK

This function can be used to change the assignment of the softkey functions 1 - 4.

For more detail description refer to Section NO TAG.

Search

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

Text

Type the text you wish to find in the input line and start the Search operation by selecting the OK softkey.

If the character string you have specified cannot be found in the program file, an error message appears that must be acknowledged with OK.

You can exit the dialog box without starting the search by selecting RECALL.

Line no.

Type the line number in the input line.

The search is started by pressing OK.

You can quit the dialog box without starting the search by selecting RECALL.

Contin. Search

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

Close

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

Editing cyrillic letters

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

Procedure

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

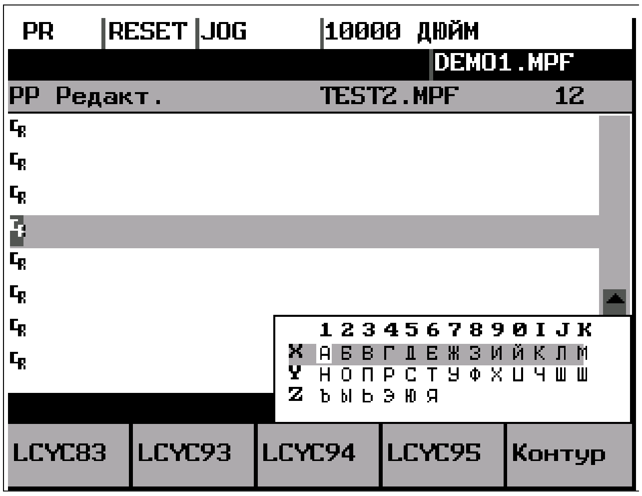


Fig.6-4

To select a character,

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

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

6.3 Programming support

Functionality

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

6.3.1 Vertical menu

Functionality

The vertical menu is displayed in the program editor.

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

Operating sequence

You are in the program editor.

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

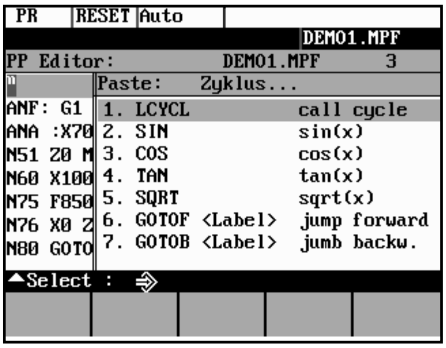


Fig.6-5 Vertical menu

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

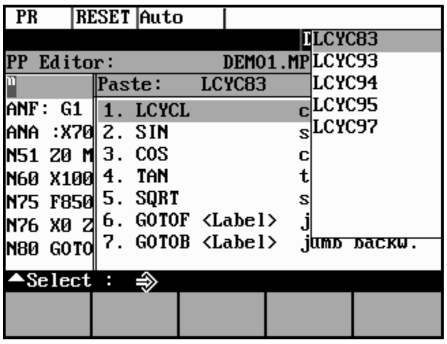


Fig.6-6 Vertical menu

Use the paging keys to browse through the list.

Confirm your entry by pressing **Input**.

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

6.3.2 Cycles

Functionality

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

Operating sequences

LCYC 93

LCYC 94

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

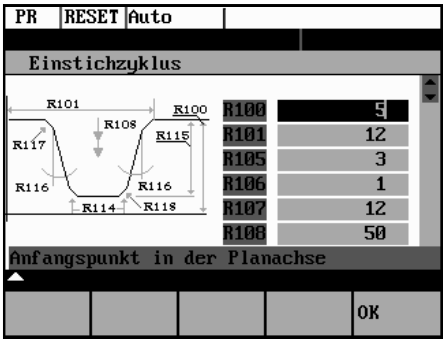


Fig.6-7

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

OK

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

6.3.3 Contour

Functionality

The control system provides you with various contour forms to assist you in creating part programs quickly and reliably. Enter the necessary parameters in the screen forms and confirm your inputs.

The contour screen forms can be used to program the following contour elements and contour sections:

- Straight section with specification of end point or angle
- Circle sector with specification of center point / end point
- Circle sector with specification of center point / opening angle
- Circle sector with specification of center point / radius
- Straight line/straight line contour section with specification of angle and end point
- Straight line/circle contour section with tangential transition; calculated from angle, redius and end point
- Straight line/circle contour section with any transition; calculated from angle, center point and end point
- Circle/straight line contour section with tangential transition; calculated from angle, radius and end point
- Circle/straight line contour section with any transition; calculated from angle, center point and end point.
- Circle/circle contour section with tangential transition; calculated from center point, radius and end point
- Circle/circle contour section with any transition; calculated from center point and end positon
- Circle - straight line - circle contour section with tangential transitions
- Circle - circle - circle contour section with tangential transitions

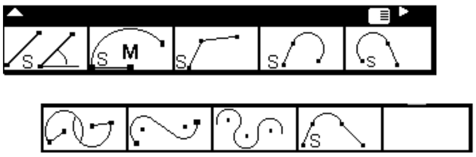
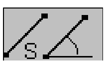


Fig.6-8

Softkeys



The sofkey functions branch to the contour elements.

Programming aid for programming straight line sections.

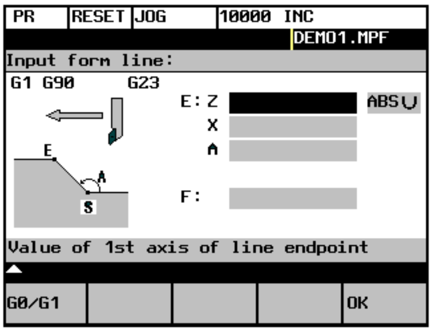


Fig.6-9

Enter the end point of the straight line.

G0/G1

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

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

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

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

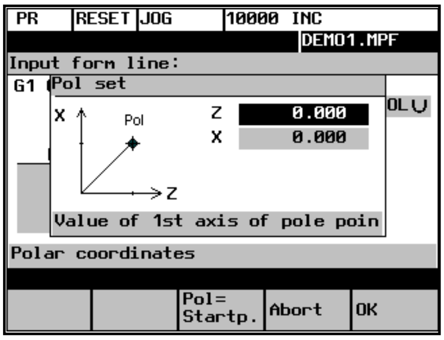


Fig.6-10

OK

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

Additional functions

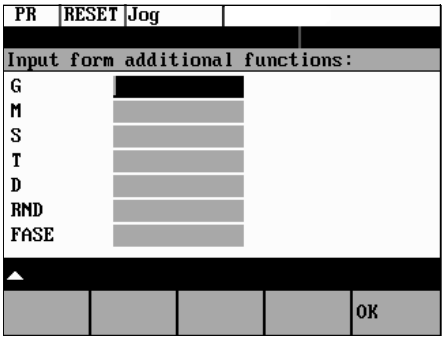


Fig.6-11 Additional functions screen form

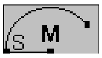
Enter additional commands in the fields. The commands can be separated by means of blanks, commas or semi-colons.

This screen form is available for all contour elements.

OK

The OK softkey transfers the commands to the part program.

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



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

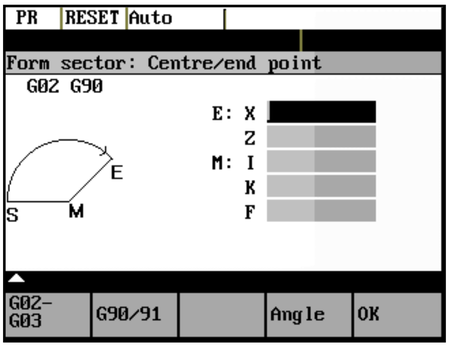


Fig.6-12

Enter the center point coordinates in the input fields.

To enter the coordinates, there are three variants:

- absolute
- incremental
- polar

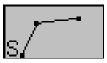
G2/G3

This softkey changes the direction of rotation from G2 to G3. G3 appears on the display.

When you press the softkey again, you will return to G2.

OK

Pressing the OK softkey will accept the block into the part program and will offer additional commands in another interactive screenform.



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

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

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

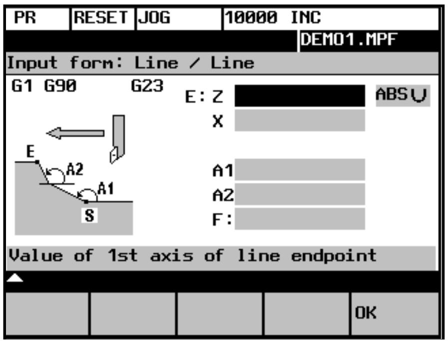


Fig. 6-13 Calculating the intersection point between two straight lines

Table 6–1 Input in the interactive screenform

End point of straight line 2	E	Specify the end point of the straight line.
Angle of straight line 1	A1	The angle must be specified in the CCW direction in the range between 0 and 360 degrees.
Angle of straight line 2	A2	The angle must be specified in the CCW direction in the range between 0 and 360 degrees.
Feedrate	F	Feedrate

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

To calculate intersection points with any transition angles, the POI softkey function will display the center point coordinates.

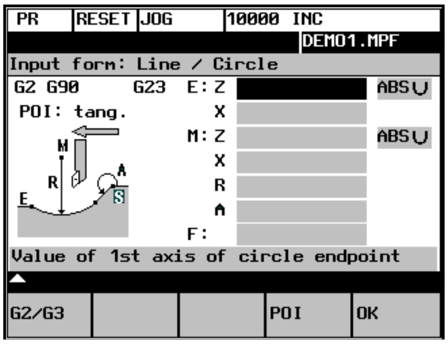


Fig. 6–14 Straight line - circle with tangential transition

Table 6–2 Input in the interactive screenform

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

G2/G3

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

G90/G91

The end point can be acquired either in the absolute dimension, incremental dimension or as polar coordinates.

The current setting is displayed in the interactive screenform.

POI

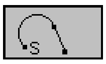
You can choose between tangential or any transition.

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

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

If there are several intersection points, the operator must select the desired intersection point from a dialog.

If a coordinate was not entered, the program tries to caluclate it from the existing information. If there are several possibilities, the operator must choose an appropriate possibility from the dialog.



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

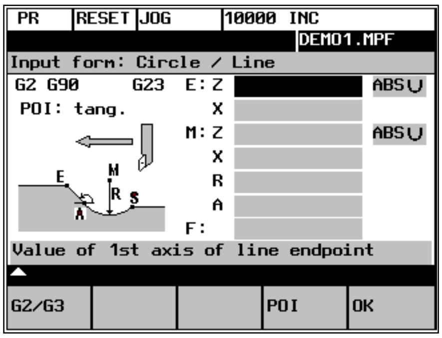


Fig. 6–15 Tangential transition

Table 6–3 Input in the interactive screenform

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

G2/G3

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

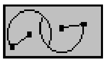
POI

Use this softkey to choose between tangential or any transition.

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

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

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



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

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

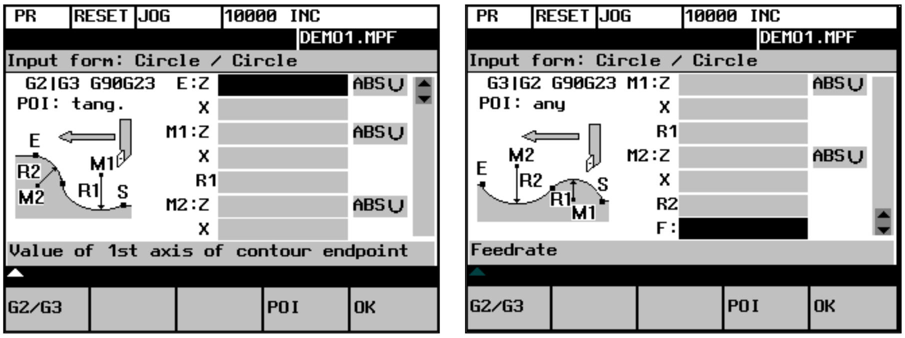


Fig. 6–16 Tangential transition

Table 6–4 Input in the interactive screenform

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

The points are specified depending on the previsouly selected caluclation method (absolute, incremental dimension or polar coordinates). Input fields no longer needed are hidden. If a value is omitted in the center point coordinates, the radius must be entered.

G2/G3

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

POI

Use this softkey to choose between tangential or any transition.

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

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

Selecting the intersection point

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

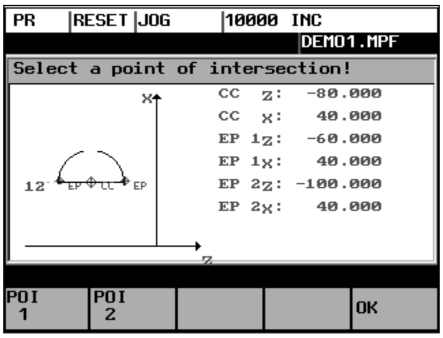


Fig. 6–17

POI 1

The contour is drawn using intersection point 1.

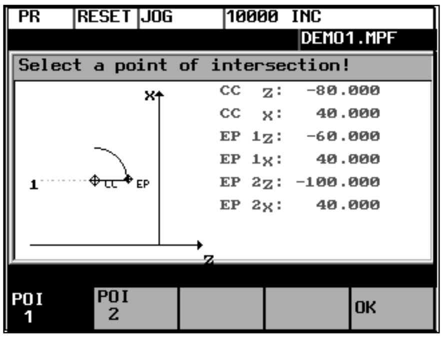


Fig. 6–18 Selection of intersection point 1

POI 2

The contour is drawn using intersection point 2.

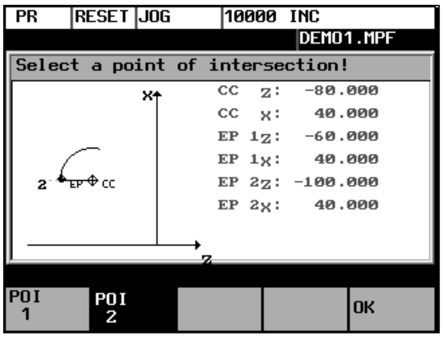
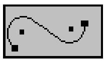


Fig. 6–19 Selection of intersection point 2

OK

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



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

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

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

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

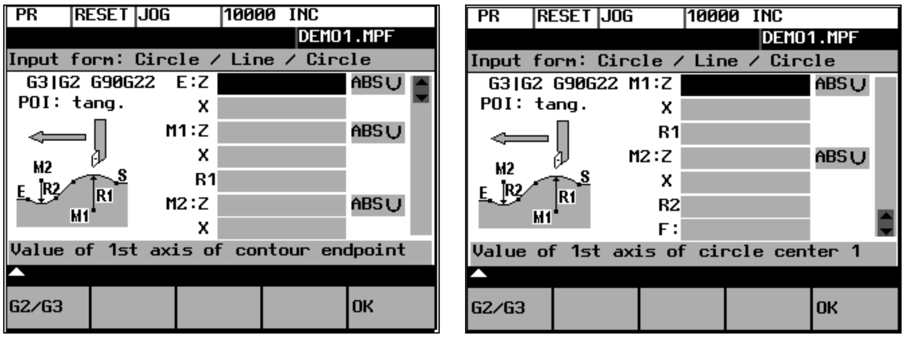


Fig. 6–20 Screenform for calculating the contour section ‘circle - straight line - circle’

Table 6–5 Input in the interactive screenform

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

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

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

G2/G3

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

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

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

Example DIAMON

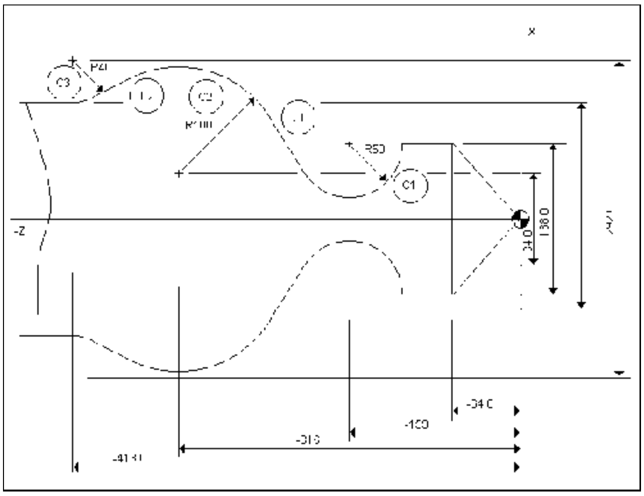


Fig.

Given:	R1	50 mm	
	R2	100 mm	
	R3	40mm	
	M1	Z -159	X 138
	M2	Z -316	X 84
	M3	Z -413	X 292

Starting point: The point X = 138 and Z = -109 mm (-159 -R50) is supposed as the starting point.

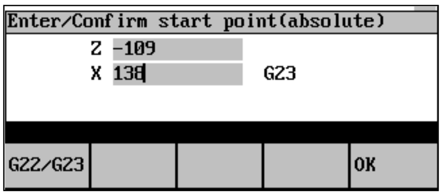


Fig. 6–22 Setting the starting point

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

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

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

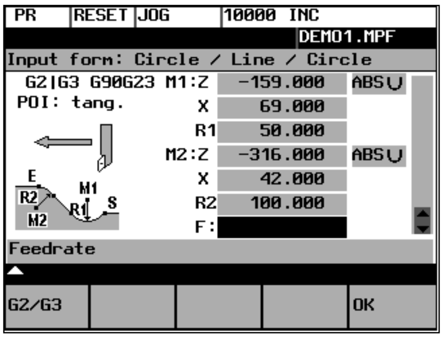


Fig. 6–23 Calling the screenform

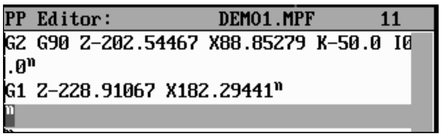


Fig. 6–24 Result of step 1

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

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

Now, call the screenform for calculating the contour section C2 - C3 again. The end point of the contour section are the coordinates Z=-413.0 and X=212.

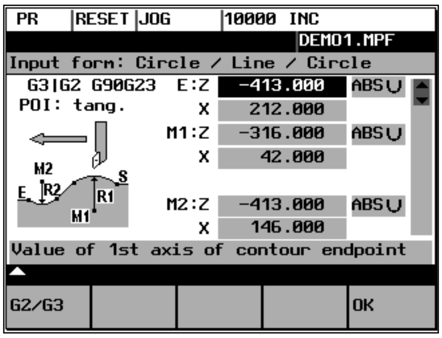


Fig. 6–25 Calling the screenform

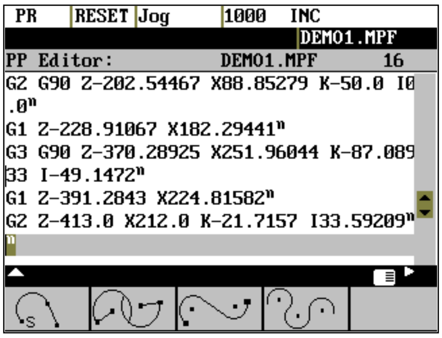
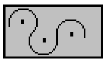


Fig. 6–26 Result of step 2



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

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

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

The selected setting is displayed in a help screen.

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

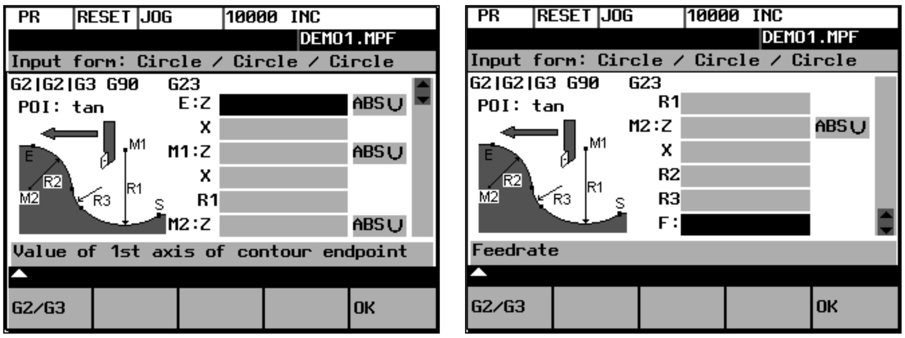


Fig. 6–27 Screenform for calculating the contour section ’circle - circle - circle

Table 6–6 Input in the dialog screenform

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

If the starting point cannot be deteremined from the previous blocks, the respective coordinates must be entered in the “Starting point” screenform.

G2/G3

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

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

Example DIAMON - G23

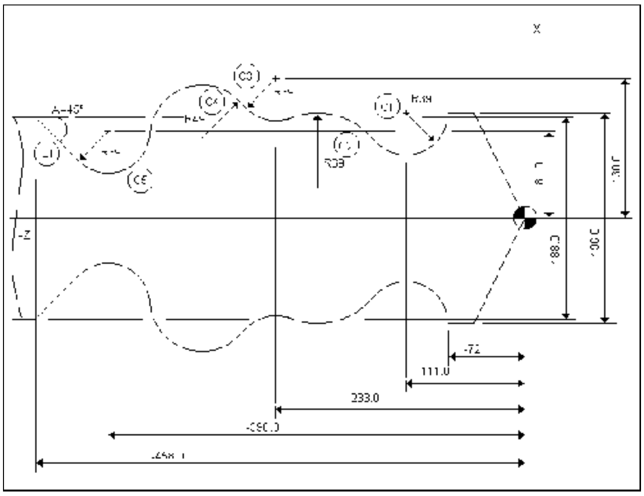


Fig.6-28

Given:	R1	39 mm	
	R2	69 mm	
	R3	39 mm	
	R4	49 mm	
	R5	39 mm	
	M1	Z -111	X 196
	M2	Z -233	X 260
	M3	Z -390	X 162

The coordinates Z -72, X 196 will be selected as the starting point.

After you have confirmed the starting point, use the screenform to caluclate the contour section C1 - C3. The end point is left open, since the coordinates.

Use softkey 1 to set the direction of rotation of the two circles (G2 - G3 - G2) and to fill out the parameter list.

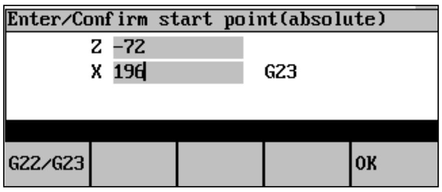


Fig. 6–29 Setting the starting point

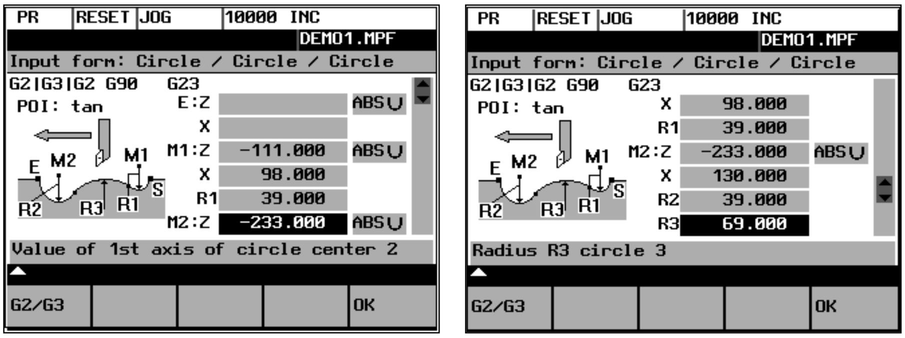


Fig. 6–30 Screenform 'circle - circle - circle'

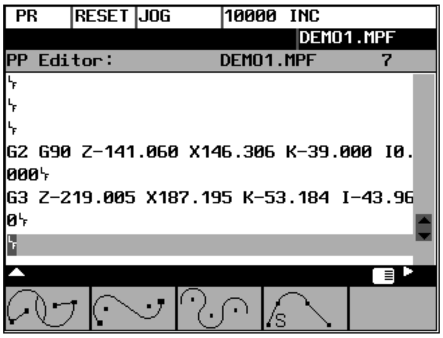
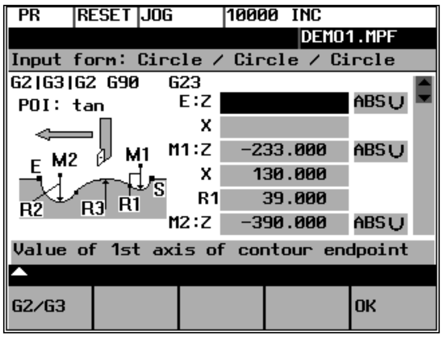


Fig.6-31 Result of step 1

In the second step, screenform is used to calculate the contour section C3 - C5. For calculation, select direction of rotation G2 - G2 - G3. Starting point is the end point of the first caluclation.



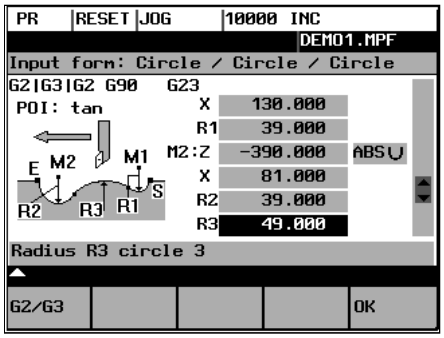


Fig. 6–32 Screenform 'circle - circle - circle'

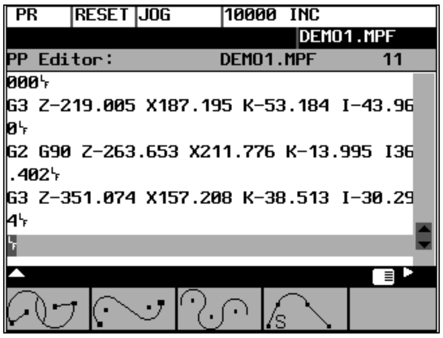


Fig. 6–33 Result of step 2

The result provided by the function is the intersection point between circle sector 4 and circle sector 5 as the end point.

To caluclate the tangential transition between C5 and L1, the circle-straight line screenform is used.

