SIEMENS

SINUMERIK 802D sl

Operation and Programming Nibbling

Valid for

Control	*Software version*
SINUMERIK 802D sl	1

06/2006 Edition

Safety information

This manual contains information that must be observed to ensure your personal safety and to prevent property damage. Notices referring to your personal safety are highlighted in the manual by a safety alert symbol; notices referring to property damage only have no safety alert symbol, and, depending on the degree of hazard, represented as shown below:

⚠ **Danger**

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

⚠ **Warning**

indicates that death or serious injury **may** result if proper precautions are not taken.

⚠ **Caution**

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

Caution

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

Notice

indicates that an undesirable event or state **may** arise if the relevant note is not observed.

If several hazards of different degrees occur, the hazard with the highest degree must always be given priority. A warning notice accompanied by a safety alert symbol indicating a risk of bodily injury can also indicate a risk of property damage.

Qualified personnel

The associated device/system must only be set up and operated using this documentation. Only **qualified personnel** should be allowed to commission and operate the device/system. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Proper use

Please note the following:

⚠ **Warning**

The equipment must only be used for single purpose applications explicitly described in the catalog and in the technical description and it must only be used along with third-party devices and components approved by Siemens. This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

Registered trademarks

All designations with the trademark symbol ® are registered trademarks of Siemens AG. Other designations in this documentation may be trademarks whose use by third parties for their own purposes may infringe the rights of the owner.

Disclaimer of liability

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

Siemens AG
Automation and Drives
Postfach 4848
90437 NÜRNBERG
GERMANY

6FC5398-3CP10-0BA0

Siemens AG, 2006.
Subject to change without prior notice.

Preface

SINUMERIK documentation

The SINUMERIK documentation is organized in 3 parts:

- General Documentation
- User Documentation
- Manufacturer/service documentation

For more detailed information on SINUMERIK 802D sl and other publications covering all SINUMERIK controls (e.g. universal interface, measuring cycles...), please contact your local Siemens office.

A list of documents, updated on a monthly basis, is available on the Internet for the available languages at:
http://www.siemens.com/motioncontrol
Select “Support”/“Technical Documentation”/“Overview of Documents”.

The Internet version of the DOConCD (DOConWEB) is available at:
http://www.automation.siemens.com/doconweb

Target readership of this documentation

This document is designed for machine tool manufacturers. The present documentation provides all information required by the manufacturer to start up the SINUMERIK 802D sl CNC.

Standard scope

This Instruction Manual describes the functionality of the standard scope. Additions or revisions made by the machine tool manufacturer are documented by the machine tool manufacturer.

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

Hotline

If you have any questions, please contact the following hotline:
A&D Technical Support
Phone: +49 (0) 180 / 5050 - 222
Fax: +49 (0) 180 / 5050 - 223
Internet: http://www.siemens.com/automation/support-request

If you have any comments, suggestions, or corrections regarding this documentation, please fax or e-mail them to:
Fax: +49 (0) 9131 / 98 - 63315
E-mail: motioncontrol.docu@siemens.com

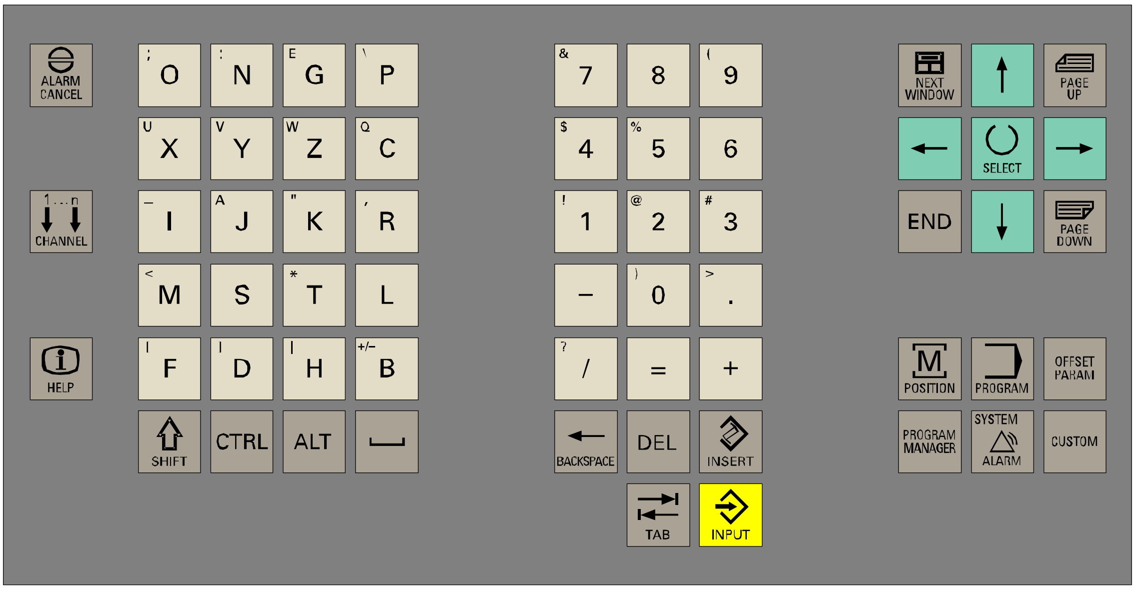
Fax form: See the reply form at the end of the document.

Internet address

http://www.siemens.com/motioncontrol

Contents

SINUMERIK 802D sl Key Definition



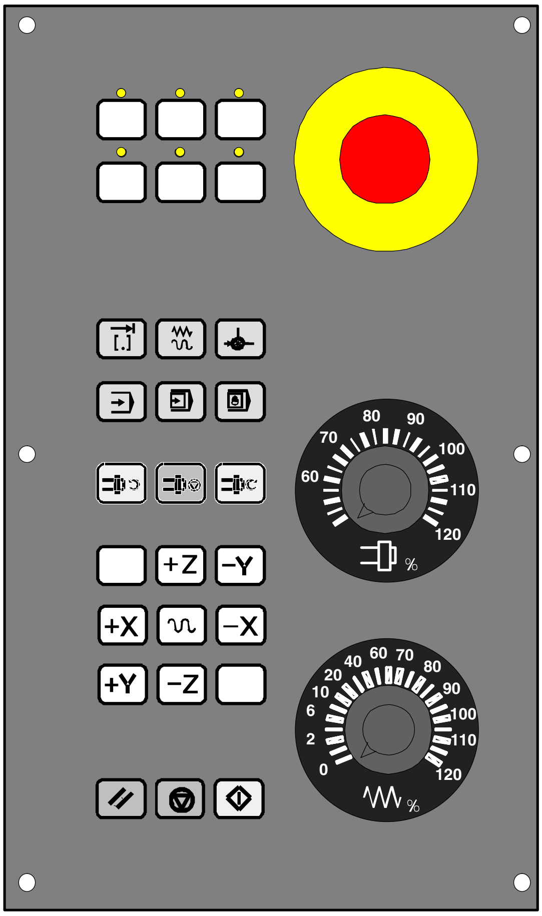
^	Recall key
>	ETC key
ALARM CANCEL	"Acknowledge alarm" key
1...n CHANNEL	without function
HELP	Info key
SHIFT	Shift key
CTRL	Control key
ALT	Alt key
␣	SPACE
← BACKSPACE	Delete key (Backspace)
DEL	Clear key
INSERT	Insert key
TAB	Tabulator
INPUT	ENTER / Input key

M POSITION	"Position" operating area key
PROGRAM	"Program" operating area key
OFFSET PARAM	"Parameter" operating area
PROGRAM MANAGER	"Program Manager" operating area key
SYSTEM ALARM	"Alarm/System" operating area (SHIFT key)
CUSTOM	
NEXT WINDOW	not assigned
PAGE UP PAGE DOWN	PageUp / PageDown keys
↑ ← → ↓	Cursor keys
SELECT	Selection key / toggle key
END	
A J W Z	Alphanumeric keys Double assignment on the Shift level

0 9

Numeric keys
Double assignment on the Shift level

External machine control panel



RESET

NC STOP

NC START

EMERGENCY STOP

Spindle override (option)