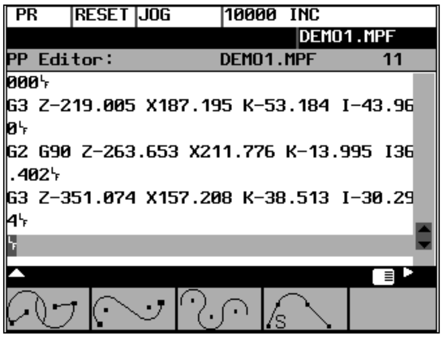


Fig. 6–34 Screenform 'circle - straight line'

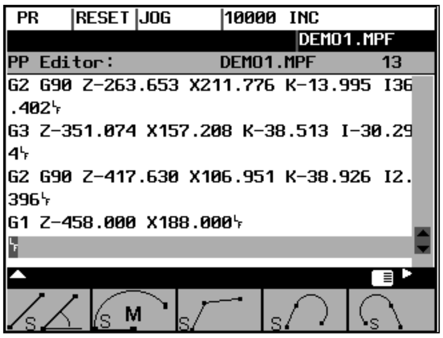
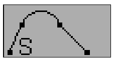


Fig. 6–35 Result of step 3



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

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

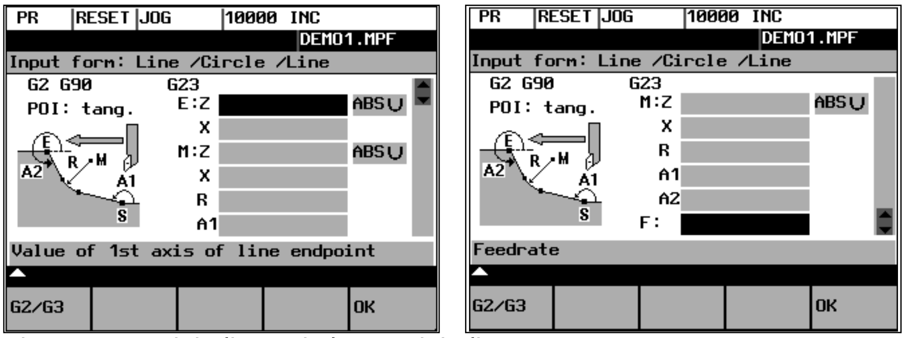


Fig. 6–36 Straight line - circle - straight line

Table 6–7 Input in the interactive screenform

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

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

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

6.3.4 Free softkey assignment

Assign SK

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

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

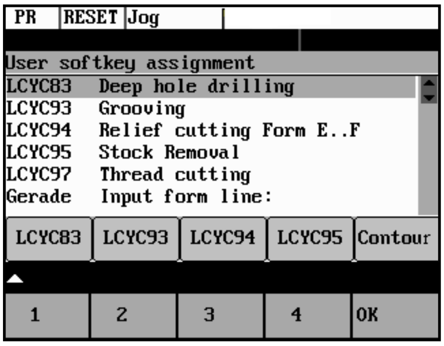


Fig.6-37

Position the cursor on the element you wish to assign.

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

OK

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

Services and Diagnosis 7

7.1 Data transfer via the RS232 Interface

Functionality

You can use the RS 232 interface of the CNC to output data (e.g. part programs) to an external data storage medium or to read in them from there. The RS232 interface and the data storage device must be matched to one another. The control system provides an interactive screenform in which you can set the special data for your storage medium.

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

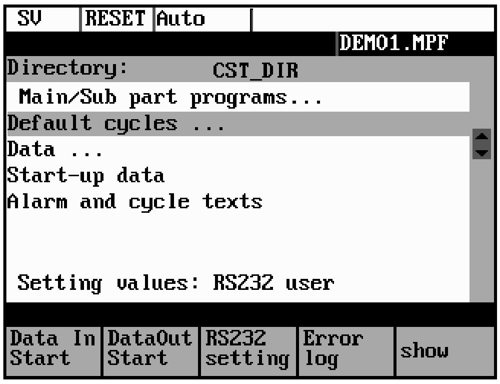


Fig.7-1 Service main screen

File types

Provided the access authorization is set, files can be read in or read out via the RS232 interface.

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

- Data
  - Option data
  - Machine data
  - Setting data
  - Tool offsets
  - Zero offsets
  - R parameters
- Part programs
  - Part programs
  - Subroutines

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

Operating Sequence

Service

Use the Service softkey to select the Services operating area.

Softkeys

Data In Start

This key starts reading in data.

DataOut Start

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

RS232 setting

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

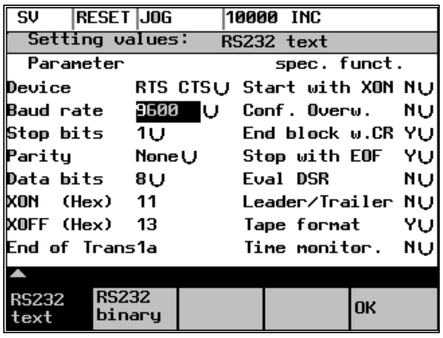


Fig. 7–2 Interface settings

Position the cursor on the desired data.

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

Activating the transmission log

These softkeys are intended to adapt the RS232 interface to the transmission log. 2 logs are set by default.

RS232 text

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

RS232 binary

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

OK

Press this softkey to save your settings.
Select RECALL to exit the window without saving your settings.

Error log

A log is output for the transferred data.

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

Transmission messages:

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

Show

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

>

Menu extension

Execut f. ext.

An external program is transferred to the control system via the RS232 interface and executed immediately by pressing NC Start (see Section 5.5).

Note

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

7.1.1 Interface parameters

Table 7–1 Interface parameters

Parameter	Description
Device type	• XON/XOFF One possible method of controlling the transmission operation is to use the XON (DC1, DEVICE CONTROL 1) and XOFF (DEVICE CONTROL 2) control characters. As soon as the buffer of the I/O device is full, it returns XOFF, and as soon as it can receive data again, it sends XON. • RTS/CTS The RTS signal (Request to Send) controls the send operation of the data transmission device. Aktive signal: Send data Passive signal: Do not exit send mode until all transferred data have been sent. The CTS signal is the acknowledgment signal for RTS and indicates that the data transmission device is ready to send.
XON	This is the character that is used to start transmission. It is effective only for device type XON/XOFF.
XOFF	This is the character with which data transmission is stopped.
End of transmission	This is the character that signals end of transmission of a text file. The special function “Stop with end of transmission” character may not be active if binary data are to be transferred.
Baud rate	Interface speed settings 300 baud 600 baud 1200 baud 2400 baud 4800 baud 9600 baud 19200 baud 38400 baud
Data bits	Number of data bits for asynchronous transmission. Input: 7 data bits 8 data bits (default)
Stop bits	Number of stop bits for asynchronous transmission. Input: 1 stop bit (default) 2 stop bits
Parity	Parity bits are used to detect errors. These are added to the coded character in order to obtain either an even or odd number of positions set to “1”. Input: No parity (default) Even parity Odd parity

7.1.2 Special functions

Table 7–2 Special functions

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

7.1.3 Interface parameterization

Please find examples for setting the RS232 interface below.

Start-up data

Settings for transferring archives with the start-up data

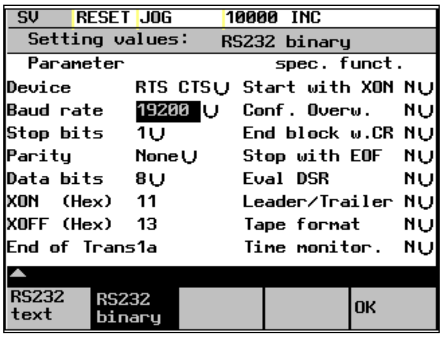


Fig. 7–3

Punched-tape input / output

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

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

Device type:	RTS/CTS
XON:	0
XOFF:	0
End of transm.:	0
Baud rate:	9600 baud
Data bits:8	
Stop bits:	2
Parity:	No parity

	Start with XON
	Overwrite with confirmation
X	Ends of block with CR LF
	Stop at transmission end
X	Evaluate DSR signal
	Leader and trailer
X	Tape format
X	Timeout monitoring

Parameters for a serial printer

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

Device type:	RTS/CTS
XON:	11(H)
XOFF:	13(H)
End of transm.:	1A(H)
Baud rate:	9600 baud
Data bits:8	
Stop bits:	1
Parity:	No parity

	Start with XON
	Overwrite with confirmation
X	End of block with CR LF
	Stop at transmission end
X	Evaluate DSR signal
X	Leader and trailer
X	Tape format
X	Timeout monitoring

7.2 Diagnosis and start-up – ”Diagnostics” operating area

Functionality

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

Operating sequence

Diagnostics

Selecting the Diagnostics softkey will open the Diagnostics main screen.

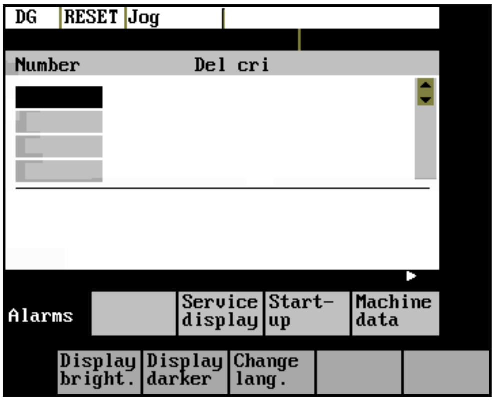


Fig.7-4 Diagnostics main screen

Softkeys for diagnostic functions

Alarms

This window displays all pending alarms line by line, starting with the alarm with the highest priority.

Alarm number, cancel criterion and error text are displayed. The error text refers to the alarm number on which the cursor is positioned.

Explanations with regard to the screenform above:

- Number

The “Number” item displays the alarm number. The alarms are displayed in chronological sequence.

- Cancel criterion

The symbol of the key required to reset the alarm is displayed for every alarm.

— Switch the device off and on again.

— Press the RESET key.

— Press the “Acknowledge alarm” key.

— Alarm is reset by NC START.

- Text The alarm text is displayed.

Service display

The Service Axes window appears on the screen.

Service axes

The window displays information about the axis drive.

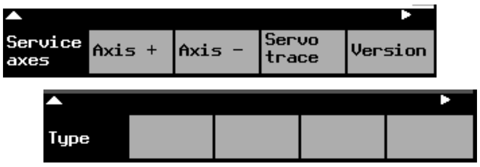


Fig. 7–5 The “Service Axes” window

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

Servo trace

To optimize the drives, an oscillograph function is provided for graphical representation of the velocity setpoint. The velocity setpoint corresponds to the ± 10V interface.

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

The following functions can be used to analyze the result:

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

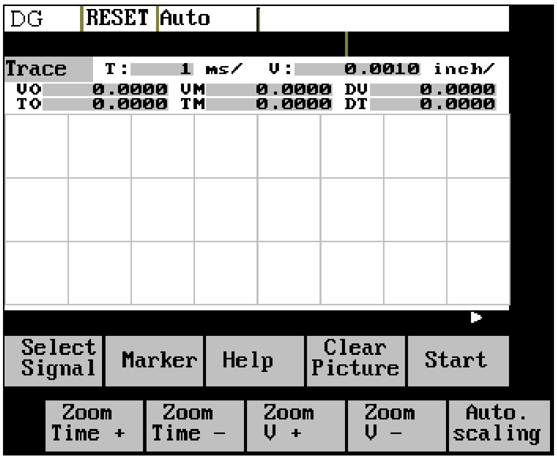


Fig. 7–6 The “Servo Trace” main screen

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

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

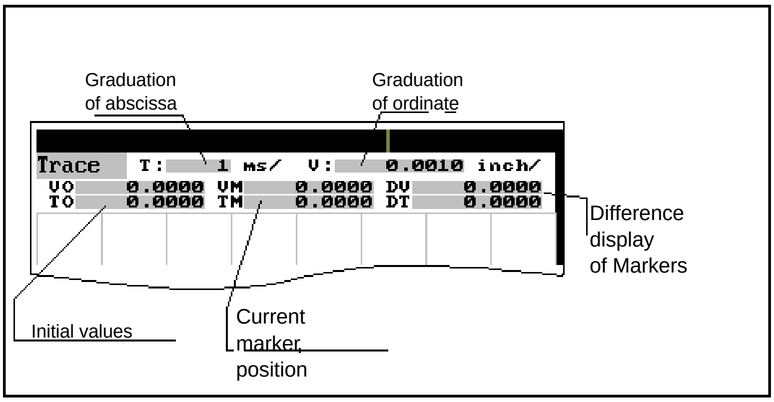


Fig. 7–7 Meaning of the fields

Select signal

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

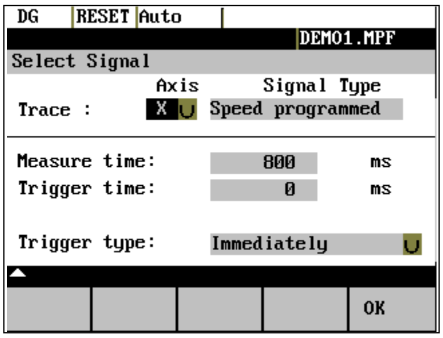


Fig. 7–8 Signal selection

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

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

- Selecting the trigger condition: Position the cursor on the Trigger Condition field and select the condition using the toggle key.

— No trigger, i.e. the measuring starts immediately after pressing the Start softkey.

— Negative edge

— Exact stop fine reached

— Exact stop coarse reached

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

Marker

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

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

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

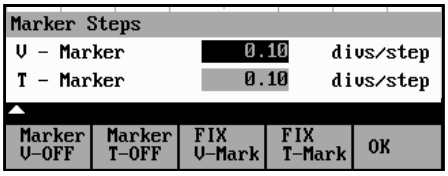


Fig. 7–9 Setting the markers

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

Help

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

Start

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

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

Stop

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

Zoom Time +

Zoom Time –

The scaling changes in the following steps:

1, 2, 5, 10, 20, 50, 100, 200, 500, 1,000 ms/div.

Zoom V +

Zoom V –

The horizontal scaling changes in the following steps:

0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500, 1,000, 5,000 unit/ div

Auto. scaling

This function calculates the vertical scaling from the peak values.

Version

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

Type

displays the control type

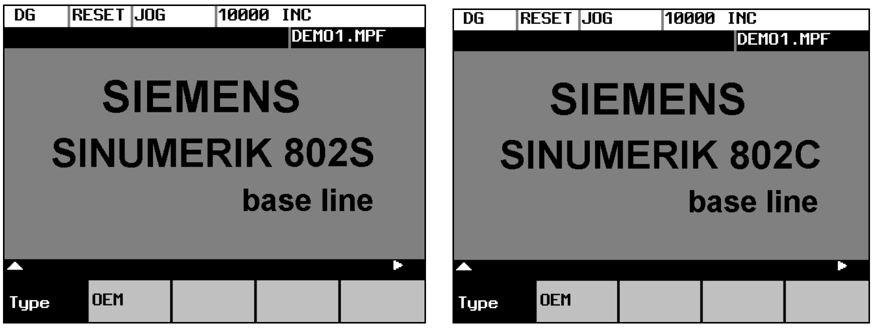
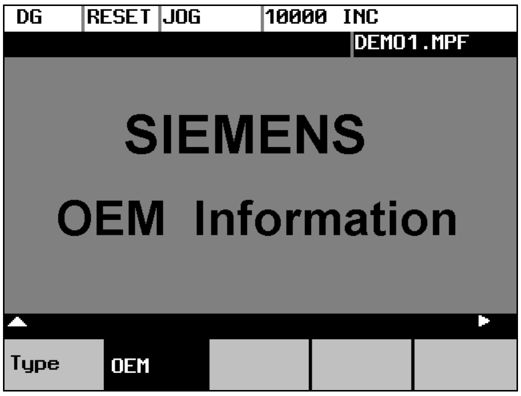


Fig. 7–10 Control type

OEM

displays the OEM picture here.



Softkeys for start-up functions

Note

See also Technical Manual

Start-up

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

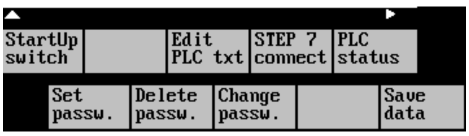


Fig. 7–11

Start-up switch

Start-up switch

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

Caution

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

NC

Selecting the power-up mode of the NC.

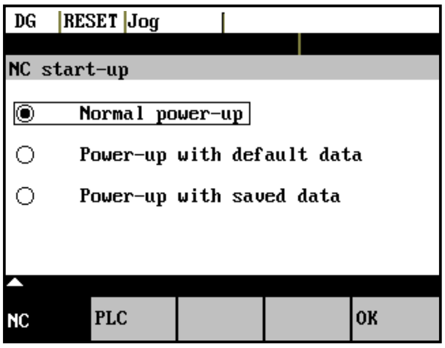


Fig. 7–12 NC Start-up

PLC

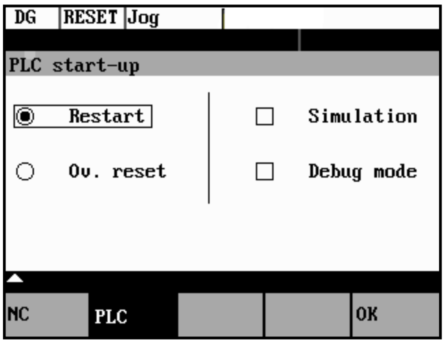


Fig. 7–13 PLC start-up

The PLC can be started in the following modes:

- Restart
- General reset

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

- subsequent simulation or
- subsequent debugging mode.

OK

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

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

Edit PLC txt

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

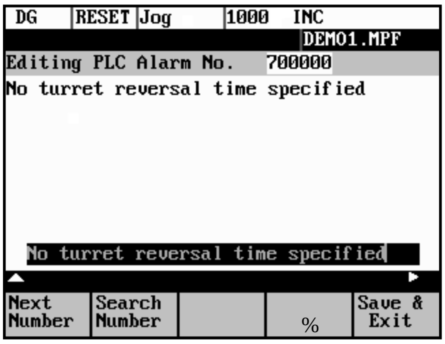


Fig. 7–14 Screenform for editing a PLC alarm text

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

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

Next Number

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

Search Number

This function selects the entered number for editing.

Save & Exit

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

Recall

The editor is quitted without saving the changes.

Editing Chinese characters

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

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

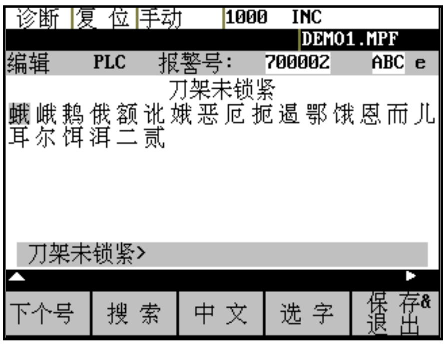


Fig. 7–15 screenform for editing a PLC alarm text in Chinese

The following softkey functions are realized:

Next Number

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

Search Number

This function selects the entered number for editing.

Change Mode

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

Choose Char

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

Save & Exit

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

Recall

The editor is quitted without saving the changes.

STEP 7 connect

The S7-Conn menu can be used to link the PLC with the external programming package S7-200.

If the RS232 interface is already occupied by the data transfer, you can link the control system with the programming package only when the transmission is completed.

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

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

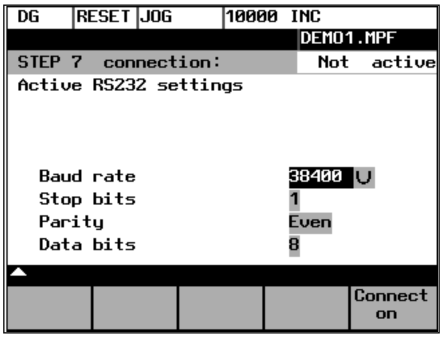


Fig. 7–16 S7-200 Connection

Conn. on

Conn. off

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

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

Press RECALL to quit the menu.

PLC status

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

It is possible to display 6 operands simultaneously.

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

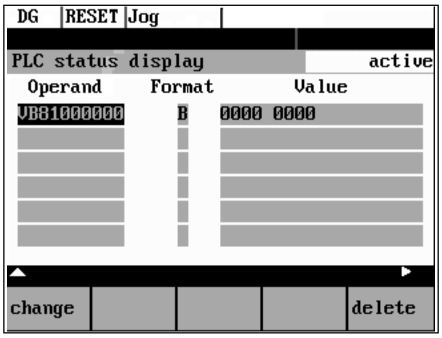


Fig. 7–17 PLC status display

There are further softkeys provided under this menu item.

- Edit

Cyclic updating of the values is interrupted. You can then edit the operand values.

- Cancel

Cyclic updating continues without the entered values being transferred to the PLC.

- Accept

The entered values are transferred to the PLC; cyclic updating continues.

- Delete

All operands are deleted.

- Operand +

The address of the operand can be incremented in steps of 1.

- Operand –

The address of the operand can be decremented in steps of 1.

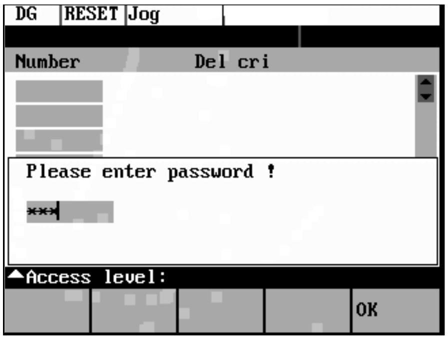
Set password

Set password

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

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

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



Enter the password.

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

The password is set when you press the OK softkey.

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

Delete password

The access authorization is reset.

Change password

Change password

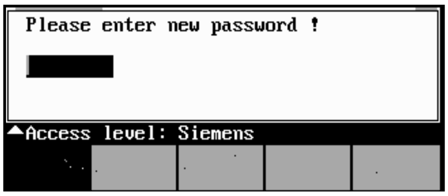


Fig. 7–18

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

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

The system asks you to confirm the new password again.

Press OK to complete the password change.

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

Save data

Save data

This function saves the contents of the volatile memory to a non-volatile memory area.

Prerequisite: No program is currently being run.

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

Softkeys for service functions

Machine data

Machine data (see also Technical Manual)

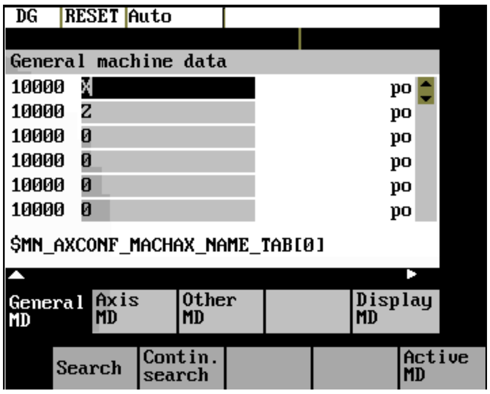


Fig.7-19

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

Units	userdef	User-defined
	M/s**2	Meters per second
	U/s**3	Revolutions per second
	S	Second
	Kgm**2	Moment of inertia
	MH	Inductivity
	Nm	Torque
	µs	Microseconds
	µA	Microamperes
	µVs	Microvolt seconds
Effective ness	So	Effective immediately
	Cf	With confirmation
	Re	Reset
	Po	Power ON

General MD

General machine data

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

Axis MD

Axis-specific machine data

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

Fig.7-20

The data of the axis are displayed.

Other MD

Other machine data

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

Display MD

Display machine data

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

Search

Search

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

The cursor jumps to the target data.

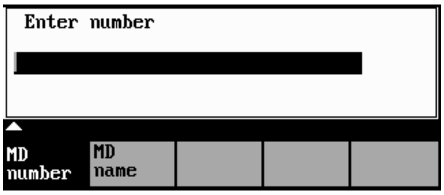


Fig. 7–21

Continue search

The search for the next number or name continues.

Axis +

Axis –

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

Active MD

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

Display bright.

Brightness

Display darker

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

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

Change lang.

Switching the language

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

Programming 8

8.1 Fundamentals of NC programming

8.1.1 Program structure

Structure and contents

The NC program consists of a sequence of blocks (see Table NO TAG).

Each block constitutes a machining step.

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

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

Table 8–1 NC program structure

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

Program names

Every program has its own program name.

Note

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

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

Example: SHAFT52/

8.1.2 Word structure and address

Functionality/structure

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

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

- an address character

The address character is generally a letter,

- and a numerical value.

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

A positive sign (+) can be omitted.

	Word	**Word**	**Word**
	Address ⋮ Value	Address ⋮ Value	Address ⋮ Value
Example:	**G1**	**X–20.1**	**F300**
Explanation:	Traverse with linear interpolation	Path or end position for X axis: -20.1 mm	Feed: 300 mm/min

Fig.8-1 word structure

Several address characters

A word may also contain several address letters. In such cases, however, an “=” sign must be inserted to assign the numerical value to the address letters.
Example: CR=5.23

8.1.3 Block structure

Functionality

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

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

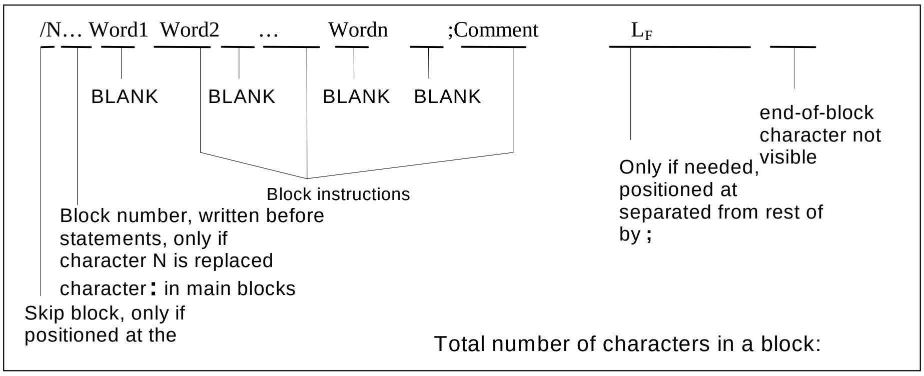


Fig.8-2 Diagram of block structure

Word sequence

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

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

Note with regard to the block numbers

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

Block skipping (see Fig. 5–3)

Program blocks that must not be executed during every program run can be marked with a slash “ / ” in front of the block number word.
Block skipping is activated by means of an operator input or by the interface control (signal). A program section can be skipped by skipping several successive blocks with “ / ”.
If block skipping is active during program execution, none of the blocks marked with “ / ” is executed. Any statements contained in such blocks are ignored. The program continues at the next block not marked.

Comment, remark

Comments (remarks) can be used to explain the staements in the blocks of a program.
Comments are displayed together with the other contents of the block in the current block display.

Programming example

N10	;G&S Order No. 12A71
N20	;Pump part 17, Drawing No.: 123 677
N30	;Program created by Mr. Adam Dept.TV 4

```
:50 G17 G54 G94 F470 S20 D0 M3    ;Main block
N60 G0 G90 X100 Y200
N70 G1 Y185.6
N80 X112
/N90 X118 Y180                    ;Block can be skipped
N100 X118 Y120
N110 X135 Y70
N120 X145 Y50
N130 G0 G90 X200
N140 M2                           ;End of program
```

8.1.4 Character set

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

Letters

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

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

Digits

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

Printable special characters

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

Non-printable special characters

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

8.1.5 Overview of instructions

Address	Meaning	Value assignment	Information	Programming
D	Tool compensation number	0 ... 9, integers only, without sign	Contains compensation data for a particular tool T... ; D0->compensation values= 0, max. 9 D numbers for one tool	D...
F	Feedrate (in combination with G4, the dwell time is also programmed under F)	0.001 ... 99 999.999	Tool/workpiece path velocity in mm/min or mm/rev depending on whether G94 or G95 is programmed	F...
G	G function (preparatory function)	Only specific integer values	The G functions are divided into G groups. Only one G function of a G group can be programmed in any one block. A G function can be modal (until canceled by another function of the same group) or non-modal - it is only active for the block in which it is programmed. G group:	G...
G0	Linear interpolation with rapid traverse		1: Motion commands (interpolation type)	G0 X... Z...
G1 *	Linear interpolation with feed			G1 X... Z... F...
G2	Circular interpolation in clockwise direction			G2 X... Z... I... K... F... ;center point and end point G2 X... Z... CR=... F... ;Radius and end point G2 AR=... I... K... F... ;Angle of aperture and center point G2 AR=... X... Z... F... ;Angle of aperture and end point
G3	Circular interpolation in counterclockwise direction			G3 .... ;otherwise as for G2
G5	Circular interpolation via interpolation point			G5 X...Z... IX=...KZ=... F...
G33	Thread cutting with constant pitch		Modal	G33 Z... K... SF=... ;Cylindrical thread G33 X... I... SF=... ;Cross thread G33 Z... X... K... SF=... ;Tapered thread, path greater in Z axis than in X axis G33 Z... X... I... SF=... ;Tapered thread, path greater in X axis than in Z axis

Address	Meaning	Value assignment	Information	Programming
G4	Dwell time		2: Special movements, non modal	G4 F... or G4 S.... ;in separate block
G74	Reference point approach			G74 X...Z... ;in separate block
G75	Fixed point approach			G75 X... Z... ;in separate block
G158	Programmable offset		3: Write to memory non modal	G158 X...Z... ;in separate block
G25	Lower spindle speed limit			G25 S... ;in separate block
G26	Upper spindle speed limit			G26 S... ;in separate block
G17	(required for center drilling)		6: Plane selection	
G18 *	Z/X plane			
G40 *	Tool radius compensation OFF		7: Tool radius compensation modal	
G41	Tool radius compensation left of contour			
G42	Tool radius compensation right of contour			
G500 *	Settable zero offset OFF		8: Settable zero offset modal	
G54	1st settable zero offset			
G55	2nd settable zero offset			
G56	3rd settable zero offset			
G57	4th settable zero offset			
G53	Non–modal suppression of settable zero offset		9: Suppression of settable zero offset non modal	
G60 *	Exact positioning		10: Approach behaviour modal	
G64	Continuous path mode			
G9	Non-modal exact stop		11: Non–modal exact positioning non–modal	
G601 *	Exact positioning window fine for G60, G9		12: Exact positioning window modal	
G602	Exact positioning window coarse for G60, G9			
G70	Dimensions in inches		13: Dimensions in inches/metric values modal	
G71 *	Dimensions in metric values			
G90 *	Absolute dimensions		14: Absolute/incremental dimension modal	
G91	Incremental dimensions			
G94	Feedrate F in mm/min		15: Feedrate/spindle modal	
G95 *	Feedrate F in mm/revolution of spindle			
G96	Constant cutting speed for turning ON (F in mm/rev, S in m/min)			G96 S... LIMS=... F...
G97	Constant cutting speed for turning OFF			
G450 *	Transition circle		18: Behaviour at corners with tool radius compensation modal	
G451	Point of intersection			
G22	Radius input		29: Radius/diameter input modal	
G23 *	Diameter input			
The functions marked with an * are active from the beginning of the program (with the version of the control supplied unless otherwise programmed).				

Address	Meaning	Value assignment	Information	Programming
I	Interpolation parameter	± 0.001 ... 99 999.999 thread: 0.001 ... 2000.000	For X axis, meaning depends on whether G2,G3->circle center or G33->thread pitch has been programmed	see G2, G3 and G33
K	Interpolation parameter	± 0.001 ... 99 999.999 thread: 0.001 ... 2000.000	For Z axis, otherwise as for I	see G2, G3 and G33
L	Subroutine, name and call	7 decimal places, integers only, without sign	Instead of a user–defined name L1 ...L9999999 can also be selected; this also calls the subroutine in its own block Caution: L0001 is not the same as L1	L.... ;in separate block
M	Miscellaneous function	0 ... 99 integers only, no sign	E.g. to trigger actions such as ”coolant ON”, max. 5 M functions in one block,	M...
M0	Programmed stop		Machining is stopped at the end of a block containing M0, operation is continued with ”START”	
M1	Optional stop		As for M0, but operation only stops if a special signal has been given	
M2	End of program		Programmed in the last block to be executed	
M30	-		Reserved, do not use	
M17	-		Reserved, do not use	
M3	Spindle clockwise rotation			
M4	Spindle counterclockwise rotation			
M5	Spindle stop			
M6	Tool change		Only if activated with M6 in machine data,otherwise tool change performed directly with T command	
M40	Automatic gear stage switchover			
M41 bis M45	Gear stage 1 to gear stage 5			
M70	-		Reserved, do not use	
M...	Other M functions		This functionality is not predefined in the control and can therefore be assigned by the machine manufacturer	
N	Block number - subblock	0 ... 9999 9999 integer only, no sign	Can be used with a number to identify blocks, programmed at the beginning of a block	E.g.: N20
:	Block number - main block	0 ... 9999 9999 integer only, no sign	Special identification for blocks - used instead of N..., this block should contain all the instructions for a complete set of subsequent machining operations	E.g.: 20
P	Number of subroutine passes	1 ... 9999 integer only, no sign	Programmed in the same block as a subroutine to be called several times, e.g.: N10 L871 P3 ; called three times	E.g.: L781 P... ;in separate block

Address	Meaning	Value assignment	Information	Programming
R0 to R249	Arithmetic parameter	± 0.0000001 ... 9999 9999 (8 decimal places) or with exponent: ± (10^{-300} ... 10^{+300})	R0 to R99 -user assignable R100 to R249 -transfer parameters for machining cycles	
Arithmetic functions			In addition to the 4 basic arithmetic operations + - * / the following arithmetic functions are also available:	
SIN( )	Sine	in degrees		E.g.: R1=SIN(17.35)
COS( )	Cosine	in degrees		E.g.: R2=COS(R3)
TAN( )	Tangent	in degrees		E.g.: R4=TAN(R5)
SQRT( )	Square root			E.g.: R6=SQRT(R7)
ABS( )	Absolute value			E.g.: R8=ABS(R9)
TRUNC( )	Integer part			E.g.: R10=TRUNC(R11)
RET	End of subroutine	0.001 ... 99 999.999	Used instead of M2 to maintain continuous path mode	RET ;in separate block
S	Spindle speed or other meaning with G4, G96	0.001 ... 99 999.999	Spindle speed in rev/min if G96 is programmed, S is interpreted as constant cutting speed in m/min (turning), with G4, dwell time in spindle revolutions	S...
T	Tool number	1 ... 32 000 integer only, without sign	Tool change can be performed directly with T command or not until M6 is programmed. This can be set in machine data.	T...
X	Axis	± 0.001 ... 99 999.999	Positional data	X...
Z	Axis	± 0.001 ... 99 999.999	Positional data	Z...
AR	Angle of aperture for circular interpolation	0.00001 ... 359.99999	Given in degrees, a method of defining the circle with G2/G3	see G2; G3
CHF	Chamfer	0.001 ... 99 999.999	Inserts a chamfer of the specified length between two contour blocks	N10 X... Z.... CHF=... N11 X... Z...
CR	Radius for circular interpolation	0.010 ... 99 999.999 Negative sign - for circle selection: greater semi-circle	A method of defining a circle with G2/G3.	see G2; G3
GOTOB	GOTO instruction backwards	-	Jumps to the block defined by the label, the target of the jump is located in the direction of the beginning of the program.	E.g.: N20 GOTOB MARKE1
GOTOF	GOTO instruction forwards	-	Jumps to the block defined by the label, the target of the jump is located in the direction of the end of the program.	E.g.: N20 GOTOF MARKE2

Address	Meaning	Value assignment	Information	Programming
IF	Jump condition	-	If the jump condition is fulfilled the jump goes to the next instruction, Comparators: = = > >= <=	E.g.: N20 IF R1>5 GOTOB MARKE1
IX	Interpolation point for circular interpolation	± 0.001 ... 99 999.999	For the X axis, programmed for circular interpolation with G5	see G5
KZ	Interpolation point for circular interpolation	± 0.001 ... 99 999.999	For the Z axis, programmed for circular interpolation with G5	see G5
LCYC...	Machining cycle call	Specified values only	Machining cycles have to be called in a separate block, the transfer parameters to be used must be assigned values Transfer parameters:	
LCYC82	Drilling, spot-facing		R101: Retraction plane (absolute) R102: Safety clearance R103: Reference plane (absolute) R104: Final drilling depth R105: Dwell time at drilling depth	N10 R100=... R101=... ... N20 LCYC82 in separate block
LCYC83	Deep-hole drilling		R100: Number of the drilling axis =3 R101: Retraction plane (absolute) R102: Safety clearance R103: Reference plane (absolute) R104: Final drilling depth (absolute) R105: Dwell time at drilling depth R106: Dwell time start/stock removal R107: First drilling depth (absolute) R108: Amount of degression R109: Feedrate factor for drilling R110: Machining type: chipbreaking=0 stock removal=1 R111: Feedrate for first drilling depth	N10 R100=... R101=... .... N20 LCYC83 ;in separate block
LCYC840	Tapping with compensating chuck		R101: Retraction plane (absolute R102: Safety clearance R103: Reference plane (absolute) R104: Final drilling depth (absolute) R106: Thread lead value R126: Direction of rotation of spindle for tapping	N10 R100=... R101=... .... N20 LCYC840 ;in separate block

Address	Meaning	Value assignment	Information	Programming
LCYC85	Boring		R101: Retraction plane (absolute R102: Safety clearance R103: Reference plane (absolute) R104: Final drilling depth (absolute) R105: Dwell time at drilling depth R107: Feed for drilling R108: Feed on retract from drill hole	N10 R100=... R101=... .... N20 LCYC85 ;in separate block
LCYC93	Groove (drilling cycle)		R100: Starting point in facing axis R101: Starting point in longitudinal axis R105: Machining type (1...8) R106: Final machining allowance R107: Cutting edge width R108: Infeed depth R114: Groove width R116: Thread angle R117: Chamfer on groove edge R118: Chamfer at base of groove R119: Dwell time at base of groove	N10 R100=... R101=... .... N20 LCYC93 ;in separate block
LCYC94	Undercut (form E and F) (turning cycle)		R100: Starting point in facing axis R101: Starting point of contour in longitudinal axis R105: Form E=55, F=56 R107: Cutting edge position (1...4)	N10 R100=... R101=... .... N20 LCYC94 ;in separate block
LCYC95	Stock removal (turning cycle)		R105: Machining type (1...12) R106: Finishing allowance R108: Infeed depth R109: Infeed angle for roughing R110: Contour clearance for roughing R111: Feedrate for roughing R112: Feedrate for finishing	N10 R105=... R106= ... N20 LCYC95 in separate block
LCYC97	Thread cutting (turning cycle)		R100: Diameter of thread at starting point R101: Thread starting point in longitudinal axis R102: Thread diameter at end point R103: Thread end point in longitudinal axis R104: Thread lead value R105: Machining type (1 and 2) R106: Finishing allowance R109: Approach path R110: Run-out path R111: Thread depth R112: Starting point offset R113: Number of roughing cuts R114: Number of threads	N10 R100=... R101=... .... N20 LCYC97 ;in separate block
LIMS	Upper limit speed of spindle with G96	0.001 ... 99 999.999	Limits the spindle speed if function G96 is activated - constant cutting speed for turning	see G96
RND	Rounding	0.010 ... 99 999.999	Inserts a rounding with the radius value specified tangentially between two contour blocks	N10 X... Z.... RND=... N11 X... Z...

Address	Meaning	Value assignment	Information	Programming
SF	Thread commencement point with G33	0.001 ... 359.999	Specified in degrees, with G33 the thread commencement point is offset by the specified amount	see G33
SPOS	Spindle position	0.0000 ... 359.9999	Specified in degrees, the spindle stops at the specified position (spindle must be designed to do this)	SPOS=....
STOPRE	Preprocessing stop	-	Special function, the next block is not decoded until the block prior to STOPRE is completed	STOPRE ;in separate block
$P_TOOL	Active tool cutting edge	read-only	integer, DO to D9	IF $P_TOOL==7 GOTOF ...
$P_TOOL NO	Active tool number	read-only	integer, TO - T32000	IF $P_TOOLNO== 46 GOTOF ...
$P_TOOLP	Tool number last programmed	read-only	integer, TO - T32000	IF $P_TOOLNP== 11 GOTOF ...

8.2 Position data

8.2.1 Absolute/incremental dimensions: G90, G91

Functionality

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

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

Programming G90

;Absolute dimensioning
G91 ;Incremental dimensioning

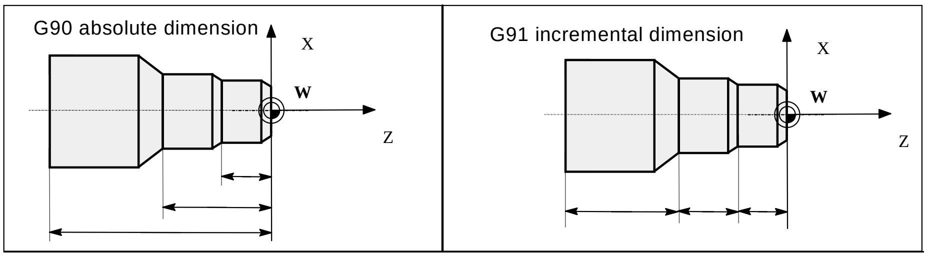


Fig.8-3 Different dimensioning in the part drawing

Absolute dimension G90

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

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

Incremental dimension G91

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

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

Programming example for G90 and G91

```
N10 G90 X20 Z90      ;Absolute dimensioning
N20 X75 Z-32         ;Absolute dimensioning still active
...
N180 G91 X40 Z20     ;Switchover to incremental dimensioning
N190 X-12 Z17        ;Incremental dimensioning still active
```

8.2.2 Metric/inch dimensions: G71, G70

Functionality

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

Programming

```
G70                 ;Inch dimension
G71                 ;Metric dimension
```

Programming example

```
N10 G70 X10 Z30     ;Inch dimension system
N20 X40 Z50         ;G70 still active
...
N80 G71 X19 Z17.3   ;Metric dimension system from here
...
```

Information

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

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

G70 and G71 affect all geometric data that refer directly to the workpiece:

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

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

8.2.3 Radius/diameter dimensions: G22, G23

Functionality

When parts are machined on turning machines, it is normal practice to program the position data for the X axis (facing axis) as a diameter dimension.
The specified value is interpreted as a diameter for this axis only by the control.
It is possible to switch over to radius dimension in the program if necessary.

Programming

G22	;Radius dimension
G23	;Diameter dimension

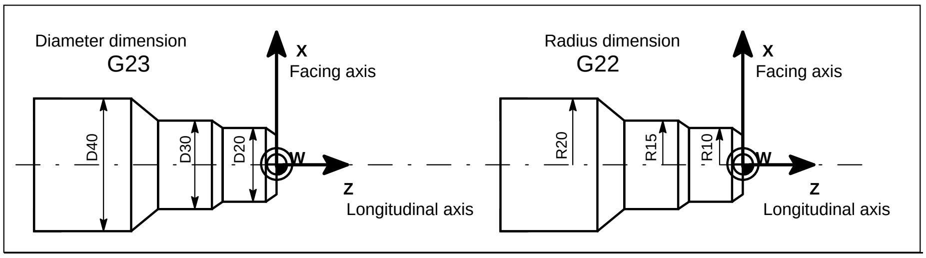


Fig.8-4 Diameter and radius dimensions for facing axis

Information

When G22 or G23 is active, the specified end point for the X axis is interpreted as a radius or diameter dimension.

The actual value is displayed correspondingly in the workpiece coordinate system. A programmable offset with G158 X... is always interpreted as a radius dimension. See the following section for a description of this function.

Programming example

```
N10 G23 X44 Z30      ;Diameter for X axis
N20 X48 Z25          ;G23 still active
N30 Z10
...
N110 G22 X22 Z30     ;Changeover to radius dimension for X axis from here
N120 X24 Z25
N130 Z10
...
```

8.2.4 Programmable zero offset: G158

Functionality

Use the programmable zero offset for frequently repeated shapes/arrangements in different positions on a workpiece or when you simply wish to choose a new reference point for the dimension data. The programmable offset produces the current workpiece coordinate system. The newly programmed dimension data then refer to this system. The offset can be applied in all axes.

A separate block is always required for the G158 instruction.

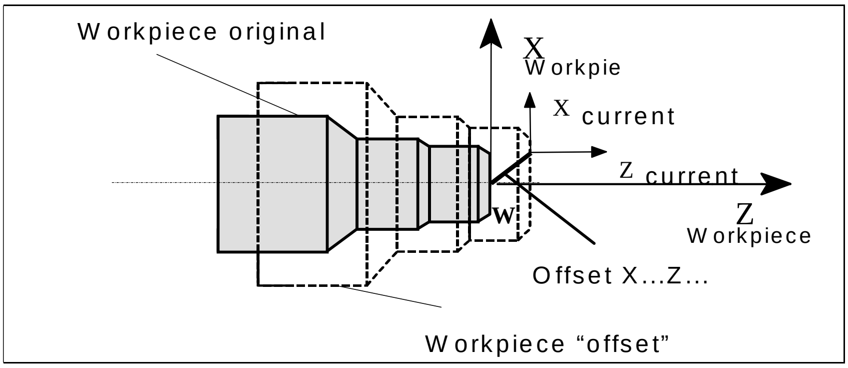


Fig.8-5 Example of programmable offset

Offset G158

A zero offset can be programmed for all axes with instruction G158. A newly entered G158 instruction replaces any previous programmable offset instruction.

Delete offset

If the instruction G158 without axes is inserted in a block , then any active programmable offset will be deleted.

Programming Example

```
N10 ...
N20 G158 X3 Z5   ;Programmable offset
N30 L10          ;Subroutine call, contains the geometry to be offset
...
N70 G158         ;Offset deleted
...
```

Subroutine call - see Section 8.10 “Subroutine System”

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

Functionality

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

See Section 3.2 “Enter/Modify Zero Offset” for operating sequence.

Programming

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

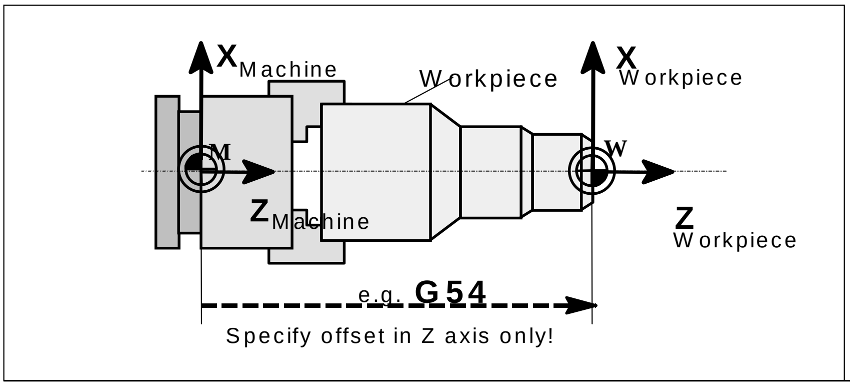


Fig.8-6 Settable zero offset

Programming Example

N10 G54 ...	;Call first settable zero offset
N20 X... Z...	;Machine workpiece
...	
N90 G500 G0 X...	;Deactivate settable zero offset

8.3 Axis movements

8.3.1 Linear interpolation at rapid traverse: G0

Functionality

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

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

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

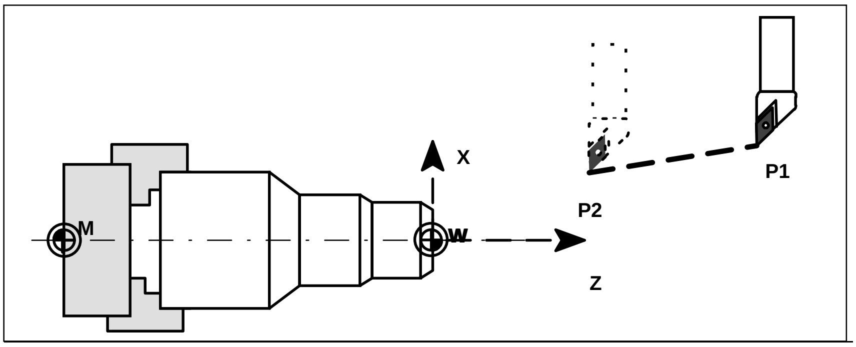


Fig.8-7 Linear interpolation with rapid traverse from point P1 to P2

Programming example

N10 G0 X100 Z65

Information

A further group of G functions is provided for programming the approach to the poisition (see Section 8.3.9 “Exact Stop/Continuous Path Control: G60, G64”). G60 (exact stop) is linked to another group which allows various accuracy settings to be selected in a window. There is also a non-modal instruction, i.e. G9, for the exact stop function.

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

8.3.2 Linear interpolation at feedrate: G1

Functionality

The tool moves from the start point to the end point along a straight path. The path speed is defined by the programmed F word.
All axes can be traversed simultaneously.
G1 remains effective until it is canceled by another instruction from the same G group (G0, G2, G3, ...).