User-defined key with LED

User-defined key without LED

INCREMENT
Incremental dimension

JOG

REFERENCE POINT
Reference point

AUTOMATIC

SINGLE BLOCK

MANUAL DATA
Manual input

SPINDLE START LEFT
Spindle CCW rotation

SPINDLE STOP

SPINDLE START RIGHT
Spindle CW rotation

RAPID TRAVERSE OVERLAY
Rapid traverse override

+X –X X axis

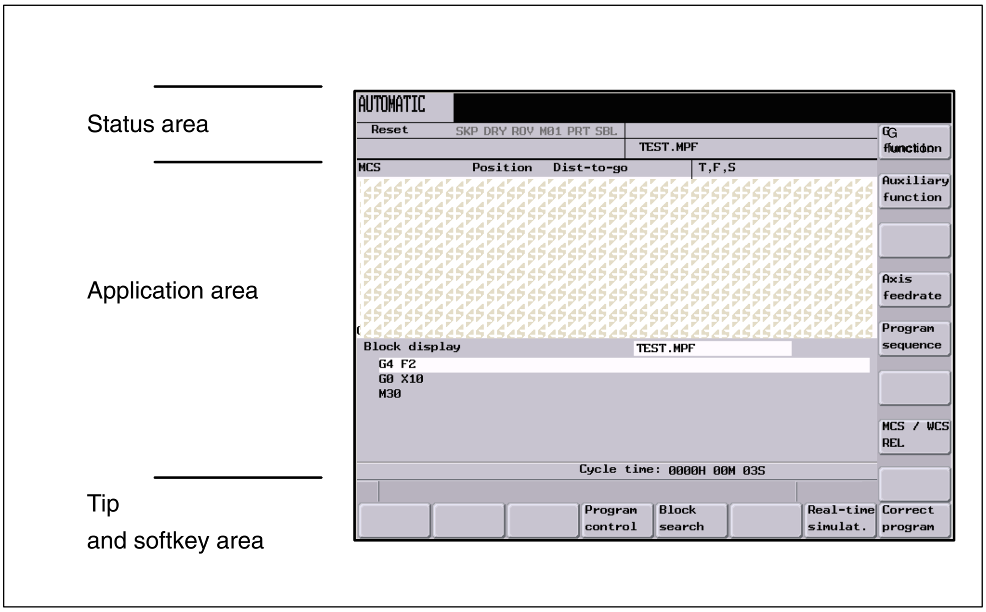
+Z –Z Z axis

Feedrate override

Notes

Introduction 1

1.1 Screen layout



```
G4 F2
G0 X10
M30
```

Fig. 1-1 Screen layout

The screen is divided into the following main areas:

- Status area
- Application area
- Tip and softkey area

Status area

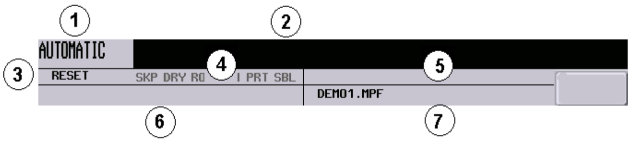


Fig. 1-2 Status area

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

Screen item	Display	Meaning
①	**Active operating area, active mode** Position JOG; 1 INC, 10 INC, 100 INC, 1000 INC, VAR INC (evaluation by increments in the JOG mode) JOG REF MDA AUTOMATIC OFFSET PARAM PROG MANAGER SYSTEM ALARM	
②	**Alarm and message line** In addition, the following is displayed: 1. Alarm number with alarm text, or 2. Message text	
③	**Program status**	
	RESET	Program canceled / default status
	RUN	Program is running
	STOP	Program stopped
④	**Program controls in the AUTOMATIC mode**	
⑤	**Path**	N: – NC internal “drive” D: – CF card
⑥	**NC messages**	
⑦	**Selected part program (main program)**	

Tip and softkey area

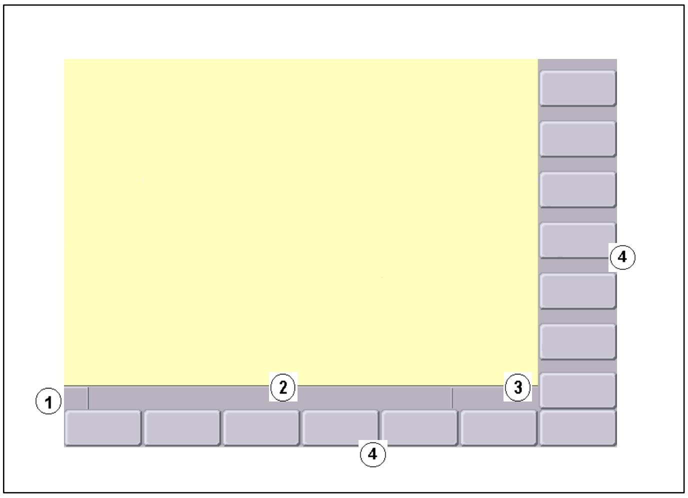


Fig. 1-3 Tip and softkey area

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

Screen item	Display	Meaning
①	⬆	**Recall symbol** Pressing the Recall key lets you return to the next higher menu level.
②		**Tip line** Displays tips for the operator
③	 ➡ Σ 	**HMI status information** **ETC** is possible (Pressing this key displays the horizontal softkey bar providing further functions.) Mixed notation active (uppercase/lowercase letters) Data transfer running Connection to the commissioning and diagnostic tool (e.g. Programming Tool 802) active
④		**Softkey bar** vertical and horizontal

Standard softkeys

Back «	Use this softkey to quit the screenform.

Abort ✗	Use this softkey to cancel the input; the window is closed.
Accept ✓	Selecting this softkey will complete your input and start the calculation.
ok ✓	Selecting this softkey will complete your input and accept the values you have entered.

1.2 Operating areas and protection levels

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

M POSITION	Position	Machine operation
OFFSET PARAM	Offset/Parameters	Input of offset values and setting data
PROGRAM	Program	Creation of part programs
PROGRAM MANAGER	Program Manager	Part program directory
SHIFT + SYSTEM ALARM	System	Diagnosis, start-up
SYSTEM ALARM	Alarm	Alarm and message lists

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

Protection levels

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

- Protection level 1 Expert password
- Protection level 2 Manufacturer password
- Protection level 3 User password

These cotnrol the individual access authorizations.

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

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

1.3 Accessibility options

1.3.1 Calculator

SHIFT =

The calculator function can be activated from any operating area using the key combination **<SHIFT +=>**.

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

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

<Input> starts the calculation. The result is displayed on the calculator.

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

Note

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

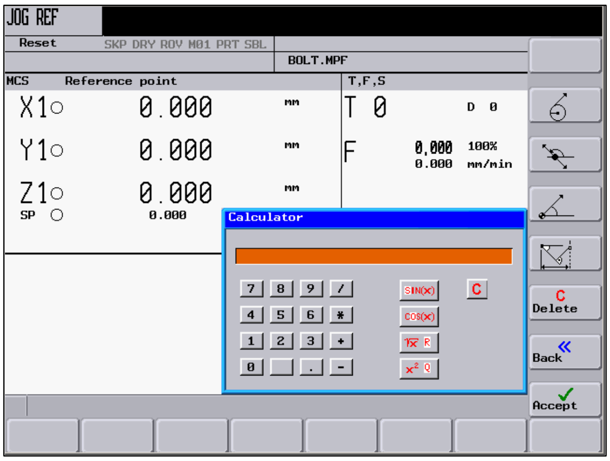

JOG REF
Reset SKP DRY ROV M01 PRT SBL
BOLT.MPF
MCS Reference point
X1 ○ 0.000 mm
Y1 ○ 0.000 mm
Z1 ○ 0.000 mm
SP ○ 0.000
T,F,S
T 0 D 0
F 0.000 100%
0.000 mm/min

Fig. 1-4 Calculator

Characters permitted for input

+, – Fundamental operations of arithmetic
*, /

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

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

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

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

( ) Bracket function (X+Y)*Z

Calculation examples

Task	Input –> Result
100 + (67*3)	100+67*3 –> 301
sin(45°)	45 S –> 0.707107
cos(45°)	45 O –> 0.707107
4^2	4 Q _–> 16
$\sqrt{}4$	4 R –> 2
(34+3*2)*10	(34+3*2)*10 –> 400

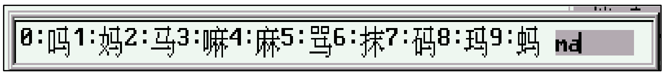
To calculate auxiliary points on a contour, the calculator offers the following functions:

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

1.3.2 Editing Chinese characters

This function is only available in the Chinese language version.