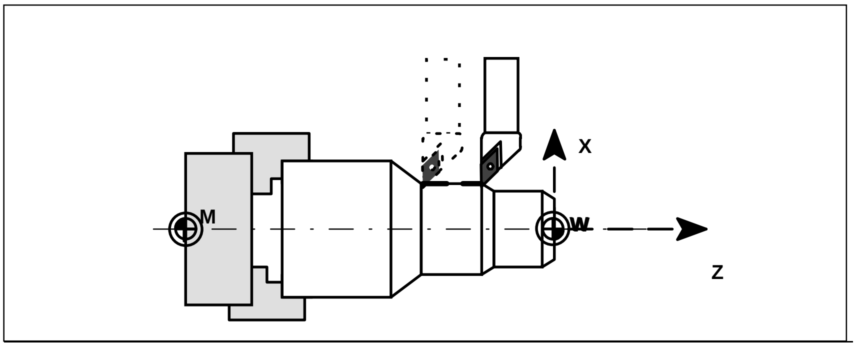


Fig.8-8 Linear interpolation with G1

Programming example

```
N05 G54 G0 G90 X40 Z200 S500 M3 ;tool is moving at rapid traverse, spindle
                                  speed = 500 rpm, CW rotation
N10 G1 Z120 F0.15               ;Linear interpolation with feed 0.15
                                  mm/rev
N15 X45   Z105
N20 Z80
N25 G0 X100                     ;Traverse clear at rapid traverse
N30 M2                          ;End of program
```

8.3.3 Circular interpolation: G2, G3

Functionality

The tool moves from the start point to the end point on a circular path. The direction is determined by the G function:

G2 - in clockwise direction

G3 - in counterclockwise direction

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

— Center point and end point

— Circle radius and end point

— Center point and aperture angle

Aperture angle and end point

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

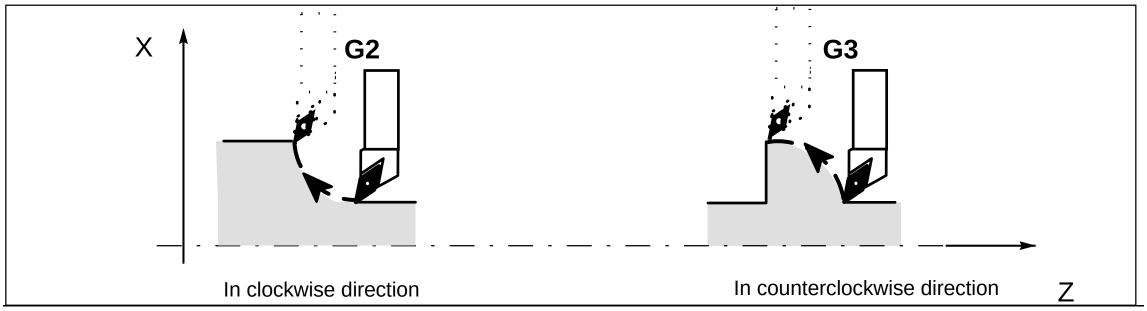


Fig.8-9 Definition of direction of rotation around circle with G2/G3

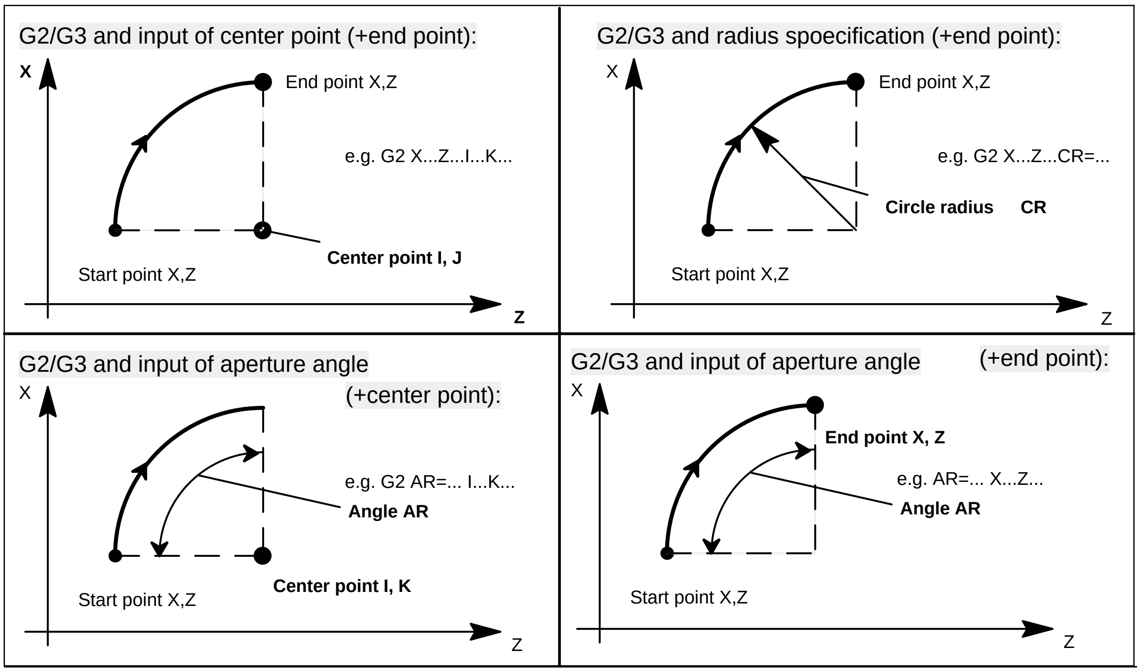


Fig.8-10 Circle programming options

Programming example

Center point and end point specification:

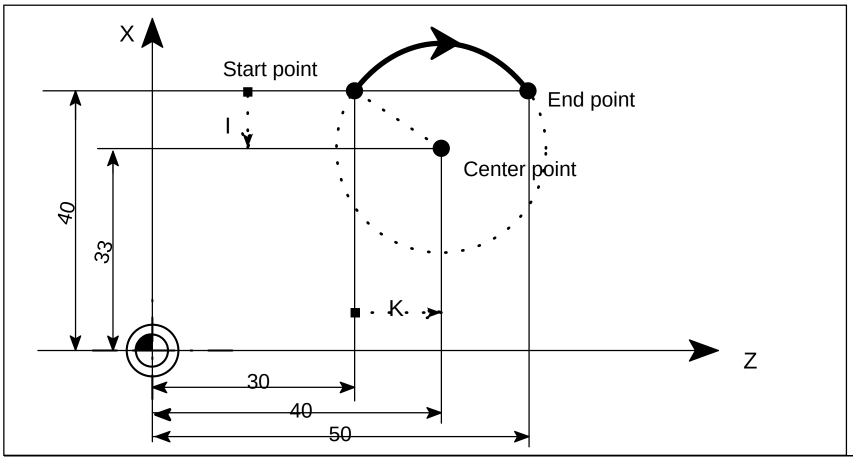


Fig.8-11 Example of center and end point specification

N5 G90 Z30 X40	;Circle start point for N10
N10 G2 Z50 X40 K10 I-7	;End point and center point

Programming example

End point and radius specification:

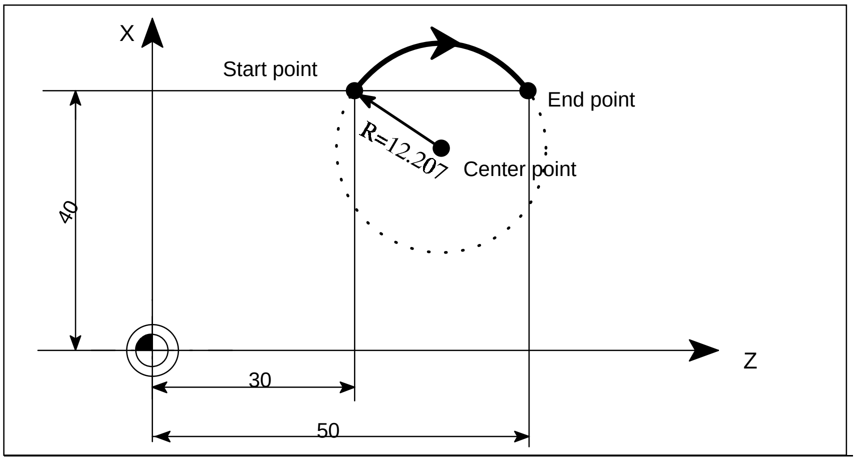


Fig.8-12 Example of end point and radius input

N5 G90 Z30 X40	;Circle start point for N10
N10 G2 Z50 X40 CR=12.207	;End point and radius

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

Programming example

End point and aperture angle:

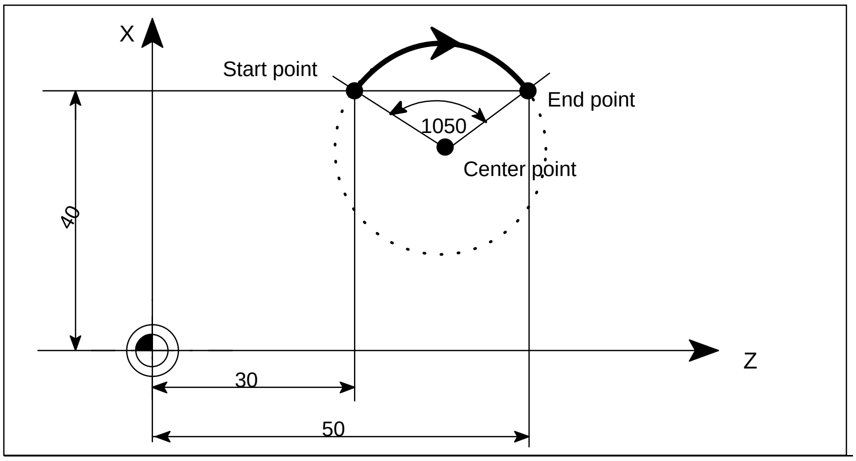


Fig.8-13 Example of end point and aperture angle specification

N5 G90 Z30 X40	;Circle start point for N10
N10 G2 Z50 X40 AR=105	;End point and aperture angle

Programming example

Center point and aperture angle:

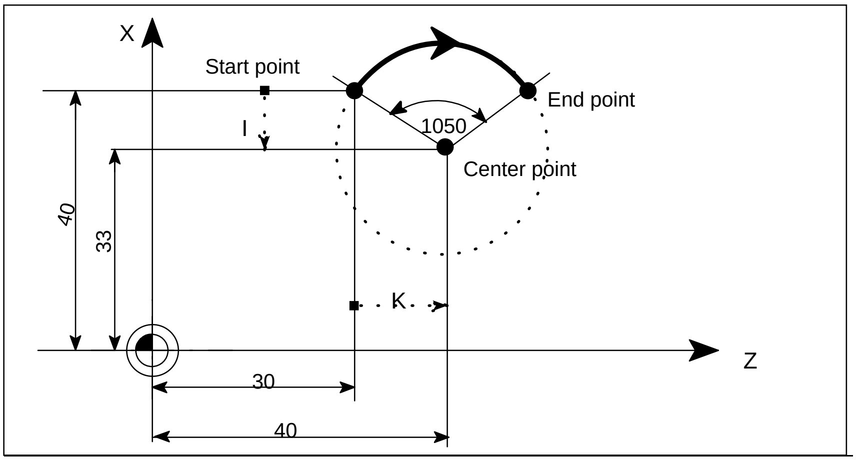


Fig.8-14 Example of center point and aperture angle specification

N5 G90 Z30 X40	;Circle start point for N10
N10 G2 K10 I-7 AR=105	;Center point and aperture angle

Input tolerances for circle

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

The tolerance value can be set via the machine data.

8.3.4 Circular interpolation via intermediate point: G5

Functionality

If you know three contour points around the circle instead of center point or radius or aperture angle, you should preferably use the G5 function.

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

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

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

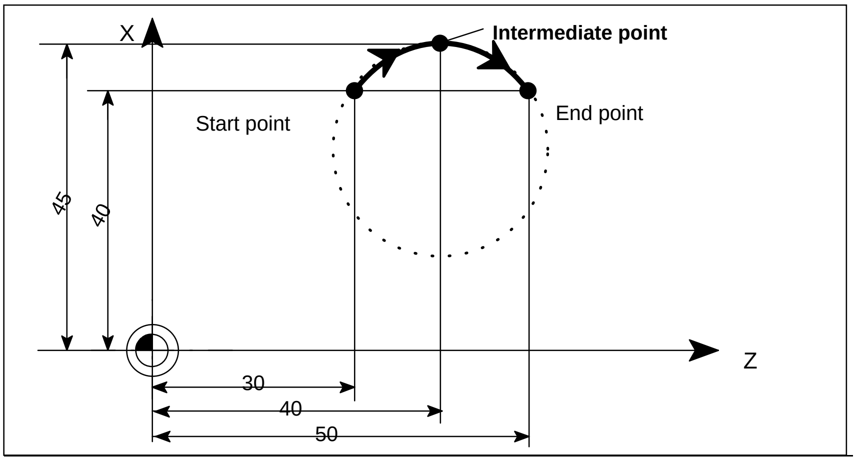


Fig.8-15 Circle with end and intermediate point specification with G90 active

Programming example

N5 G90 Z30 X40 ;Circle start point for N10
N10 G5 Z50 X40 KZ=40 IX=45 ;End and intermediate points

(XI must be programmed as a radius dimension)

8.3.5 Thread cutting with constant lead: G33

Functionality

Function G33 can be used to cut the following types of threads with constant lead:

- Thread on cylindrical bodies
- Thread on tapered bodies
- External/internal threads
- Single-start/multiple-start threads
- Multi-block threads (thread “chaining”)

This function requires a spindle with position measuring system.

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

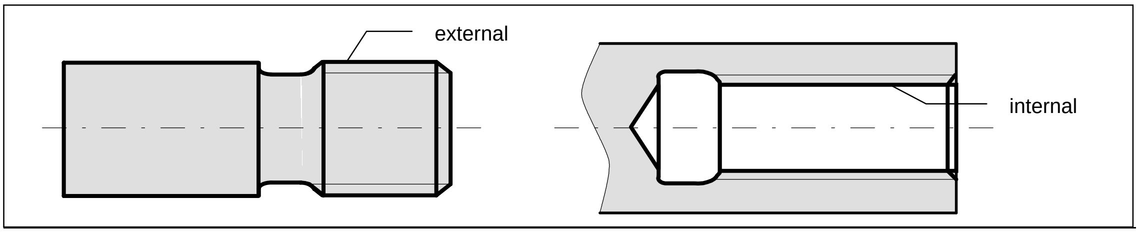


Fig.8-16 Example of external /internal thread on cylindrical body

RH or LH threads

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

Note: The approach and run-out paths must be taken into account with respect to the thread length.

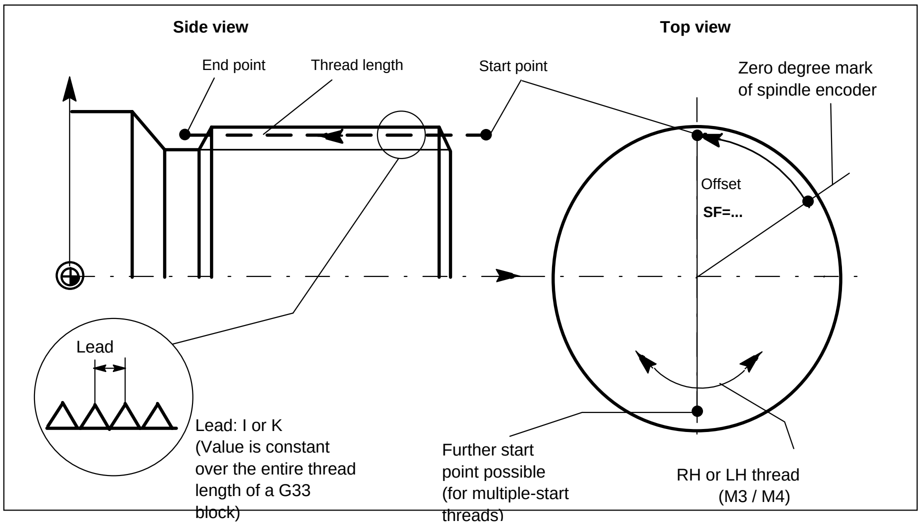


Fig.8-17 Programmable quantities for thread cutting with G33

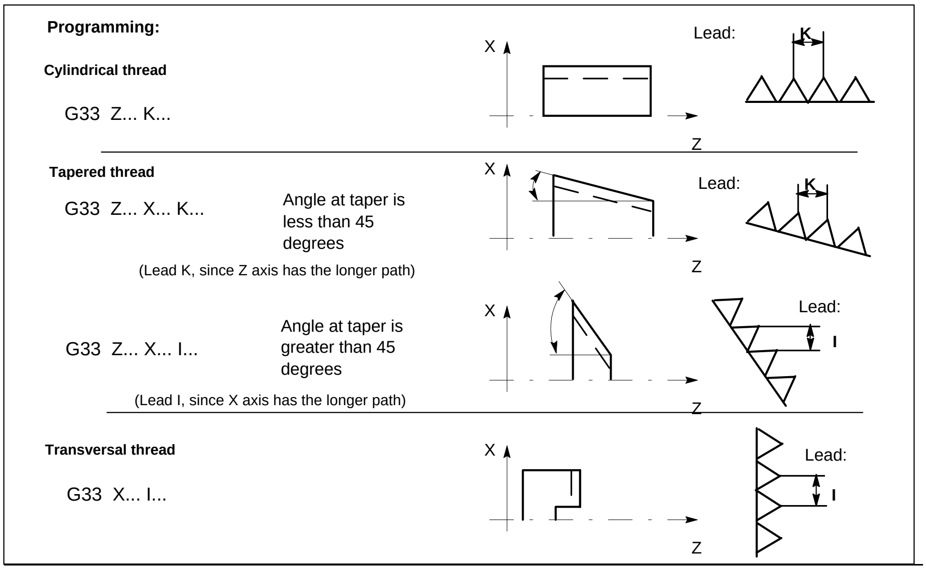


Fig.8-18 Lead assignment on the example of Z/X axis

In the case of tapered threads (2 axes must be specified), the lead address I or K of the axis with the longer path (greater thread length) must be used. A second lead is not specified.

Start-point offset SF=

A start-point offset of the spindle is required for machining multiple-start threads or threads in offset cuts. The start-point offset is programmed under address SF in the thread block with G33 (absolute position).

If a start point is not included in the block, the value from the setting data is activated.

Note: Any value programmed for SF= is always entered in the setting data as well.

Programming example

Cylindrical thread, two-start, start-point offset 180 degrees, thread length (including approach and run-out) 100 mm, thread lead 4 mm/rev.,
RH thread, cylinder premachined:

```
N10 G54 G0 G90    X50 Z0 S500 M3  ;Approach start point, CW spindle
                                   rotation
N20 G33 Z-100 K4 SF=0             ;Lead:4 mm/rev.
N30 G0 X54
N40 Z0
N50 X50
N60 G33 Z-100 K4 SF=180           ;2nd start, 180 degrees offset
N70 G0 X54 ...
```

Multi-block thread

If several thread blocks are programmed in succession (multi-block thread), it makes only sense to program a start-point offset in the 1st thread block since this is the only block in which the function is effective.

Multi-block threads are automatically linked by G64 (continuous path mode, see Section 8.3.9 ”Exact Stop/Continuous Path Control: G60, G64”).

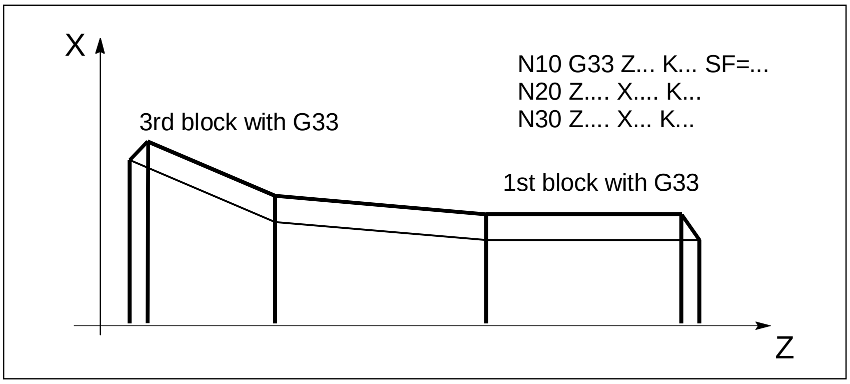


Fig.8-19 Example of multi-block thread (thread “chaining”)

Axis velocities

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

Information

Important

— The setting of the spindle speed override switch (override spindle) should not be changed for thread machining operations.

— The feed override switch has no function in this block.

8.3.6 Fixed-point approach: G75

Functionality

G75 can be used to approach to a fixed point on the machine, such as the tool change point. The position is fixed for all axes in the machine data. No offset is applied.
The speed of each axis is its own rapid traverse setting.
G75 required a separate block and is non-modal.
The G command from the Interpolation Type group (G0, G1,G2, ...) which was active prior to the block with G75 is activated after the block with G75.

Programming example

N10 G75 X0 Z0
Note: The programmed numerical values for X, Z are ignored.

8.3.7 Reference point approach: G74

Functionality

G74 is used to execute the reference-point approach in the NC program. Direction and speed of each axis are stored in machine data.
G74 requires a separate block and is non-modal. The G command for the Interpolation Type group (G0, G1,G2, ...) active prior to the block with G74 is activated again after the block with G74.

Programming example

N10 G74 X0 Z0
Note: The programmed numerical values for X, Z are ignored.

8.3.8 Feedrate F

Functionality

The feedrate F is the path speed and represents the absolute value of the geomnetric total of the speed components of all axes involved.

The axis speeds are determined by the axis path distance in relation to the total path distance.

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

Programming

F...

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

Unit for F–G94, G95

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

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

Programming example

```
N10 G94 F310      ;Feedarte in mm/min
...
N110 S200 M3      ;Spindle operation
N120 G95 F15.5    ;Feedrate in mm/rev
```

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

Information

The group with G94 and G95 has additional functions G96 and G97 for constant cutting rate for turning machines. These functions also influence the S word (see Section 8.5.1 “Constant Cutting Rate“).

8.3.9 Exact stop / continuous path mode: G9, G60, G64

Functionality

These G functions enable you to set the traversing behavior at block limits and to control program advance to the next block, thus allowing you to adapt your program optimally to various requirements. For example, you want to position quickly with the axes or process path contours over several blocks.

Programming

G60	;Exact stop - (modal)
G64	;Continuous path mode
G9	;Exact stop - (non-modal)
G601	;Exact stop window fine
G602	;Exact stop window coarse

Exact stop G60, G9

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

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

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

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

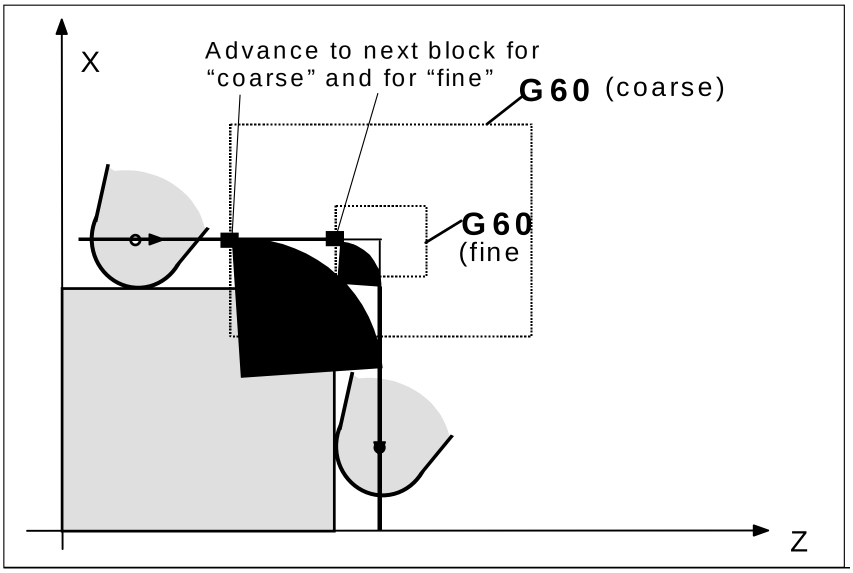


Fig.8-20 Coarse or fine exact stop window, effective with G60/G9, zoomed view of window

Programming example

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

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

Continuous path mode G64

The purpose of continuous path mode is to prevent braking of the axes at block limits to make the transition to the next block at the most constant possible speed (in the case of tangential transitions). The function operates with lookahead speed control to the next block. In the case of non-tangential path transitions (corners), the speed is reduced to such an extent in some cases that none of the axes is capable of making a speed step change that is higher than the maximum acceleration rate. In such cases, speed-dependent rounding at corners occurs.

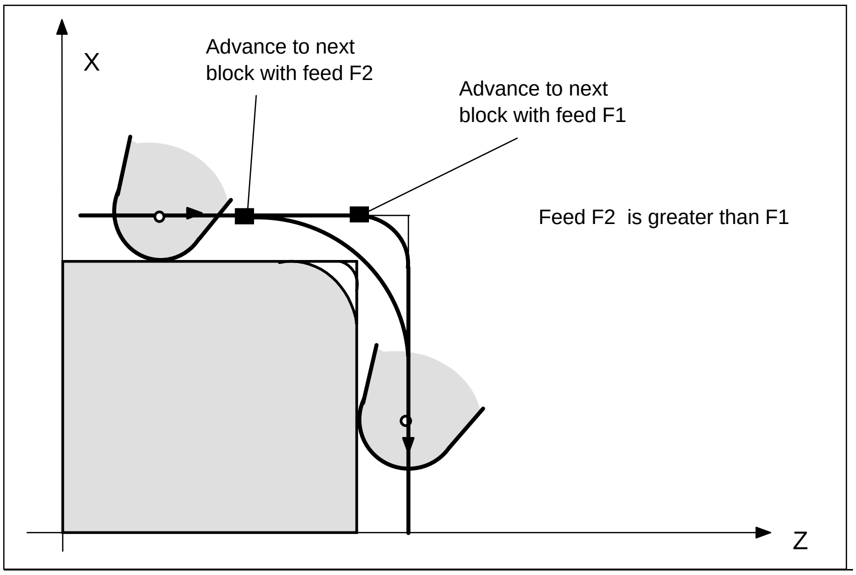


Fig.8-21 Rounding at contour corners with G64

Programming example

```
N10 G64 G1 Z... F...   ;Continuous path mode
N20   X..              ;Continuous path control mode still active
...
N180 G60 ...           ;Switching to exact stop
```

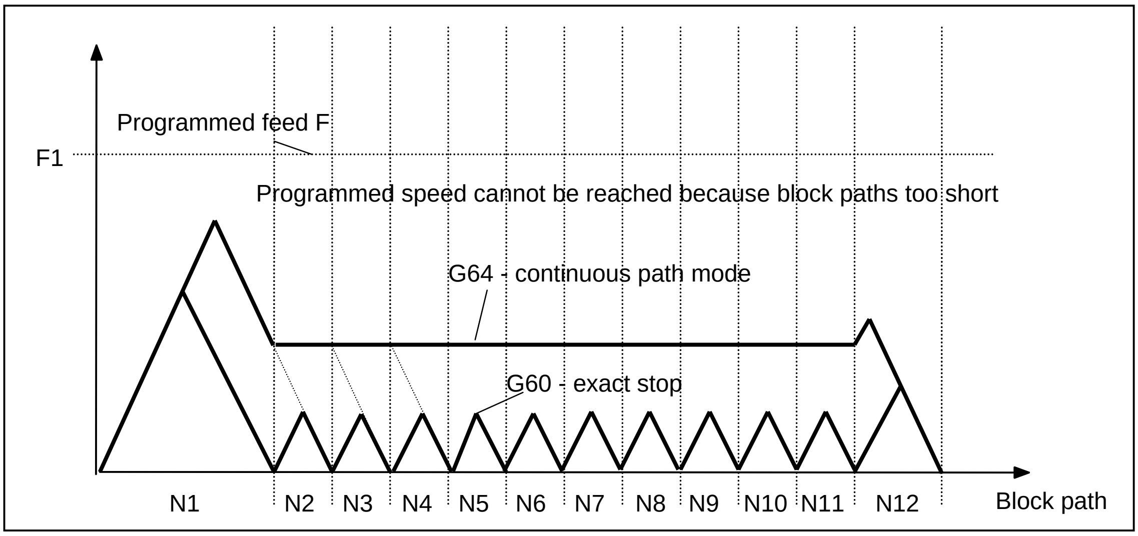


Fig.8-22 Comparison between speed responses with G60 and G64 with short block paths

8.3.10 Dwell time: G4

Functionality

You can interrupt machining between two NC blocks for a defined time period by inserting a separate block with G4, e.g. for relief cutting operations.
The words with F... or S... are used for time specifcations only in this block. Any previously programmed feed F or spindle speed S remain unaffected.

Programming

G4 F...	;Dwell time in seconds
G4 S...	;Dwell time in spindle revolutions

Programming example

N5 G1 F200 Z-50 S300 M3	;Feed F, spindle speed S
N10 G4 F2.5	;Dwell time 2.5 s
N20 Z70	
N30 G4 S30	;Dwell for 30 spindle revolutions, corresponds to t = 0.1 when S = 300 rev/min and 100% speed override
N40 X...	;Feed and spindle speed values remain effective

Note

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

8.4 Spindle movements

8.4.1 Spindle speed S, directions of rotation

Functionality

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

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

Information

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

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

Note:
Other settings can be configured via machine data.

Programming example

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

8.4.2 Spindle speed limitation: G25, G26

Functionality

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

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

Programming

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

Information

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

The special function G96 (constant cutting rate) can be used to program an additional upper limit on turning machines.

Programming example

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

8.4.3 Spindle positioning: SPOS

Functionality

Precondition: The spindle must be technically designed for operation under closed-loop position control.

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

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

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

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

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

Programming

```
SPOS=...                          ;Absolute position: 0 ... <360 Grad
```

Programming example

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

8.5 Special turning functions

8.5.1 Constant cutting rate: G96, G97

Functionality

Precondition: The machine must have a controlled spindle.

When the G96 function is active, the spindle speed is adjusted to the diameter of the workpiece currently being machined (facing axis) such that a programmed cutting rate S remains constant at the tool edge (spindle speed times diameter = constant).

The S word is interpreted as the cutting rate from the block with G96 onwards. G96 is active as a modal command until it is cancelled by another G function in the same group (G94, G95, G97).

Programming

```
G96 S... LIMS=... F...    ;Constant cutting rate ON
G97                       ;Constant cutting rate OFF
```

STL	Explanation
S	Cutting rate, unit m/min
LIMS=	Upper limit speed of spindle, effective only with G96
F	Feed in mm/rev unit of measurement - as for G95

Note:

In this case, feed F is always interpreted in the unit of measurement mm/revolution.

If G94 was active instead of G95 beforehand, then a suitable F word must be inserted again in the program!

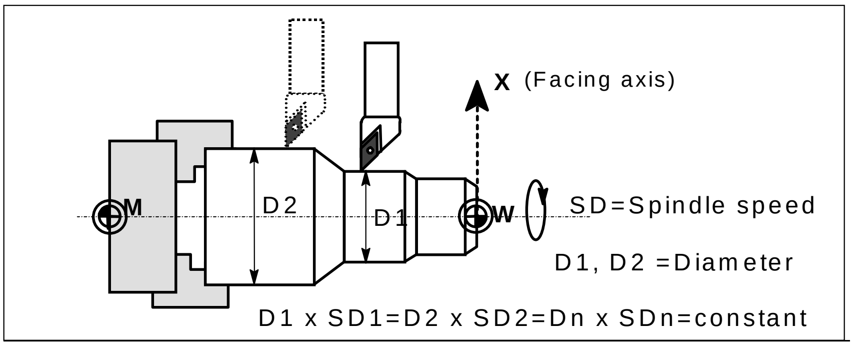


Fig.8-23 Constant cutting rate G96

Traversing at rapid traverse

No speed changes take place during rapid traversal with G0.
Exception: If the contour is approached in rapid traverse mode and the next block contains an interpolation type G1 or G2, G3, G5 (contour block), then the speed is adjusted to the value for the contour block while the approach block with G0 is being processed.

Upper limit speed LIMS=

The spindle speed may rise sharply when large diameters are machined down to small diameters. For such applications, it is advisable to specify the upper spindle speed limitation by means of LIMS=... . LIMS is effective only in conjunction with G96.

When LIMS=... is programmed, the value entered in the setting data is overwritten.

The upper speed limit programmed with G26 or via machine data cannot be overwritten by the LIMS= function.

Deactivate constant cutting rate: G97

The “Constant cutting rate” function is deactivated with G97. If G97 is active, a subsequently programmed S word is interpreted again as the spindle speed in revolutions per minute.

If no further S word is inserted in the program, then the spindle continues to rotate at the speed that was recorded when the G96 function was last active.

Programming example

```
N10 ...   M3                ;Direction of rotation of spindle
N20 G96 S120 LIMS=2500      ;Activate constant cutting rate, 120 m/min,
                             limit speed 2,500 rev/min
N30 G0 X150                 ;No speed change because block N31 includes
                             G0
N31 X50 Z...                ;No speed change because block N32 includes
                             G0
N32 X40                     ;Approach contour, new speed is automatically
                             set to value required for start of block N40
N40 G1 F0.2 X32 Z...        ;Feed 0.2 mm/rev
...
N180 G97 X... Z...          ;Deactivate constant cutting rate
N190 S...                   ;New spindle speed, rev/min
```

Information

The G96 function can also be deactivated by G94 or G95 (same G group). In this case, the last programmed spindle speed S applies for the remainder of the machining operation provided no new S word is programmed.

8.5.2 Rounding, chamfer

Functionality

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

Programming

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

Chamfer CHF=

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

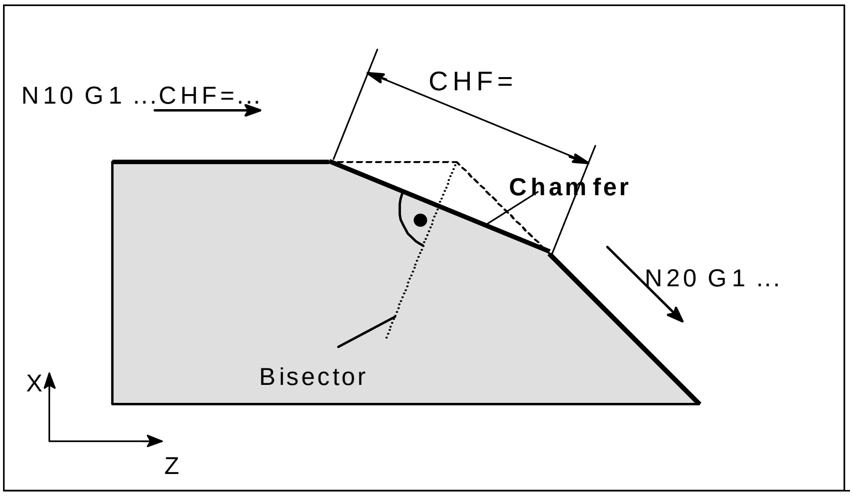


Fig.8-24 Insertion of a chamfer between two linear contours (example)

Programming example for chamfer

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

Rounding RND=

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

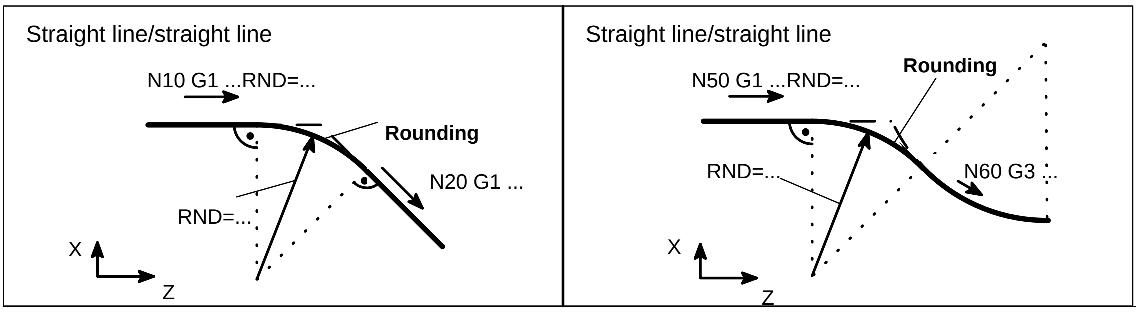


Fig.8-25 Examples of rounding insertion

Programming example for rounding

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

Information

Note:

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

No chamfer/rounding is inserted if more than one of the subsequently programmed blocks does not contain any information about traversal of the axes.

8.6 Tool and tool offset

8.6.1 General notes

Functionality

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

The tool data are entered separately in a special data area. You merely call the tool you need together with its offset data in the program. On the basis of this data, the control executes the necessary path compensations in order to produce the workpiece you have defined.

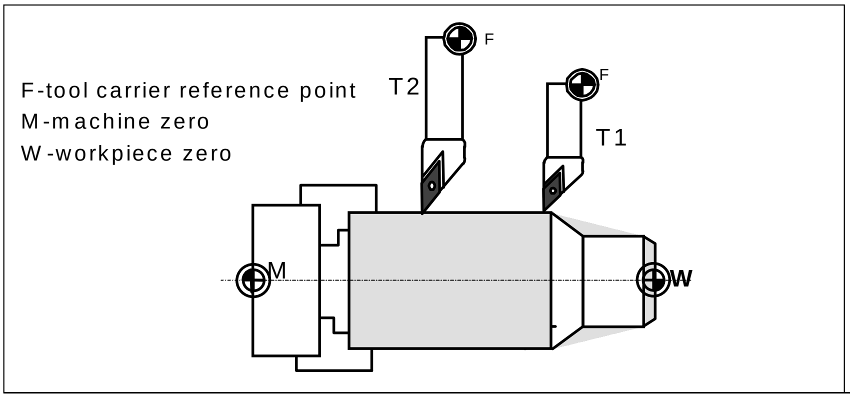


Fig.8-26 Machining of a workpiece with various tool dimensions

8.6.2 Tool T

Functionality

You select a tool by programming the T word. A machine data defines whether the T word represents a tool change or merely a preselection.

- Tool change (tool call) is implemented directly by T word (e.g. normal practice for tool revolver on turning machines) or
- the tool is changed through additional instruction M6 after preselection by T word (see also Section "Miscellaneous Functions M").

Please note:
If a certain tool has been activated, this will remain stored as the active tool even across the program end and after POWER ON of the control system.

If you change a tool manually, then enter the change into the control system also manually to make sure that the control system detects the right tool. For example, you can start a block with a new T word in the MDA mode.

Programming

T... ;Tool number: 1 ... 32 000

Note

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

Programming example

Tool change without M6:

```
N10 T1       ;Tool 1
...
N70 T588     ;Tool 588
```

8.6.3 Tool offset number D

Functionality

You can assign between 1 and 9 data fields with various tool offset blocks (for several tool edges) to each specific tool. If a special edge is required, it can be programmed by means of D plus a corresponding number.

D1 is the automatic default if no D word is programmed.
When D0 is programmed, then the offsets for the tool are not active.

Note

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

Programming

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

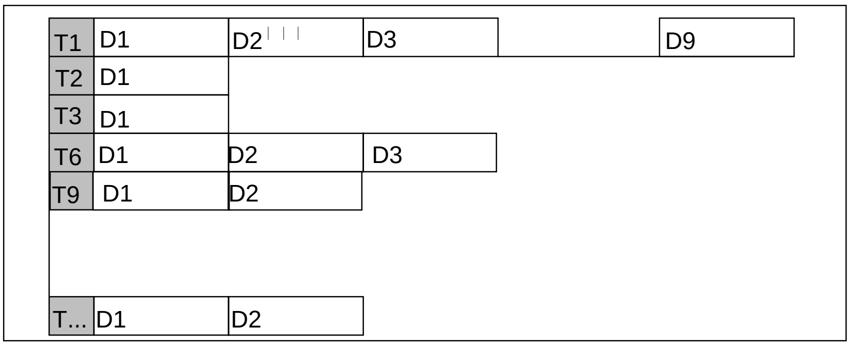


Fig.8-27 Example assignment of tool offset numbers to tool

Information

Tool length compensations take immediate effect when the tool is active. The values of D1 are applied if no D number has been programmed. The tool length is compensated when the first programmed traversal of the relevant length compensation axis is executed.

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

Programming example

```
Tool change:
N10 T1             ;Tool 1 is activated with associated D1
N11 G0 X... Z...   ;The length compensation is superimposed here
N50 T4 D2          ;Change to tool 4, D2 of T4 becomes active
...
N70 G0 Z... D1     ;D1 for tool 4 is active, only edge changed
```

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 component data to produce a resultant quantity (e.g. total length 1, total radius). The total dimension calculated in each case takes effect when the offset memory is activated.

The method used to compute these values in the axes depends on the tool type and commands G17, G18 (see diagrams below).

- Tool type

The tool type determines which geometry data are necessary and how they are computed (drills or turning tools). The hundreds place is the only distinguishing digit:

— Type 2xy: Drills

— Type 5xy: Turning tools

- Tool point direction

You must also specify the tool point direction for tool type 5sy (turning tools).

Tool parameters

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

Tool type:	DP1	
Edge length:	DP2	
	Geometry	**Wear**
Length 1:	DP3	DP12
Length 2:	DP4	DP13
Radius:	DP6	DP15

The following diagrams show which tool parameters are needed for which tool type.

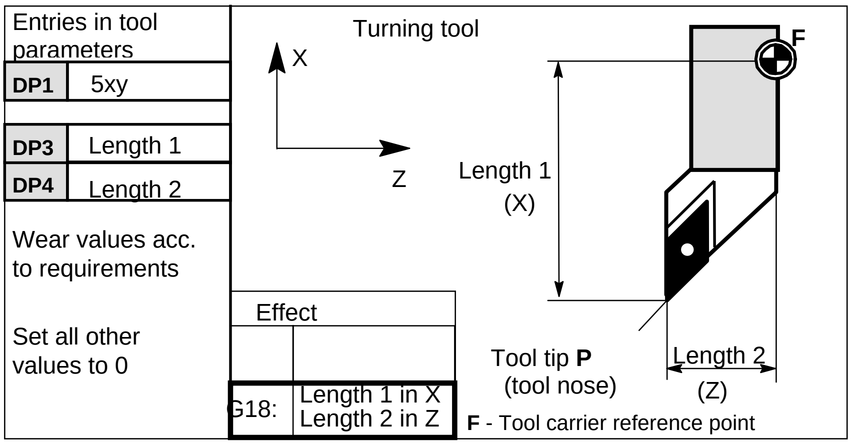


Fig.8-28 Length compensation values required for turning tools

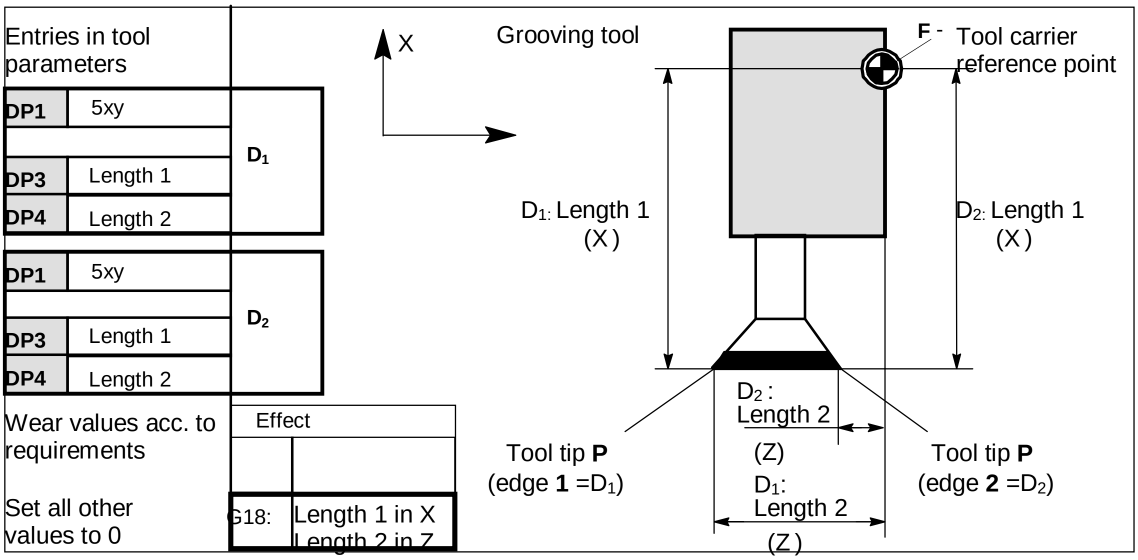
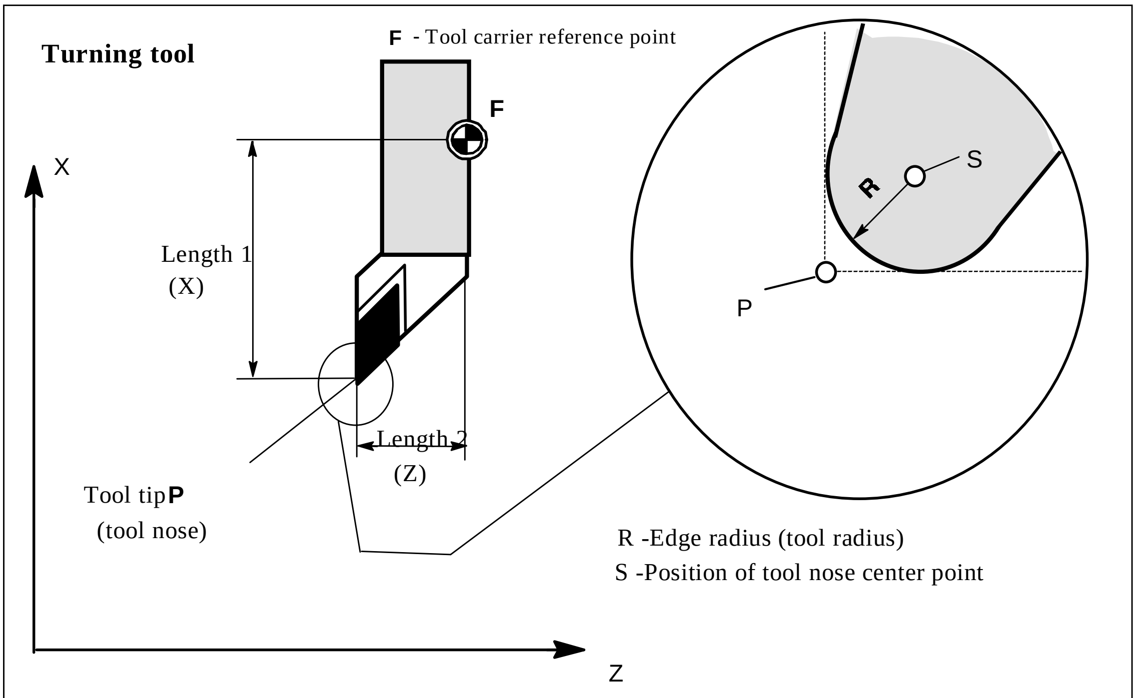
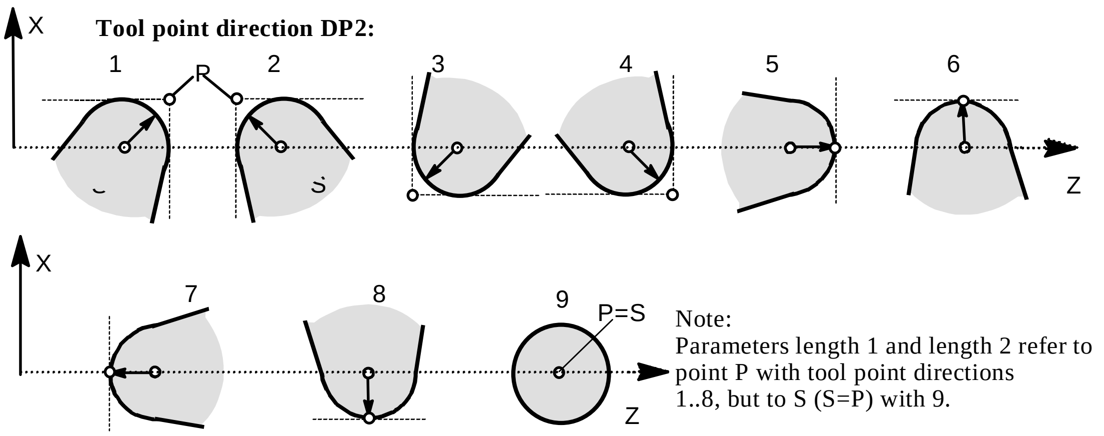


Fig.8-29 Turning tool with length compensation for two edges



The tool parameter DP2 specifies the tool point direction. Direction value 1 to 9 can be programmed:



Entries in tool parameters			Effect	
DP1	5xx	Wear values acc. to requirements		
DP2	1...9		G18:	Length 1 in X Length 2 in Z
DP3	Length1	Set all other values to 0		
DP4	Length 2			
DP6	Radius			

Fig.8-30 Offset data required for turning tools with tool radius compensation

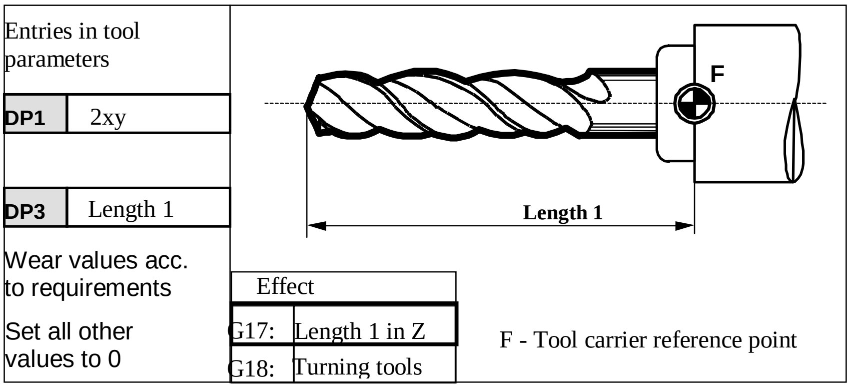


Fig.8-31 Offset data required for drills

Centre hole

To drill a center hole, switch over to G17. The length compensation then acts on the drill in the Z axis. After drilling, switch back to G18 for the normal turning tool offset.

Example:

N10 T... ;Drill, =tool type 200
N20 G17 G1 F... Z... ;Length compensation acts in Z axis
N30 Z...
N40 G18 .... ;Drilling completed

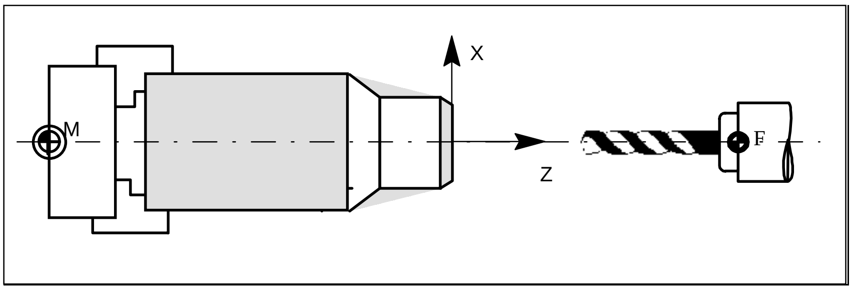


Fig.8-32 Drilling a center hole

8.6.4 Selection of tool radius compensation: G41, G42

Functionality

A tool with a corresponding D number must be active. The tool radius compensation (tool nose radius compensation) is activated by G41/G42. The control then automatically calculates the necessary tool paths equidistant from the programmed contour for the current tool radius.

G18 must be active.

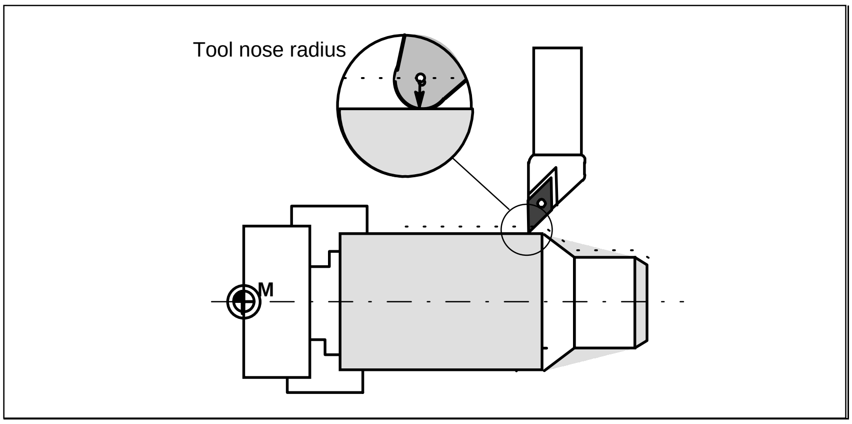


Fig.8-33 Tool (nose) radius compensation

Programming

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

Note: You may only select the function for linear interpolation (G0, G1). Program both axes. If you only specify one axis, then the last programmed value is automatically set for the second axis.

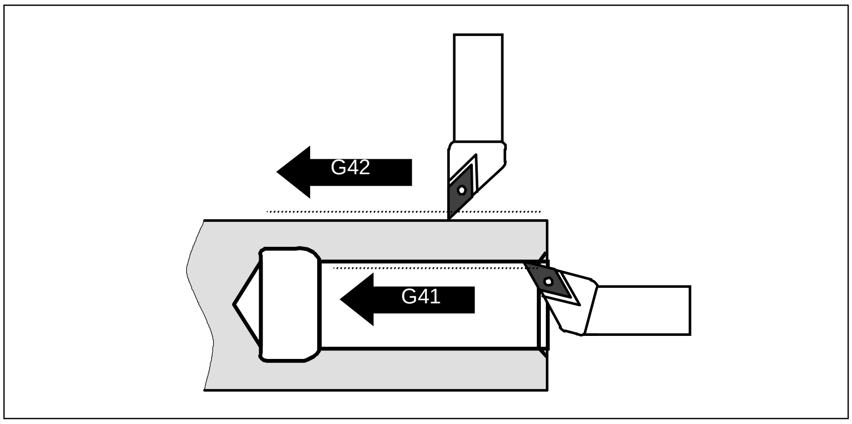


Fig. 8-34 Compensation to right/left of contour

Begin compensation

The tool approaches the contour on a linear path and positions itself perpendicular to the path tangent at the start of the contour. Select the start point such that the tool can traverse safely with no risk of collision.