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



0:吗1:妈2:马3:嘛4:麻5:骂6:抹7:码8:玛9:蚂 ma

Fig. 1-5 Chinese editor

<ALT + S> Use this key combination to turn on / off the editor

1.3.3 Hotkeys

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

<CTRL+C>	Copy
<CTRL+B>	Select
<CTRL+X>	Cut
<CTRL+V>	Paste
<ALT+L>	Switch between uppercase/lowercase letters
<ALT+H>	Use this key combination or press the Info key to call the help system

1.3.4 Copying and pasting files

In the **Program Manager** area (Chapter 6) and with the **Start-up files** function (Section 7.1), files or even complete directories can be copied into another directory or to another drive using the softkey functions **<Copy>** and **<Paste>**. When doing so, the **<Copy>** function enters the references to the files or directories in a list which is subsequently executed by the **<Paste>** function. This function will perform the actual copying process.

The list is kept until a new copying process overwrites this list.

Special situation:
If the RS232 interface has been selected as the data target, **<Paste>** will be replaced by the **<Send>** softkey function. When reading in files **(**Receive **<softkey>)**, it is not necessary to specify a target, since the name of the target directory is not contained in the data flow.

1.4 The help system

A comprehensive online help system has been implemented in the control system. The following help topics are provided:

- Brief descriptions of all important operating functions
- Overview and brief description of the NC commands
- Explanation of the drive parameters
- Explanation of the drive alarms

Operating sequence

HELP

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

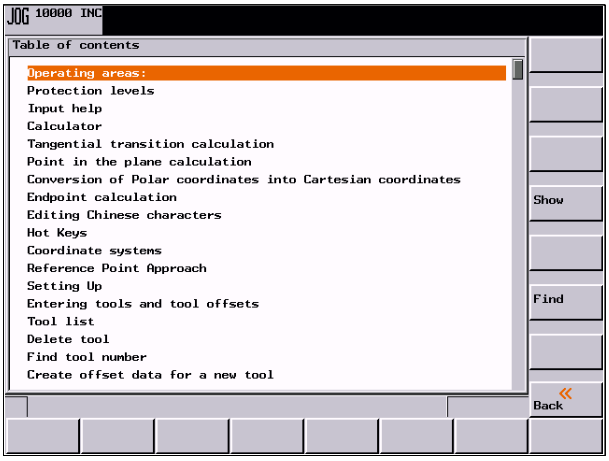


Table of contents

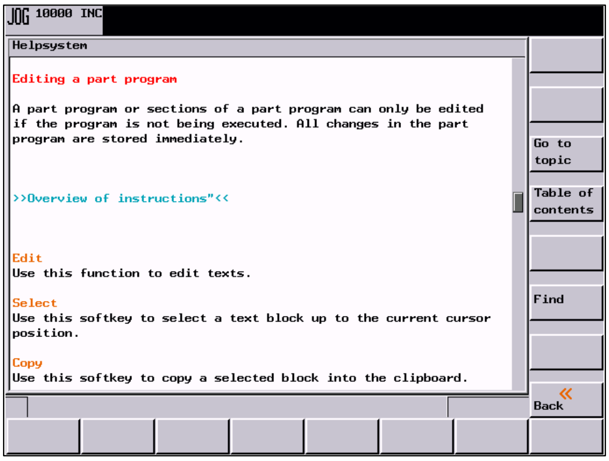
Operating areas:
Protection levels
Input help
Calculator
Tangential transition calculation
Point in the plane calculation
Conversion of Polar coordinates into Cartesian coordinates
Endpoint calculation
Editing Chinese characters
Hot Keys
Coordinate systems
Reference Point Approach
Setting Up
Entering tools and tool offsets
Tool list
Delete tool
Find tool number
Create offset data for a new tool

Fig. 1-6 Table of contents of the help system

Softkeys

Advertise ments

This function opens the selected topic.



Helpsystem

Editing a part program

A part program or sections of a part program can only be edited if the program is not being executed. All changes in the part program are stored immediately.

>>Overview of instructions"<<

Edit
Use this function to edit texts.

Select
Use this softkey to select a text block up to the current cursor position.

Copy
Use this softkey to copy a selected block into the clipboard.

Fig. 1-7 Description for a help topic

Go to line topic

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

Previous topic

Use this function to select a cross reference; in addition, the **<Previous topic>** softkey is displayed. Select this function to go back to the previous screen.