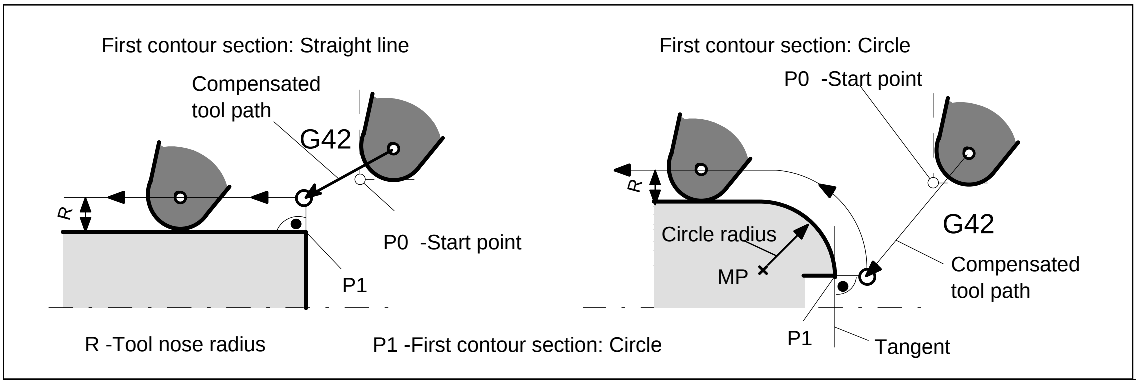


Fig.8-35 Beginning of tool radius compensation –example shows G42, tool point direction =3

Information

The block containing G41/G42 is generally followed by the first block with the workpiece contour. The contour definition may, however, be interrupted by an intermediate block that does not contain any contour information, e.g. by a block with just an M command.

Programming example

```
N10 T...   F...
N15 X... Z...          ;P0 start point
N20 G1   G42 X... Z... ;Select compensation to the right of the contour, P1
N30 X...   Z...        ;Initial contour section, circle or straight line
```

8.6.5 Behavior at corners: G450, G451

Functionality

Functions G450 and G451 are provided to allow you to set the response in the case of discontinuous transition from one contour element to another (behavior at corners) when G41/G42 is active.

The control itself detects inside and outside corners. The point at which the equidistant paths intersect is always approached in the case of inside corners.

Programming

G450 ;Transition circle
G451 ;Intersection

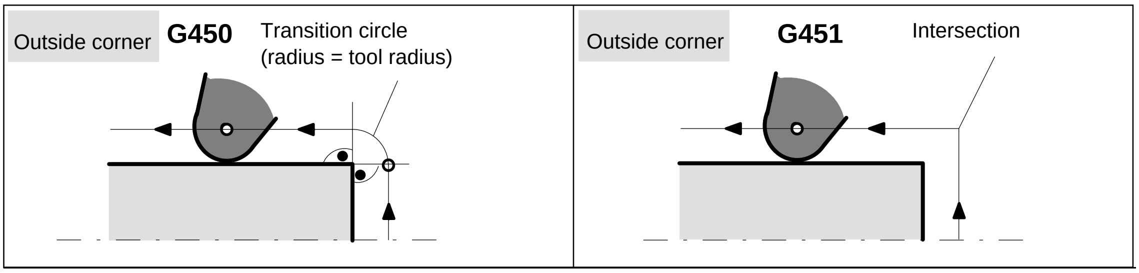


Fig.8-36 Behavior at an outside corner

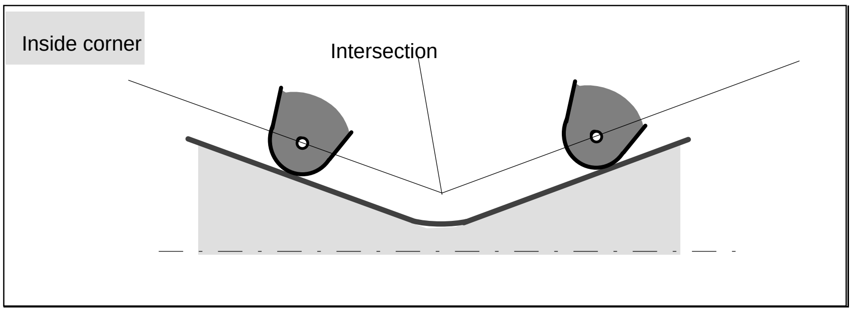


Fig 8-37 Behavior at an inside corner

Transition circle G450

The tool center point traverses round the workpiece outer corner along an arc with the same radius as the tool radius.

In processing terms, the transition circle belongs to the next block that contains traversing movements, e.g. relating to feed value.

Intersection G451

With function G451 (intersection of equidistant paths), the tool approaches the point at which the center point paths (circle or straight line) intersect.

8.6.6 Tool radius compensation OFF: G40

Functionality

Function G40 is used to canceled compensation mode G41/G42. This G function is also preset for program start.

The tool ends the block before G40 in normal position (i.e. compensation vector perpendicular to tangent at end point), independently of retraction angle. Always select the end point of the G40 block such that the tool can traverse safely with no risk of collision.

Programming

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

Note: Tool radius compensation can be canceled only in linear interpolation mode (G0, G1).

Program both axes. If you only specify one axis, then the last programmed value is automatically set for the second axis.

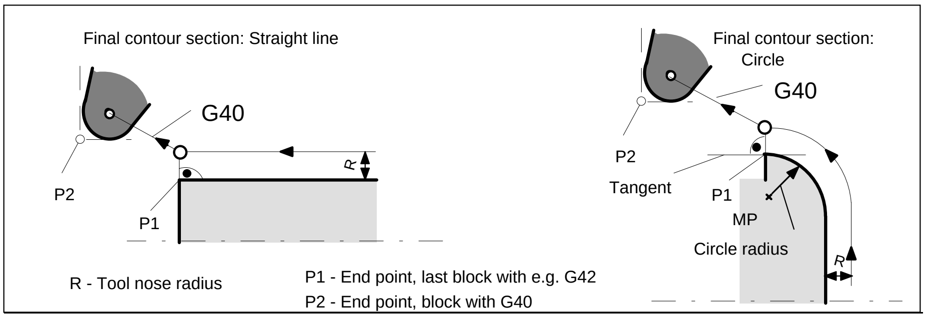


Fig.3-38 Cancelation of tool radius compensation using G40, example shows G42, tool point direction=3

Programming example

```
...
N100 X... Z...          ;Last block on contour, circle or straight line, P1
N110 G40 G1 X... Z...   ;Deactivate tool radius compensation, P2
```

8.6.7 Special cases of tool radius compensation

Change in compensation direction

The compensation direction G41 <-> can be changed without inserting a G40 instruction in-between.

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

Repetition of G41, G41 or G42, G42

The same contour can be programmed again without inserting a G40 instruction beforehand.

The last block before the new compensation call ends with the compensation vector in the normal position at the end point. The repeat compensation process is executed as described under “Begin compensation” on page 8–142.

Change in offset number D

The offset number D can be changed in compensation mode. In this case, an altered tool radius becomes effective at the beginning of the block in which the new D number is programmed. The full change in radius is not achieved until the end of the block, i.e. the change is implemented continuously over the entire block. This also applies to circular interpolation.

Cancellation of com pensation using M2

If compensation mode is aborted by means of M2 (program end) without a programmed G40 instruction, then the last block ends with coordinates corresponding to the compensation vector in normal position. No compensatory movement is executed. The program ends with this tool position.

Critical machining operations

When programming machining operations, watch out for cases where the contour path at inner corners is smaller than the tool radius and, with two consecutive inner corners, smaller than the diameter.

This type of programming error must be avoided!

Check sequences of several blocks to make sure that the contour does not contain any “bottlenecks”.

When you carry out a test/dry run, use the largest available tool radius.

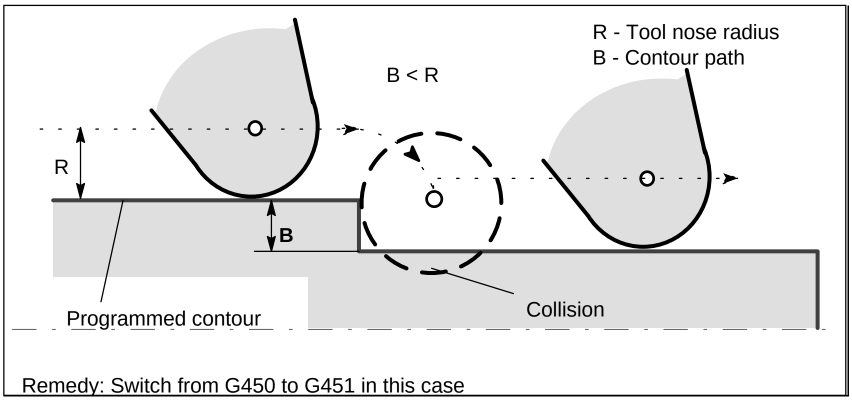


Fig.3-39 Critical maching operation, example shows transition circle

8.6.8 Example of tool radius compensation

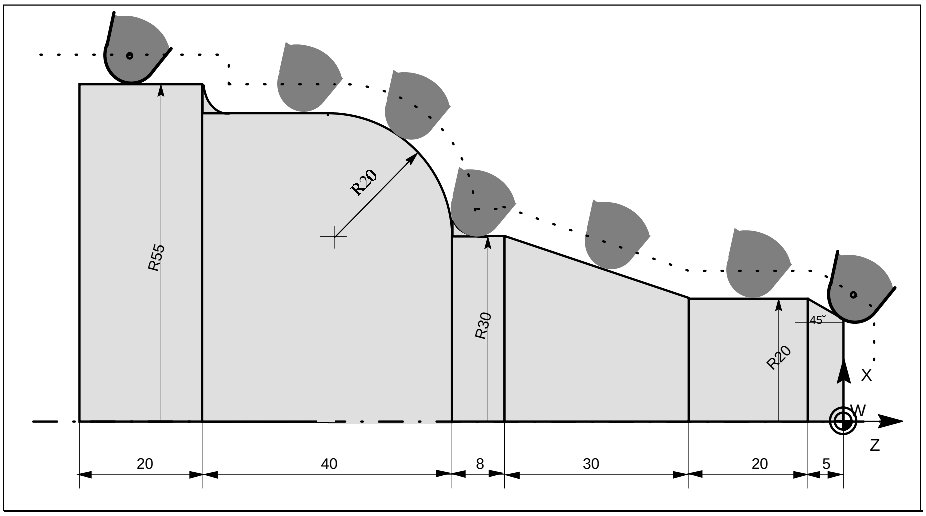


Fig.8-40 Example of tool radius compensation, tool nose radius magnified

Programming example

```
N1 ;                              ;Contour section
N2 T1                             ;Tool 1 with offset D1
N10 G22 F... S... M...            ;Radius dimension specification,
                                   technological values
N15 G54 G0 G90 X100 Z15
N20 X0 Z6
N30 G1 G42 G451 X0 Z0             ;Begin compensation mode
N40 G91 X20 CHF=(5* 1.41)         ;Insert chamfer
N50 Z-25
N60 X10 Z-30
N70 Z-8
N80 G3 X20 Z-20 CR=20
N90 G1 Z-20
N95 X5
N100 Z-25
N110 G40 G0 G90 X100              ;End compensation mode
N120 M2
```

8.7 Miscellaneous function M

Functionality

Miscellaneous function M can be used, for example, to initiate switching operations such as “Coolant ON/OFF”, among other tasks.

The control system manufacturer preassigns certain functions to a small number of the M functions. The others can be freely assigned to functions by the user.

A block may contain a maximum of 5 M functions.

Note

You will find an overview of all M functions reserved and used in the control system in Section 8.1.5. “List of instructions“

Programming

M...

Activation

Activation in blocks with axis movements:
If functions M0, M1 and M2 are programmed in a block that includes axis movements, then they take effect after the traversing movements have been executed.

Functions M3, M4 and M5 are transferred to the internal interface control before the traversing movements. The axis movements are not executed until the spindle has run up in M3 or M4. In the case of M5, however, the axis movements commence before the spindle has reached a standstill.

All the other M functions are transferred to the internal interface control at the same time as the traversing movements.

If you wish to program an M function specifically before or after an axis movement, then insert a separate block with the M function.

Remember: This block will interrupt G64 continuous path mode and generate an exact stop!

Programming example

```
N10 S...
N20 X... M3                ;M function in block with axis movement Spindle runs
                            up before X axis movement
N180 M78 M67 M10 M12 M37
                           ;Max. 5 M functions in block
```

8.8 Arithmetic parameters R

Functionality

If you want an NC program in which you can vary the values to be processed, or if you simply needed to compute arithmetic values, then you can use R (arithmetic) parameters. The control system will calculate or set the values you need when the program is executed.

An alternative method is to input the arithmetic parameter values directly. If the R parameters already have value settings, then they can be assigned in the program to other NC addresses that have variable values.

Programming

R0=...
to
R249=...
(to R299=..., if there are no machining cycles)

Explanation

250 arithmetic parameters with the following classification are available:

R0 ... R99	-	for free assignment
R100 ... R249	-	transfer parameters for machining cycles.
R250 ... R299	-	internal arithmetic parameters for machining cycles.

If you do not intend to use machining cycles (see Section NO TAG “Machining Cycles”), then this range of arithmetic parameters is also available for your use.

Value assignment

You can assign values in the following range to the R parameters:

± (0.000 0001 ... 9999 9999)
(8 decimal places and sign and decimal point).

The decimal point can be omitted for integer values. A positive sign can also be omitted.

Example:
R0=3.5678 R1=-37.3 R2=2 R3=-7 R4=-45678.1234

You can assign an extended numerical range using exponential notation:

± (10^{-300} ... 10^{+300}).

The value of the exponent is typed after the characters EX. Maximum number of characters: 10 (including sign and decimal point).
Value range of EX: -300 to +300.

Example:
R0=-0.1EX-5 ;Meaning: R0 = -0,000 001
R1=1.874EX8 ;Meaning: R1 = 187 400 000

Note: Several assignments (including arithmetic expressions) can be programmed in one block.

Assignment to other addresses

You can obtain a flexible NC program by assigning arithmetic parameters or arithmetic expressions with R parameters to other NC addresses. Values, arithmetic expressions or R parameters can be assigned to any NC address with the exception of addresses N, G and L.

When making assignments of this kind, type the character “=” after the address character. Assignments with a negative sign are also permitted.

If you wish to make assignments to axis addresses (traversal instructions), then you must do so in a separate program block.

Example:
N10 G0 X=R2 ;Assignment to X axis

Arithmetic operations / functions

Operators/arithmetic functions must be programmed using the normal mathematical notation. Processing priorities are set by means of round brackets. Otherwise the “multiplication/division before addition/subtraction” rule applies. Degrees are specified for trigonometric functions.

Programming example: R parameter

```
N10 R1= R1+1          ;The new R1 is product of old R1 plus 1
N20 R1=R2+R3    R4=R5-R6    R7=R8* R9    R10=R11/R12
N30 R13=SIN(25.3)     ;R13 is the sine of 25.3 degrees
N40 R14=R1*R2+R3      ;“Multiplication/division before addition/subtraction” rule
                       R14=(R1*R2)+R3
N50 R14=R3+R2*R1      ;Result as for block N40
N60 R15=SQRT(R1*R1+R2*R2)
```

;Meaning: R15= $\sqrt{R1^2 + R2^2}$

Programming example: Assignment to axes

```
N10 G1 G91 X=R1 Z=R2 F300
N20 Z=R3
N30 X=-R4
N40 Z=-R5
...
```

8.9 Program branches

8.9.1 Labels - destination for program branches

Functionality

Labels are used to mark blocks as the branch destination for branches in the program sequence.

Labels can be selected freely, but must have a minimum of 2 and a maximum of 8 letters or digits. However the first two characters must be letters or underscore characters.

Labels end in a colon in the block that is to act as a branch destination. They are always positioned at the beginning of the block. If the block also has a block number, then the label is positioned after the number.

Labels must be unique within the same program.

Programming example

```
N10 MARKE1: G1 X20  ;MARKE1 is label, branch destination
...
TR789: G0 X10 Z20   ;TR789 is label, branch destination No block number
```

8.9.2 Unconditional program branches

Functionality

NC programs process the blocks they contain in the same order as they were typed by the programmer.

The processing sequence can be altered through the insertion of program branches.

The only possible branch destination is a block with label. This block must be included in the program.

An unconditional branch instruction must be programmed in a separate block.

Programming

GOTOF Label ;Branch forwards
GOTOB Label ;Branch backwards

STL	Explanation
GOTOF	Branch direction forwards (towards last block in program)
GOTOB	Branch direction backwards (towards first block in program)
Label	Selected character string for label

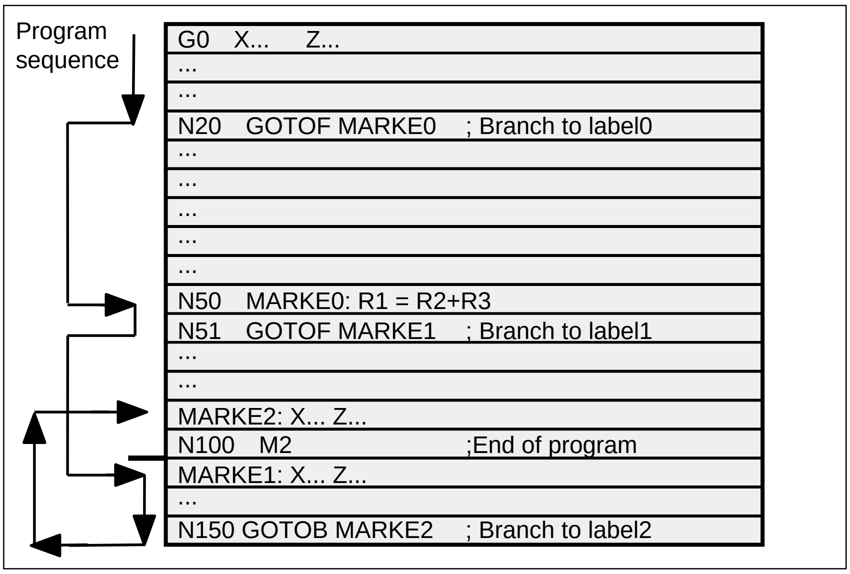


Fig.8-41 Example of unconditional branches

8.9.3 Conditional branches

Functionality

Branch conditions are formulated after the IF instruction. If the branch condition is fulfilled (value not equal to zero), then the program branches. The branch destination can only be a block with corresponding label. This block must be contained within the program.

Conditional branch instructions must be programmed in a separate block. Several conditional branch instructions can be programmed in the same block.

You can reduce program processing times significantly by using conditional program branches.

Programming

```
IF condition GOTOF Label    ;Branch forwards
IF condition GOTOB Label    ;Branch backwards
```

STL	Explanation
GOTOF	Branch direction forwards (towards last block in program)
GOTOB	Branch direction backwards (towards first block in program)
Label	Selected character string for label
IF	Introduction of branch condition
Condition	Arithmetic parameter, arithmetic expression in comparison for formulation of condition

Comparison operations

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

The comparison operations are used to formulate branch conditions. Arithmetic expressions can also be compared.

The result of comparison operations is either “fulfilled” or “not fulfilled”. “Not fulfilled” is equivalent to a value of zero.

Programming example for comparison operators

```
R1>1                 ;R1 greater than 1
1 < R1               ;1 less than R1
R1<R2+R3             ;R1 less than R2 plus R3
R6>=SIN( R7*R7)      ;R6 greater than or equal to SIN (R7)²
```

Programming example

```
N10 IF R1 GOTOF MARKE1      ;If R1 is not zero, branch to block with
                             MARKE1
...
N100 IF R1>1 GOTOF MARKE2   ;If R1 is greater than 1, branch to block with
                             MARKE2
...
```

```
N1000 IF R45==R7+1 GOTOB MARKE3
                                        ;If R45 is equal to R7 plus 1, branch to block
                                         with MARKE3
...
```

Several conditional branches in block:

```
...
N20 IF R1==1 GOTOB MA1 IF R1==2 GOTOF MA2 ...
...
```

Note: The program branches at the first fulfilled condition.

8.9.4 Example of program with branches

Objective of program

Approach points on an circle segment:
Let us assume the following values:

Start angle:	30°	in R1
Circle radius:	32 mm	in R2
Position spacing:	10°	in R3
Number of points:	11	in R4
Position of circle center point in Z:	50 mm	in R5
Position of circle center point in X:	20 mm	in R6

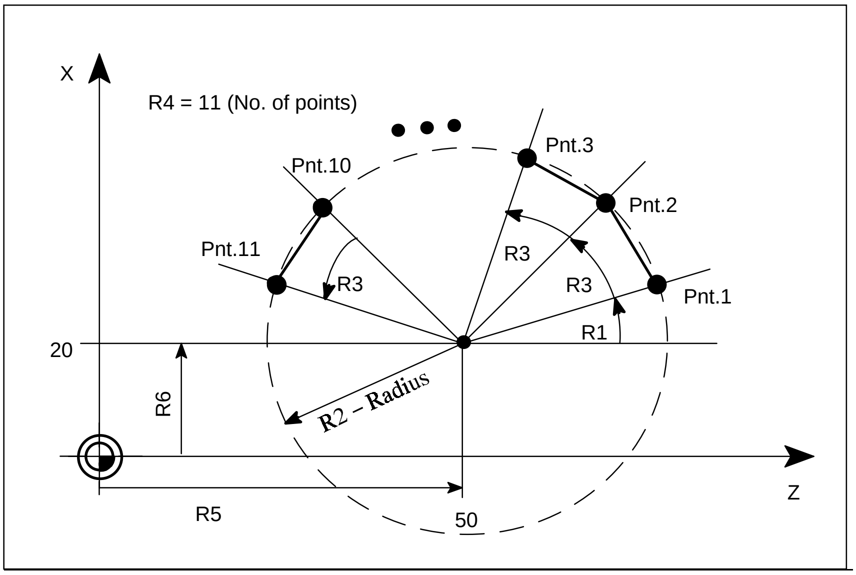


Fig.8-42 Approaching points along a circle segment

Programming example

```
N10 R1=30   R2=32   R3=10   R4=11   R5=50   R6=20
                        ;Assignment of start values
N20 MA1: G0 Z=R2*COS (R1)+R5   X=R2*SIN(R1)+R6
                        ;Computation and assignment to axis addresses
N30   R1=R1+R3   R4= R4-1
N40 IF R4 > 0 GOTOB MA1
N50   M2
```

Explanation

The initial conditions are assigned to the appropriate arithmetic parameters in block N10. The coordinates in X and Z are calculated in N20 and processed.

In N30, R1 is increased by the angle R3 and R4 is decremented by 1. If R4 > 0, N20 is processed again. Otherwise the program continues with N50 and end of program.

8.10 Subroutine technique

Application

There is no essential difference between a main program and a subroutine.

Subroutines contain frequently recurring machining sequences, for example, certain contour shapes. This type of subroutine is called at the appropriate locations in the main program and then processed.

One type of subroutine is the machining cycle. Machining cycles contain generally applicable machining operations (e.g. thread cutting, stock removal, etc.). By supplying these cycles with values by means of the arithmetic parameters provided, you can adapt the program to your specific application (see Section “Machining Cycles”).

Structure

Subroutines are structured in exactly the same way as main programs (see Section “Program structure”). M2 (end of program) is programmed in the last block of the subroutine sequence in exactly the same way as for main programs. In this case, program end means a return to the program level that called the subroutine.

Program end

The M2 end–of–program instruction can be substituted by the end instruction RET in subroutines.

RET must be programmed in a separate block.

An RET instruction must be used when it is necessary to avoid an interruption in continuous path mode G64 when the program branches back to main program level from the subroutine. If an M2 instruction is programmed, G64 mode is interrupted and an exact stop generated.

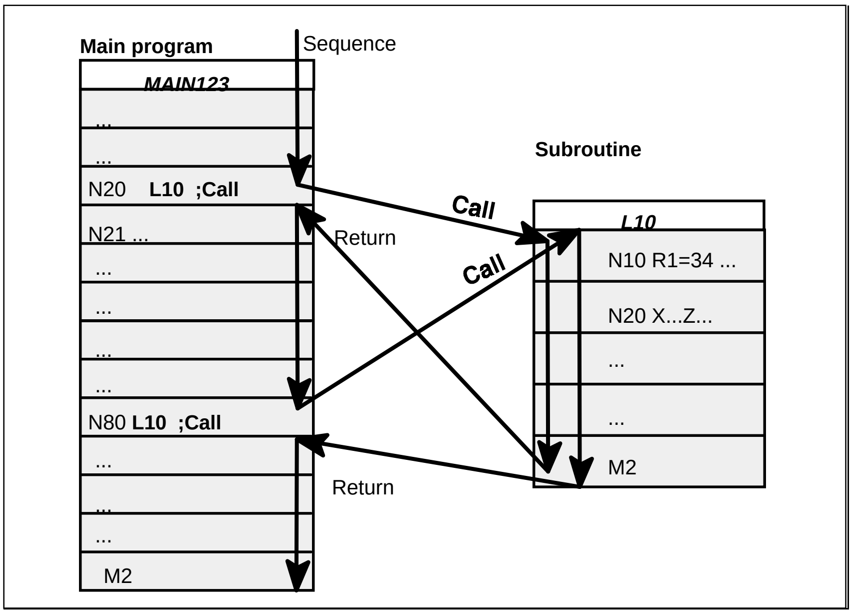


Fig. 8-43 Example of program sequence in which subroutine is called twice

Subroutine name

A subroutine is given its own specific name so that it can be selected from all the others. The name can be chosen freely subject to the following conditions when the subroutine is generated:

— The first two characters must be letters
— The others may be letters, digits or underscore
— Maximum of 8 characters in total
— No dashes (see Section “Character set”)

The same rules apply as for main program names.
Example: BUCHSE7

There is the additional option of using the address word L... for subroutines. This value may have 7 decimal places (integers only).

Please note: Leading zeros are interpreted as distinguishing digits in the L address.
Example: L128 is not L0128 or L00128!
These are 3 different subroutines!

Subroutine call

Subroutines are called by their name in a program (main program or subroutine). These calls must be programmed in separate blocks.
Example:
N10 L785 ;Call subroutine L785
N20 WELLE7 ;Call subroutine WELLE7

Program repeat P...

If a subroutine must be repeated several times in succession, then enter the number of runs under address P after the subroutine name in the block containing the subroutine call. A maximum of 9999 runs can be programmed (P1 ... P9999).
Example:
N10 L785 P3 ;Call subroutine L785, 3 runs

Nesting depth

It is not only possible to call subroutines in main programs, but also in other subroutines. There is a total of 4 program levels (including the main program level) available for programming this type of nested call.

Note: If you are working with machining cycles, please remember that these also need one of the four program levels.

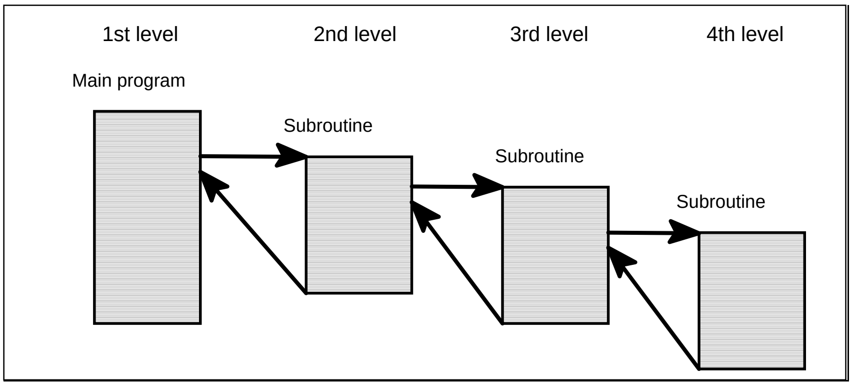


Fig.8-44 Sequence with four program levels

Information

It is possible to change modal G functions, e.g. G90 -> G91, in subroutines. Make sure that all modal functions are set in the way you require when the program branches back to the level on which the subroutine was called.

The same applies to the arithmetic (R) parameters. Make sure that the arithmetic parameters you are using in the upper program levels do not change to different settings in lower levels.

Cycles 9

Preface

Cycles are process-related subroutines that support general implementation of specific machining processes such as, for example, drilling, stock removal or thread cutting. The cycles are adapted to the specific problem in hand by means of supply parameters.

Standard cycles for turning applications are provided in the system.

9.1 General Information about Standard Cycles

This section provides general programming notes for SIEMENS standard cycles.

9.1.1 Overview of Cycles

LCYC82	Drilling, spot-facing
LCYC83	Deep hole drilling
LCYC840	Tapping with compensation chuck
LCYC85	Boring
LCYC93	Recess
LCYC94	Undercut (forms E and F to DIN)
LCYC95	Stock removal with relief cuts
LCYC97	Thread cutting

Supply parameters

The arithmetic parameters in the R100 to R249 range are used as supply parameters for cycles.

Before a cycle is called, values must be assigned to its transfer parameters. These value settings are unchanged after the cycle has been executed.

Arithmetic parameters

If you intend to use machining cycles, you must ensure that arithmetic parameters R100 to R249 are reserved for this purpose, and are not used for other functions within the program. The cycles use R250 to R299 as internal arithmetic parameters.

Call and return conditions

G23 (for LCYC93, 94, 95, 97) or G17 (for LCYC82, 83, 840, 85) (diameter programming) must be active before a cycle is called. Otherwise, the error message 17040 illegal axis index is output. The appropriate values for feedrate, spindle speed and spindle direction of rotation must be programmed in the part program if there are no supply parameters for these quantities in the cycle.

G0 G90 G40 are always effective at the end of a cycle.

9.1.2 Error messages and error handling in cycles

Error handling in cycles

Alarms with numbers between 61000 and 62999 are generated in the cycles. In turn, this number range is subdivided into alarm reactions and reset criteria.

Table 9–1 Alarm numbers, reset criteria, alarm reactions

Alarm number	Reaction	Program continued by
61000...61999	Block preparation in the NC is aborted	NC RESET
62000...62999	Block preparation is interrupted, can be continued with NC start after alarm reset	Reset key

The error text that is displayed at the same time as the alarm number provides further details about the cause of the error.

Overview of cycle alarms

The following Table gives an overview of errors that can occur in cycles, the location of their origin and guidance on how to eliminate them.

Table 9–2 Cycle alarms

Alarm Number	Alarm Text	Source (Cycle)	Remedial Action
61001	Thread lead incorrectly defined	LCYC840	Check parameter R106 (R106=0).
61002	“Machining type incorrectly programmed”	LCYC93, 95, 97	The value of parameter R105 for the machining type is incorrectly set and must be altered.
61003	3rd geometry axis missing	LCYC82, 83, 840	Check machine configuration and plane selection (connect 3rd geometry axis).
61101	Reference plane incorrectly defined	LCYC82, 83, 840, 85	Check parameters R101, R103, R104 - R103 = R104, or R103 is not between R101 and R104.
61102	No spindle direction defined	LCYC840	Value in parameter R107 is greater than 4 or less than 3.
61107	“First drilling depth incorrectly defined”	LCYC83	Change the value for 1st drilling depth (first drilling depth is in opposition to total drilling depth)
61601	“Finished part diameter too small”	LCYC94	A finished part diameter of < 3mm is programmed. This setting is illegal.
61602	“Tool width incorrectly defined”	LCYC93	The tool width (parameter R107) does not match the programmed recess type.
61603	“Recess form incorrectly defined”	LCYC93	The recess form is incorrectly programmed.
61606	“Error when preparing the contour”	LCYC95	Check contour subroutine. Check machining type parameter (R105)
61608	“Incorrect tool point direction programmed”	LCYC94	A tool point direction 1 ... 4 that matches the undercut form must be programmed.
61609	“Form incorrectly defined”	LCYC94	Check parameters for undercut form.
61610	“No infeed depth programmed”	LCYC95	The parameter for infeed depth R108 must be set >0 for roughing.

9.2 Drilling, counter boring – LCYC82

Function

The tool drills with the spindle speed and feedrate programmed down to the entered final depth. When the final drilling depth is reached, a dwell time can be programmed. The drill is retracted from the drill hole at rapid traverse rate.

Call

LCYC82

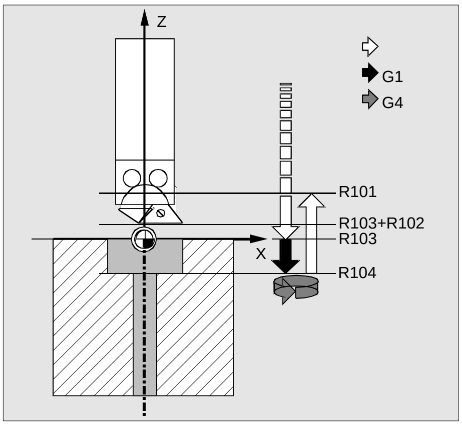


Fig.9-1 Motional sequence and parameters in the cycle

Precondition

The spindle speed and the direction of rotation, as well as the feed of the drilling axis must be defined in the higher-level program.

The drilling position must be approached before calling the cycle in the higher-level program.

The required tool with tool offset must be selected before calling the cycle.

G17 must be active.

Parameters

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time in seconds

Information

R101

The retract plane determines the position of the drilling axis at the end of the cycle.

R102

The safety clearance acts on the reference plane, i.e. the reference plane is shifted forward by an amount corresponding to the safety clearance.

The direction in which the safety clearance acts is automatically determined by the cycle.

R103

The starting point of the drill hole shown in the drawing is programmed under the reference plane parameter.

R104

The drilling depth is always programmed as an absolute value with refer to workpiece zero.

R105

The dwell time at drilling depth (chip breakage) is programmed in seconds under R105.

Motional sequence

Position reached prior to beginning of cycle:
last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0.
2. Traverse to final drilling depth with G1 and the feedrate programmed in the higher-level program.
3. Execute dwell time to final drilling depth.
4. Retract to retract plane with G0.

Example

Drilling - counter boring

The program produces a 27 mm deep drill hole in the position X24 Y15 in the XY plane using the cycle LCYC82. The dwell time is 2 s, and the safety clearance in the drilling axis (here: Z) amounts to 4 mm. On completion of the cycle, the tool stands on X24 Y15 Z110.

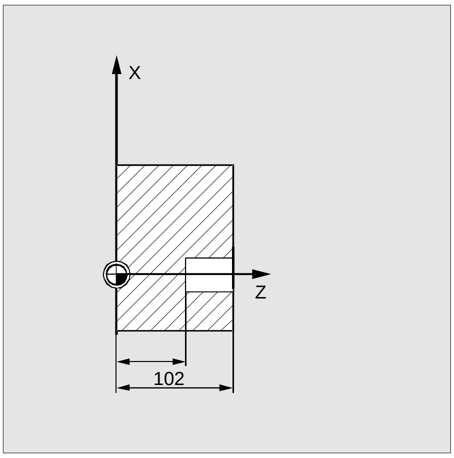


Fig.9-2 Example drawing

N10 G0 G17 G90 F500 T2 D1 S500 M4	; Define technology values
N20 X24 Y15	; Approach drilling position
N25 G17	
N30 R101=110 R102=4 R103=102 R104=75	; Supply parameters
N35 R105=2	; Supply parameters
N40 LCYC82	; Call cycle
N50 M2	; End of program

9.3 Deep hole drilling – LCYC83

Function

The deep-hole drilling cycle produces center holes down to the final drilling depth by repeated, step-by-step deep infeed whose maximum amount can be parameterized. The drill can be retracted either to the reference plane for swarf removal after each infeed depth or by 1 mm in each case for chip breakage.

Call

LCYC83

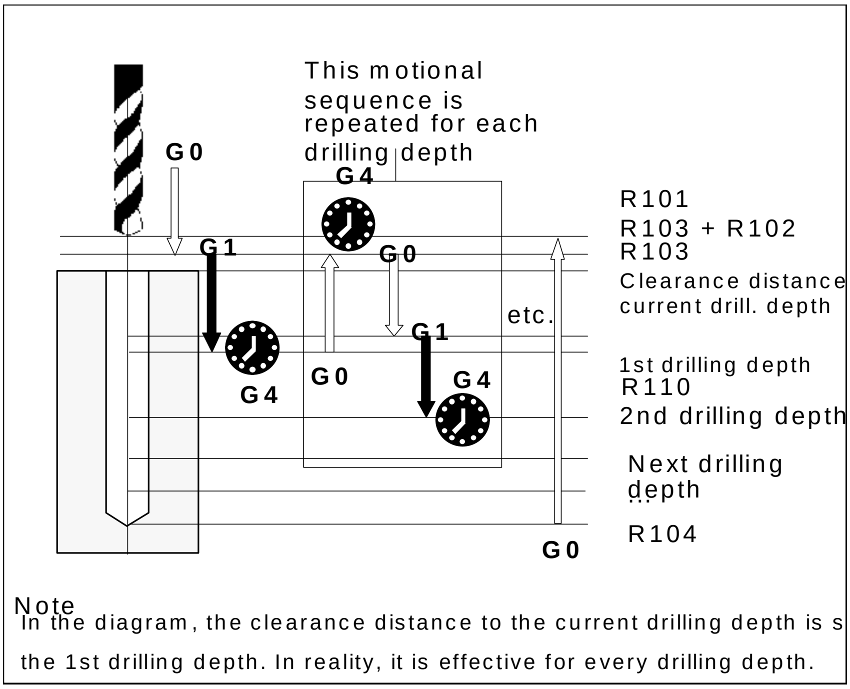


Fig.9-3 Motional sequence and parameters in the cycle

Precondition

The spindle speed and the direction of rotation must be defined in the higher-level program.

The drilling position must be approached before calling the cycle in the higher-level program.

Before calling the cycle, a tool offset for the drill must be selected.

G17 must be active.

Parameters

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance, enter without sign
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time to drilling depth (chip breakage)
R107	Feed for drilling
R108	Feed for first drilling depth
R109	Dwell time at starting point and for swarf removal
R110	First drilling depth (absolute)
R111	Absolute degression, enter without sign
R127	Machining type: Chip breakage = 0 Swarf removal = 1

Information

R101

The retract plane determines the position of the drilling axis at the end of the cycle.

The cycle is programmed on the assumption that the retract plane positioned in front of the reference plane, i.e. its distance to the final depth is greater.

R102

The safety clearance acts on the reference plane, i.e. the reference plane is shifted forward by an amount corresponding to the safety clearance.

The direction in which the safety clearance acts is automatically determined by the cycle.

R103

The starting point of the drill hole shown in the drawing is programmed under the reference plane parameter.

R104

The drilling depth is always programmed as an absolute value regardless of how G90/91 is set prior to cycle call.

R105

The dwell time at drilling depth (chip breakage) is programmed in seconds under R105.

R107, **R108**

The feed for the first drilling stroke (under R108) and for all subsequent drilling strokes (under R107) are programmed via the parameters.

R109

A dwell time at the starting point can be programmed in seconds under parameter R109.

The dwell time at the starting point is executed only for the “with swarf removal” variant.

R110

Parameter R110 determines the depth of the first drilling stroke.

R111

Parameter R111 for the absolute degression value determines the amount by which the current drilling depth is reduced with subsequent drilling strokes.

The second drilling depth corresponds to the stroke of the first drilling depth minus the absolute degression value provided that this value is greater than the programmed absolute degression value.

Otherwise, the second drilling depth also corresponds to the absolute degression value.

The next drilling strokes correspond to the absolute degression value provided that the remaining degression depth is still greater than twice the absolute degression value. The remainder is then distributed evenly between the last two drilling strokes.

If the value for the first drilling depth is in opposition to the total drilling depth, the error message
61107 ”First drilling depth incorrectly defined“
is displayed, and the cycle is not executed.

R127

Value 0:
The drill travels 1 mm clear for chip breakage after it has reached each drilling depth.

Value 1:
The drill travels to the reference plane, which is shifted forward by an amount corresponding to the safety clearance for swarf removal after each drilling depth.

Motional sequence

Position reached prior to beginning of cycle:
last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0.
2. Traverse to first drilling depth with G1; the feedrate results from the feedrate programmed prior to cycle call after it has been computed with the setting in parameter R109 (feedrate factor).
   Execute dwell time at drilling depth (parameter R105).

   With chip breakage selected:

   Retract by 1 mm from the current drilling depth with G1 for chip breakage.

   With swarf removal selected:

   Retract for swarf removal to reference plane shifted forward by an amount corresponding to the safety clearance with G0 for swarf removal, executing the dwell time at starting point (parameter R106), approach last drilling depth minus clearance distance calculated in the cycle using G0.
3. Traverse to next drilling depth with G1 and the programmed feed; this motional sequence is continued as long as the final drilling depth is reached.
4. Retract to retract plane with G0.

Example: Deep-hole drilling

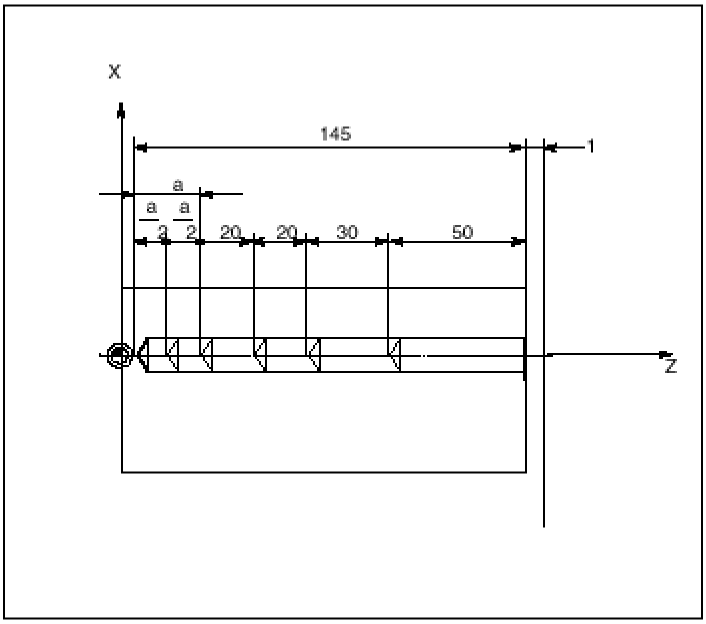


Fig. 9-4 Example drawing

```
;This program executes the cycle LCYC83 at position X0.
N100 G0 G18 G90 T4 S500 M3            ;Define technology values
N110 Z155
N120 X0                               ;Approach first drilling position
N125 G17
R101=155 R102=1 R103=150
R104=5 R105=0 R109=0 R110=100         ;Parameter assignment
R111=20 R107=500 R127=1 R108=400
N140 LCYC83                           ;1st call of cycle
N199 M2
```

9.4 Tapping with compensating chuck – LCYC840

Function

The tool drills with the programmed spindle speed and direction of rotation down to the entered thread depth. The feed of the drilling axis results from the spindle speed. This cycle can be used for tapping with compensating chuck and spindle actual-value encoder. The direction of rotation is automatically reversed in the cycle. The retract can be carried out at a separate speed. M5 acts after the cycle has been executed (spindle stop).

Call

LCYC840

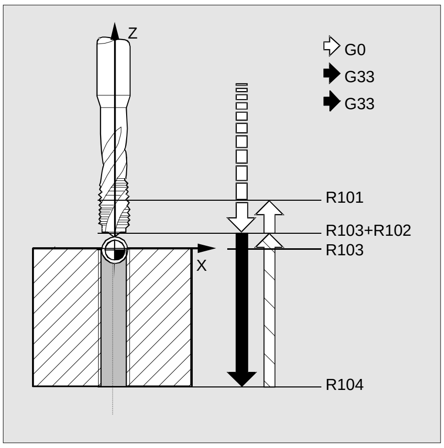


Fig.9-5

Precondition

This cycle can only be used with a speed-controlled spindle with position encoder. The cycle does not check whether the actual-value encoder for the spindle really exists.

The spindle speed and the direction of rotation must be defined in the higher-level program.

The drilling position must be approached before calling the cycle in the higher-level program.

The required tool with tool offset must be selected before calling the cycle.

G17 must be active.

Parameters

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R106	Thread lead as value value range: 0.001 .... 2000.000 mm
R126	Direction of rotation of spindle for tapping Value range: 3 (for M3), 4 (for M4)

Information

R101 -R104 See LCYC84

R106 Thread lead as value

R126 The tapping block is executed with the direction of rotation of spindle programmed under R126. The direction of rotation is automatically reversed in the cycle.

Motional sequence Position reached prior to beginning of cycle:
- last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0
2. Tapping down to final drilling depth with G33
3. Retract to reference plane shifted forward by an amount corresponding to the safety clearance with G33
4. Retract to retract plane with G0

Example This program is used for tapping on the position X0; the Z axis is the drilling axis. The parameter for the direction of rotation R126 must be parameterized. A compensating chuck must be used for machining. The spindle speed is defined in the higher-level program.

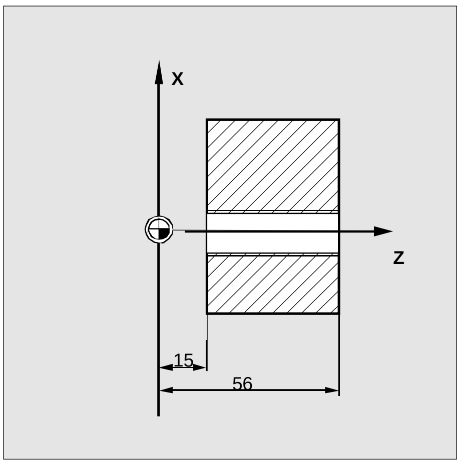


Fig.9-6 Example drawing

N10 G0 G17 G90 S300 M3 D1 T1	; Define technology values
N20 X35 Y35 Z60	; Approach drilling position
G17	
N30 R101=60 R102=2 R103=56 R104=15	; Parameter assignment
N40 R106=0.5 R126=3	; Parameter assignment
N40 LCYC840	; Cycle call
N50 M2	; End of program

9.5 Boring – LCYC85

Function

The tool drills with the spindle speed and feedrate programmed down to the entered final drilling depth. When the final drilling depth is reached, a dwell time can be programmed. The approach and retract movements are carried out with the feedrates programmed under the respective parameters.

Call

LCYC85

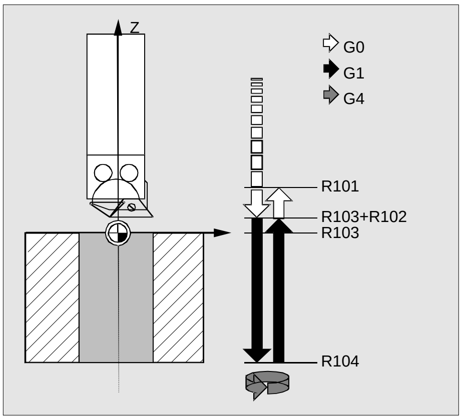


Fig.9-7 Motional sequence and parameters of the cycle

Precondition

The spindle speed and the direction of rotation must be defined in the higher-level program.

The drilling position must be approached before calling the cycle in the higher-level program.

Before calling the cycle, the respective tool with tool offset must be selected.

Parameters

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time at drilling depth in seconds
R107	Feed for drilling
R108	Feed when retracting from drill hole

Information

Parameters R101 - R105 see LCYC82

R107

The feed value defined here acts for drilling.

R108

The feed value entered under R108 acts for retracting from the drill hole.

Motional sequence

Position reached prior to beginning of cycle:
last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0
2. Traverse to final drilling depth with G1 and the feed programmed under parameter R106.
3. Execute dwell time at final drilling depth.
4. Retract to reference plane shifted forward by an amount corresponding to the safety clearance with G1 and the retract feed programmed under R108.

Example

The cycle LCYC85 is called in Z70 and X50 in the ZX plane. The Y axis is the drilling axis. No dwell time is programmed. The workpiece upper edge is at Y=102.

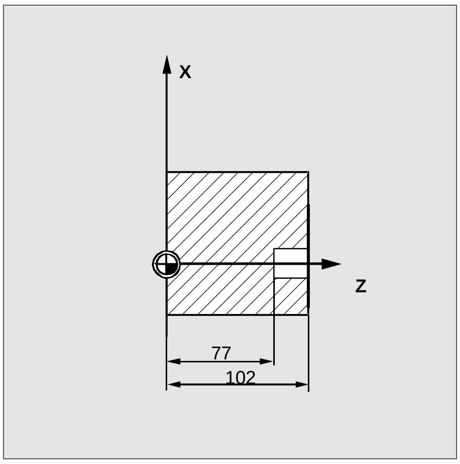


Fig.9-8 Example drawing

```
N10 G0 G90 G18 F1000 S500 M3 T1 D1        ; Define technology values
N20 Z70 X50 Y105                          ; Approach drilling position
N30 R101=105 R102=2 R103=102 R104=77      ; Define parameters
N35 R105=0 R107=200 R108=400              ; Define parameters
N40 LCYC85                                ; Call drilling cycle
N50 M2                                    ; End of program
```

9.6 Recess cycle – LCYC93

Fuction

The recess cycle is designed to produce symmetrical recesses for longitudinal and face machining on cylindrical contour elements. The cycle is suitable for machining internal and external recesses.

Call

LCYC93

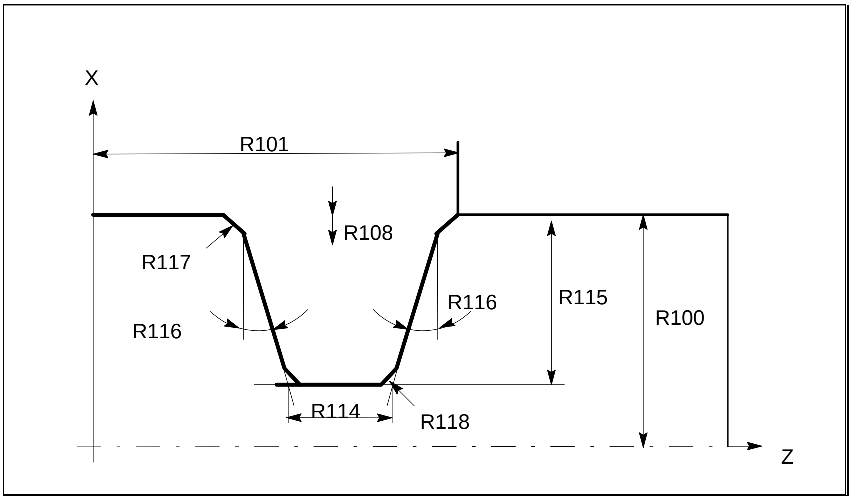


Fig.9-9 Parameters in the recess cycle in longitudinal machining

Precondition

The recess cycle can only be called if G23 (diameter programming) is active. The tool offset of the tool whose tool nose width has been programmed with R107 must be activated before the recess cycle is called. The zero position of the tool nose faces machine zero.

Parameters

Table 9–3 Parameters for LCYC93 cycle

Parameter	Meaning, Value Range
R100	Starting point in facing axis
R101	Starting point in longitudinal axis
R105	Machining method, Value range 1 ... 8
R106	Finishing allowance, without sign
R107	Tool nose width, without sign
R108	Infeed depth , without sign
R114	Recess width, without sign
R115	Recess width, without sign
R116	Flank angle, without sign, between 0 <= R116 < = 89.999 degrees
R117	Chamfer on rim of recess
R118	Chamfer on recess base
R119	Dwell time on recess base

Information

R100

The recess diameter in X is specified in parameter R100.

R101

R101 determines the point at which the recess starts in the Z axis.

R105

R105 defines the recess variant:

Table 9–4 Recess variants

Value	Longitudinal/Facing	External/Internal	Starting Point Position
1	L	A	Left
2	P	A	Left
3	L	I	Left
4	P	I	Left
5	L	A	Right
6	P	A	Right
7	L	I	Right
8	P	I	Right

If the parameter is set to any other value, the cycle is aborted with the alarm

61002 “Machining type incorrectly programmed”.

R106

Parameter R106 determines the finishing allowance for roughing of the recess.

R107

Parameter R107 determines the tool nose width of the recessing tool. This value must correspond to the width of the tool actually used.

If the tool nose of the active tool is wider, the contour of the programmed recess will be violated. Such violations are not monitored by the cycle.

If the programmed tool nose width is wider than the recess width at the base, the cycle is aborted with the alarm

G1602 “Tool width incorrectly defined”.

R108

By programming an infeed depth in R108, it is possible to divide the axis-parallel recessing process into several infeed depths. After each infeed, the tool is retracted by 1 mm for chip breakage.

Recess form

Parameters R114 ... R118 determine the form of the recess. The cycle always bases its calculation on the point programmed under R100, R101.

R114

The recess width programmed in parameter R114 is measured on the base. The chamfers are not included in the measurement.

R115

Parameter R115 determines the depth of the recess.

R116

The value of parameter R116 determines the angle of the flanks of the recess. When it is set to “0”, a recess with axis-parallel flanks (i.e. rectangular form) is machined.

R117

R117 defines the chamfers on the recess rim.

R118

R118 defines the chamfers on the recess base.

If the values programmed for chamfers do not produce a meaningful recess contour, then the cycle is aborted with the alarm

61603 “Recess form incorrectly defined”.

R119

The dwell time on the recess base to be entered in R119 must be selected such that at least one spindle revolution can take place during the dwell period. It is programmed to comply with an F word (in seconds).

Motional Sequence

Position reached prior to beginning of the cycle:

- Any position from which each recess can be approached without risk of collision.

The cycle produces the following motional sequence:

- Approach with G0 starting point cacluated internally in the cycle.
- Execute depth infeeds:
  Roughing in parallel axes down to base, taking finishing allowance into account. Tool travels clear for chip breakage after each infeed.
- Execute width infeeds:
  Width infeeds are executed perpendicular to the depth infeed with G0, the roughing process for machining the depth is repeated.

The infeeds both for depth and width are distributed evenly with the highest possible value.

- Rough the flanks. Infeed along the recess width is executed in several steps if necessary.
- Finish-machine the whole contour, starting at both rims and working towards center of recess base, at the feedrate programmed before the cycle call.

Example

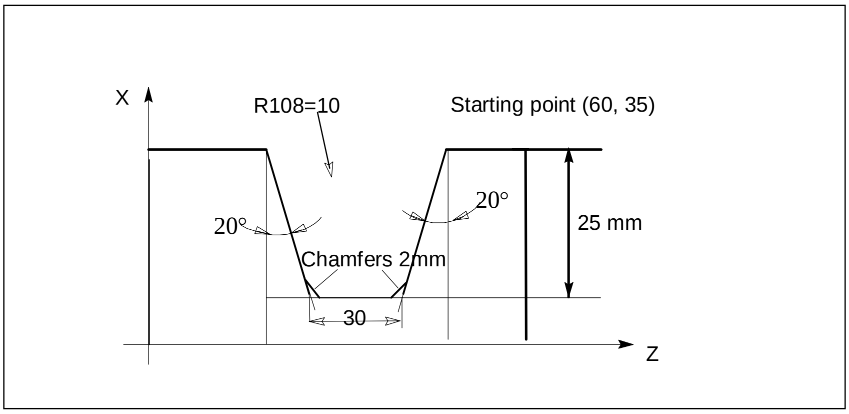


Fig.9-10 Example diagram

;A recess is machined that starts at point (60.35) 25 mm in depth

;and 30 mm in width.

;Two chamfers of 2 mm in length are programmed on the base.

;The finishing allowance is 1 mm.

```
N10 G0 G90 Z100 X100 T2 D1 S300 M3 G23   ;Select start position
N20 G95   F0.3                           ;and technology values
R100=35 R101=60 R105=5 R106=1 R107=12    ;Parameters for cycle call
R108=10 R114=30 R115=25 R116=20
R117=0 R118-2 R119=1

N60 LCYC93                               ;Call recess cycle
N70 G90 G0 Z100 X50                      ;Next position
N100 M2
```

Note on example

The tool offset of the recessing tool must be stored in D1 of tool T2. The tool nose width must be 12 mm.

9.7 Undercut cycle – LCYC94

Function

This cycle machines undercuts of forms E and F in compliance with DIN 509 for normal stressing on finished part diameters > 3 mm.

A tool offset must be activated before the cycle is called.

Call

LCYC94

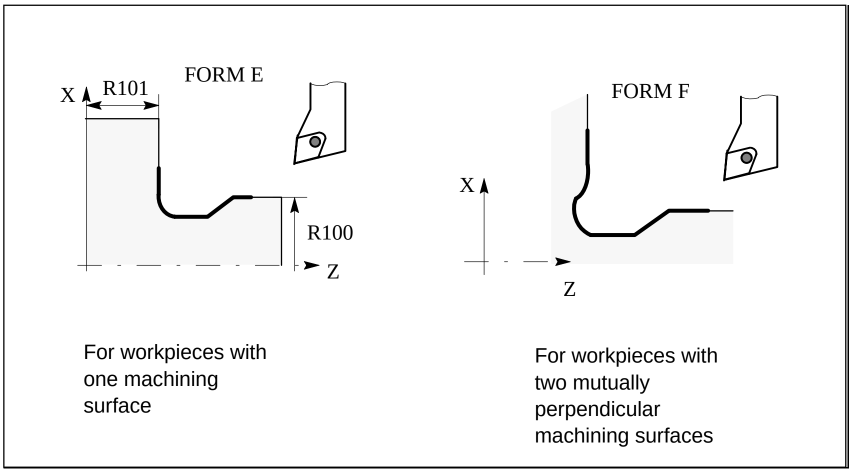


Fig.9-11 Undercut forms E and F

Condition

G23 (diameter programming) must be active for this cycle.

Parameters

Table 9–5 Parameters for LCYC94 cycle

Parameter	Meaning, Value Range
R100	Starting point in facing axis, without sign
R101	Starting point in longitudinal axis
R105	Definition of form: Value 55 for form E Value 56 for form F
R107	Definition of tool point direction: Values 1...4 for directions 1...4

Information

R100

The finished part diameter for the undercut is specified in parameter R100.

If the value programmed for R100 corresponds to a final diameter of <= 3 mm, then the cycle is aborted with the alarm

61601 “Finished part diameter too small”.

R101

R101 determines the finished part dimension in the longitudinal axis.

R105

Forms E and F are defined in DIN509 and must be selected using one of these parameters.

If parameter R105 is set to a value other than 55 or 56, then the cycle is aborted and generates the alarm

61609 “Form incorrectly defined”.

R107

This parameter defines the tool point direction and thus the undercut position. The value set here must correspond to the actual point direction of the tool selected prior to cycle call.

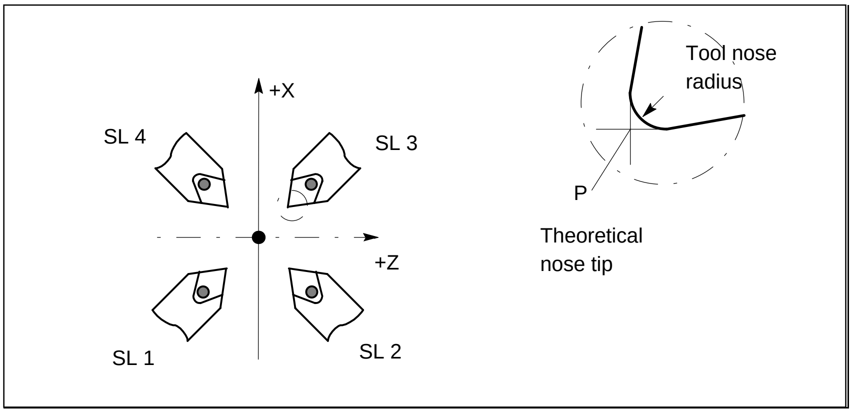


Fig.9-12 Tool point directions 1…4

If the parameter is set to any other value, the alarm

61608 “Incorrect tool point direction programmed” is output and the cycle is aborted.

Motional sequence

Position reached prior to beginning of cycle:

- Any position from which undercut can be approached without risk of collission.

The cycle produces the following motional sequence:

- Approach with G0 starting point calculated internally in the cycle.
- Select tool nose radius compensation in accordance with active tool nose direction and traverse undercut contour at feedrate programmed prior to cycle call.
- Return to starting point with G0 and deselect tool nose radius compensation with G40.

Example

```
;This program machines an undercut of form E.
N50 G0 G90 G23 Z100 X50 T25 D3 S300 M3   ;Select starting position
N55 G95    F0.3                          ;and enter technology values
R100=20 R101=60 R105=55 R107=3           ;Parameters for cycle call
N60 LCYC94                               ;Call undercut for cycle
N70 G90 G0 Z100 X50                      ;Next position
N99 M02
```

9.8 Stock removal cycle – LCYC95

Function

This cycle can machine a contour, which is programmed in a subroutine, in a longitudinal or face machining process, externally or internally, through axis-parallel stock removal.

The technology (roughing/finishing/complete machining) can be selected. The cycle can be called from any chosen collision-free position.

A tool offset must have been activated in the program with the cycle call.

Call

LCYC95

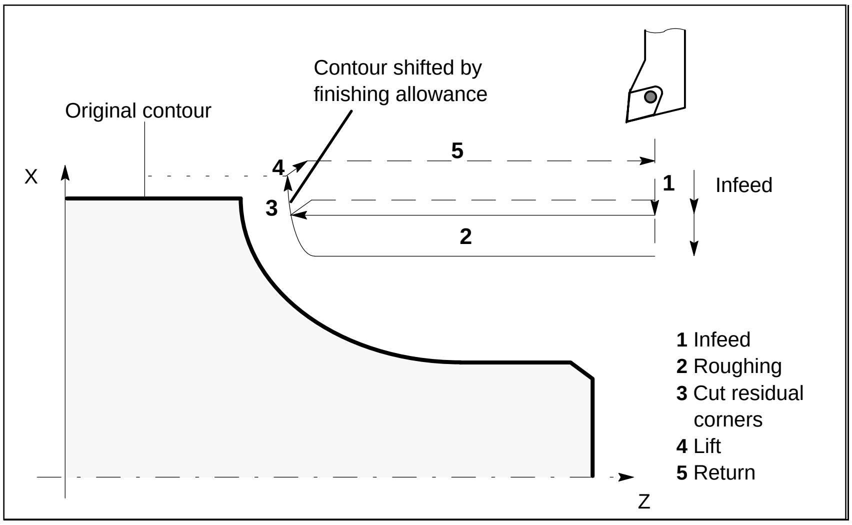


Fig.9-13 Motional sequence with LCYC 95 cycle

Condition

- The cycle requires an active G23 (diameter programming).
- The file SGUD.DEF, which is supplied on the cycles diskette, must be available in the control system.
- The stock removal cycle can be called to the 3rd program level.

Parameters

Table 9–6 Parameters for the LCYC95 cycle

Parameter	Meaning, Value Range
R105	Machining type, value range 1 ... 12
R106	Finishing allowance, without sign
R108	Infeed depth, without sign
R109	Infeed angle for roughing, it should be zero at face machining.
R110	Contour clearance distance for roughing
R111	Feedrate for roughing
R112	Feedrate for finishing

Information

R105

The machining types:

- longitudinal/facing
- internal/external
- roughing/finishing/complete machining

are defined by the parameter determining the type of machining.

When longitudinal machining is selected, the infeed always takes place in the facing axis, and vice versa.

Table 9–7 Variants of stock removal

Value	Longitudinal/Facing (P)	External/Internal (A/I)	Roughing/Finishing/ Complete Machining
1	L	A	Roughing
2	P	A	Roughing
3	L	I	Roughing
4	P	I	Roughing
5	L	A	Finishing
6	P	A	Finishing
7	L	I	Finishing
8	P	I	Finishing
9	L	A	Complete
10	P	A	Complete
11	L	I	Complete
12	P	I	Complete

If any other value is programmed for the parameter, the cycle is aborted and the following alarm output

61002 “Machining type incorrectly programmed”.

R106

A finishing allowance can be programmed in parameter R106.

The workpiece is always rough-machined down to this finishing allowance. In this case, the residual corner produced in the course of each axis-parallel roughing process is immediately cut away in parallel with the contour at the same time. If no finishing allowance is programmed, the workpiece is rough-machined right down to the final contour.

R108

The maximum possible infeed depth for the roughing process is entered under parameter R108. However, the cycle itself calculates the current infeed depth that is applied in rough-machining operations.

R109

The infeed motion for roughing can be executed at an angle which can be programmed in parameter R109. In the face machining process a slanting immerse is not possible, R109 must be programmed to ZERO.

R110

Parameter R110 specifies the distance by which the tool is lifted from the contour in both axes after each roughing operation so that it can be retracted by G0.

R111

The feedrate programmed under R111 applies to all paths on which stock is removed during roughing operations.

If finishing is the only machining type selected, then this parameter has no meaning at all.

R112

The feedrate programmed under R112 is applied for finishing operations. If roughing is the only machining type selected, then this parameter has no meaning at all.

Contour definition

The contour to be machined by stock removal is programmed in a subroutine. The name of the subroutine is transferred to the cycle via the _CNAME variable.

The contour may consist of straight lines and circle segments; radii and chamfers can be inserted. The programmed circle sections can be quarter circles as a maximum.

Undercuts may not be contained in the contour. If an undercut element is detected, the cycle is aborted, and the alarm

61605 “Contour incorrectly defined” is output.

The contour must always be programmed in the direction that is traversed when finishing according to the selected machining direction.

Example of contour programming

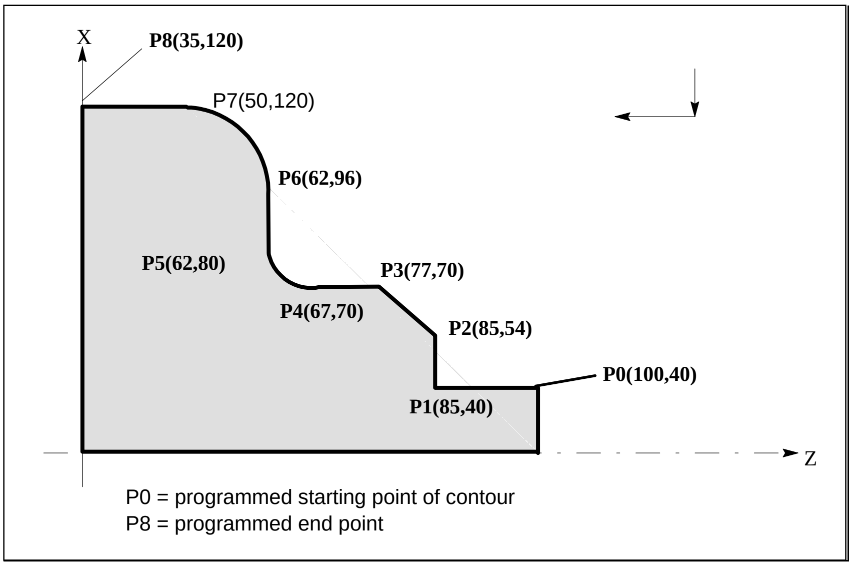


Fig.9-14 Example of contour programming

With the coordinates given in the program, the contour must be programmed for longitudinal external machining as follows:

```
N10 G1 Z100 X40        ;Starting point
N20 Z85                ;P1
N30 X54                ;P2
N40 Z77 X70            ;P3
```

```
N50 Z67                  ;P4
N60 G2 Z62 X80 CR=5      ;P5
N70 G1 Z62 X96           ;P6
N80 G3 Z50 X120 CR=12    ;P7
N90 G1 Z35               ;P8
M17
```

For external facing, the contour must be programmed starting at P8 (35,120) and finishing at P0 (100,40).

Motional sequence

Position reached prior to beginning of cycle:

- Any position from which the contour starting point can be approached without risk of collision.

The cycle produces the following motional sequence

Roughing

- Approach cycle starting point (calculated internally) with G0 in both axes simultaneously.
- Perform depth infeed with the angle programmed under R109 to the next roughing depth.
- Approach roughing cut point in parallel axes with G1 and at a feedrate programmed in R111.
- Travel in parallel with contour along contour + finishing allowance up to the last roughing cut point with G1/G2/G3 and at feedrate R111.
- Lift in each axis by the clearance (in mm) programmed in R110 and retract with G0.
- Repeat this sequence until the final roughing depth is reached.

Finishing

- Approach the cycle starting point in individual axes with G0
- Approach the contour starting point in both axes simultaneously with G0.
- Finish-machine along the contour with G1/G2/G3 and at the feedrate programmed in R112.
- Retract to cycle starting point in both axes with G0.

When finishing is selected, the tool radius compensation is automatically activated internally in the cycle.

Starting point

The cycle automatically calculates the point at which machining must start.
The starting point is always approached in both axes simultaneously for roughing and in individual axes for finishing. In this case, the infeed axis approaches the starting point first.

When complete machining is selected, the tool does not return to the internally calculated starting point after the last roughing cut

Example

The cycle requires the following two programs:

- program with cycle call
- contour subroutine (TESTK1.MPF)

```
;The contour shown in the example must be machined externally
;in a complete machining operation in the longitudinal axis.
;The maximum infeed is 5 mm, the finishing allowance is 1.2 mm,
;and the infeed angle is 7 degrees.

N10 T1 D1 G0 G23 G95 S500 M3 F0.4      ;Definition of technology values
N20 Z125 X162                          ;Collision-free approach position
                                        prior to the call
_CNAME= "TESTK1"                       ;Name of contour subroutine
R105=9 R106=1.2 R108=5 R109=7          ;Set further parameters for
R110=1.5 R111=0.4 R112=0.25            ;cycle call
N20 LCYC95                             ;Cycle call
N30 G0 G90 X81                         ;Re-approach starting position
N35 Z125                               ;Approach in individual axes
N99 M30

Subroutine "TESTK1"
N10 G1 Z100 X40                        ;Starting point
N20 Z85                                ;P1
N30 X54                                ;P2
N40 Z77 X70                            ;P3
N50 Z67                                ;P4
N60 G2 Z62 X80 CR=5                    ;P5
N70 G1 Z62 X96                         ;P6
N80 G3 Z50 X120 CR=12                  ;P7
N90 G1 Z35                             ;P8

M2
```

9.9 Thread cutting – LCYC97

Function

The thread cutting cycle is suitable for cutting external and internal, single-start or multiple-start threads on cylindrical and tapered bodies in the facing or longitudinal axis. Depth infeed is an automatic function.

Whether a right-hand or left-hand thread is produced is determined by the direction of rotation of the spindle, which must be programmed before calling the cycle. Feed and spindle override are not effective in the traversing blocks containing thread cutting operations.

Call

LCYC97

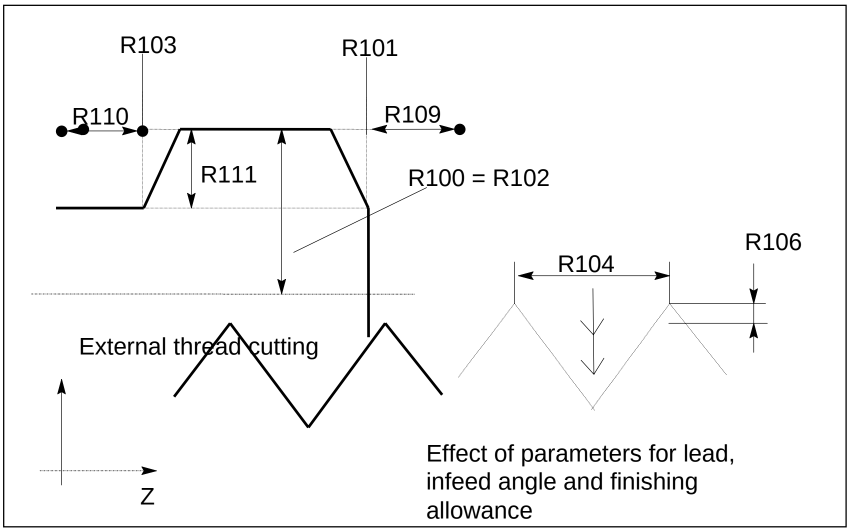


Fig.9-15 Schematic diagram of parameters for thread cutting

Parameters

Table 9–8 Parameters for LCYC97 cycle

Parameter	Meaning, Value Range
R100	Diameter of thread at starting point
R101	Thread starting point in longitudinal axis
R102	Diameter at end point
R103	Thread end point in longitudinal axis
R104	Thread lead as value, without sign
R105	Definition of thread cutting method: Value range: 1, 2
R106	Finishing allowance, without sign
R109	Approach path, without sign
R110	Run-out path, without sign
R111	Thread depth, without sign
R112	Starting point offset, without sign
R113	Number of rough cuts, without sign
R114	Number of threads, without sign

Information

R100, R101

These parameters define the thread starting point in X and Z.

R102, R103

The thread end point is programmed under R102 and R103. In the case of cylindrical threads, one of these parameters has the same value as R100 or R101.

R104

The thread lead is an axis-parallel value and is specified without sign.

R105

Parameter R105 defines whether the thread is machined internally or externally.

R105 = 1: External thread

R105 = 2: Internal thread

If the parameter is set to any other value, the cycle is aborted with the alarm

61002 "Machining type incorrectly programmed".

R106

The programmed finishing allowance is subtracted from the specified thread depth. The remainder is divided into rough cuts.

The finishing allowance is removed in one cut after roughing.

R109, R110

Parameters R109 and R110 specifiy the internally calculated thread approach and run-out paths. The cycle shifts the programmed starting point forward by the approach distance. The run-out path extends the length of the thread beyond the programmed end point.

R111

Parameter R111 defines the total depth of the thread.

R112

An angle value can be programmed in this parameter. This value defines the point at which the first thread cut starts on the circumference of the turned part, i.e. it is a starting point offset.

Possible values for this parameter are between 0.0001 ... + 359.9999 degrees.

If no starting point offset is specified, the first thread automatically starts at the zero-degree marking.

R113

Parameter R113 determines the number of roughing cuts for thread cutting operations. The cycle independently calculates the individual, current infeed depths as a function of the settings in R105 and R111.

R114

This parameter specifies the number of threads. These are arranged symmetrically around the circumference of the turned part.

Longitudinal or face thread

The cycle itself decides whether a thread must be machined in the longitudinal or facing axis. If the angle on the taper is less than or equal to 45 degrees, then the thread is machined as a longitudinal thread, otherwise as a face thread.

Motional sequence

Position reached prior to beginning of cycle:

- Any position from which the programmed thread starting point + approach path can be approached without risk of collision.

The cycle produces the following motional sequence:

- Approach starting point at the beginning of the approach path (calculated internally in the cycle) to cut first thread with G0.
- Infeed for rough cutting according to the infeed method defined under R105.
- Repeat thread cuts according to the programmed number of rough cuts.
- Remove the finishing allowance with G33.
- Repeat the whole sequence for every further thread.

Example

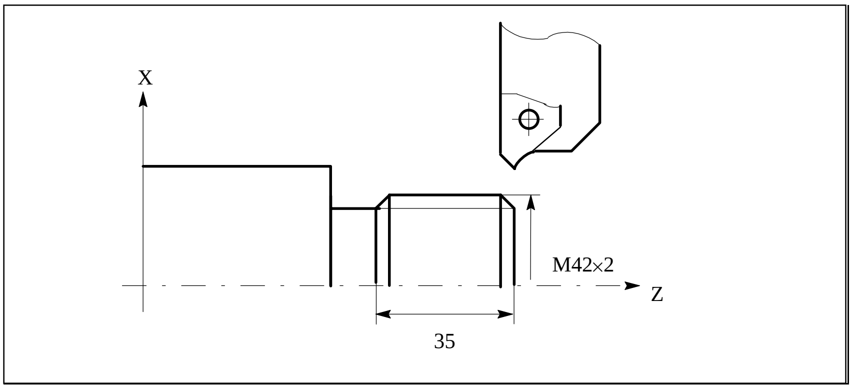


Fig.9-16 Example diagram

;A two-start thread, M42x2, must be machined.

```
N10 G23 G95 F0.3 G90 T1 D1 S1000 M4  ;Define technology values
N20 G0 Z100 X120                     ;Program start position
R100=42 R101=80 R102=42 R103=45      ;Parameters for cycle call
R105=1 R106=1 R109=12 R110=6
R111=4 R112=0 R113=3 R114=2
N50 LCYC97                           ;Cycle call
N100 G0 Z100 X60                     ;Position after cycle end
N110 M2
```

SIEMENS AG A&D MC BMS Postfach 3180 D–91050 Erlangen (Tel. +49 180 / 5050 – 222 [Hotline] Fax +49 9131 / 98 – 2176 [Documentation] Mailto: motioncontrol.docu@erlf.siemens.de)	**Suggestions** **Corrections** for Publication/Manual: SINUMERIK 802S/802C base line User Documentation
From Name Company/dept. Street Zip code: City: Telephone: / Telefax: /	Operation and Programming - Turning Order No.: 6FC5598–4AA01–0BP0 Edition: 08.03 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 802S/C base line Document Structure

General Documentation: **Catalog**

- Turning Milling

User Manual: **Operation and Programming**

- Short Guide for O&P
- Turning
- Milling

User Manual: **Diagnostics Guide**

- Turning Milling

Technical Manual: **Start-Up**

- Quick Start
- 802S base line Start-Up
- 802C base line Start-Up

Technical Manual: **Description of Functions**

- Turning Milling