Find

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

Help in the “Program editor” area

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

1.5 Network operation

Note

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

Thanks to the integrated network adapter, the control system is network-capable. The following connections are possible:

- Peer-to-Peer: Direct connection between control system and PC using a cross-over cable
- Twisted Pair: Incorporation of the control system into an existing local network using a patch cable.

Screened network operation with encrypted data transfer is possible using an 802D specific transmission protocol. This protocol is used, e.g. for transmitting and executing part programs in conjunction with the RCS tool.

1.5.1 Configuring the network connection

Requirement

The control system is connected to the PC or the local network via the X5 interface.

Entering network parameters

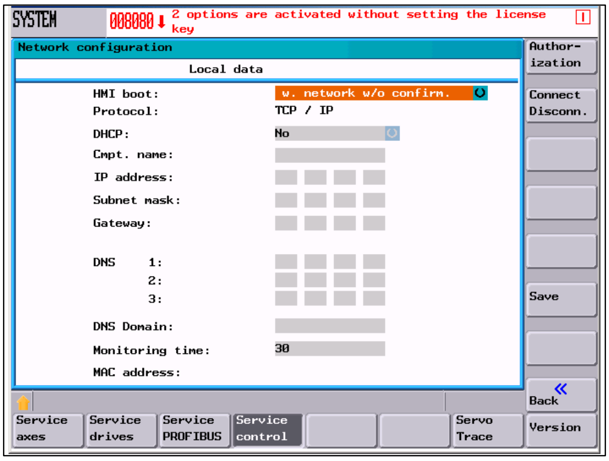
SHIFT SYSTEM ALARM

Service display | Service control

In the “System” operating area, select the **<Service display>** and **<Service control>** softkeys.

Service network

Select the **<Service network>** softkey to obtain access to the interactive screenform to input the network parameters.


SYSTEM 008080 2 options are activated without setting the license key

Network configuration

Local data

HMI boot: w. network w/o confirm.
Protocol: TCP / IP
DHCP: No
Cmpt. name:
IP address:
Subnet mask:
Gateway:
DNS 1:
2:
3:
DNS Domain:
Monitoring time: 30
MAC address:

Fig. 1-8

Table 1-3 Required network parameters

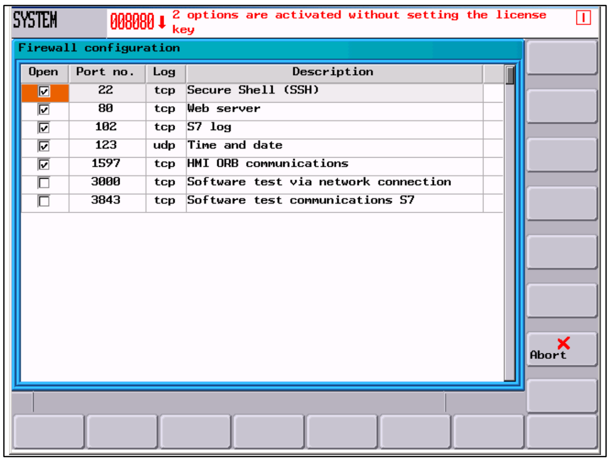
Parameters	Explanation
DHCP	DHCP protocol: The network requires a DHCP server that assigns the IP addresses dynamically. When **No** is selected, fixed network addresses are assigned. When **Yes** is selected, the network addresses are assigned dynamically. Input fields that are no longer needed are hidden.
Machine name (computer name)	Name of the control system in the network
IP address	Network address of the control system (e.g. 192.168.1.1)
Subnet mask	Network identification (e.g. 255.255.252.0)

Enabling the communication ports

Service Firewall

Use the "<Service firewall>" softkey to enable or disable communication ports.

To ensure maximum possible safety, all ports not needed should be closed.



SYSTEM 008080 2 options are activated without setting the license key

Firewall configuration

Open	Port no.	Log	Description
☑	22	tcp	Secure Shell (SSH)
☑	80	tcp	Web server
☑	102	tcp	S7 log
☑	123	udp	Time and date
☑	1597	tcp	HMI ORB communications
□	3000	tcp	Software test via network connection
□	3843	tcp	Software test communications S7

Fig. 1-9

The RCS network requires the ports 80 and 1597 for communication.

To change the port status, select the relevant port using the cursor. Pressing the INPUT key changes the port status.

Open ports are shown with the checkbox enabled.

1.5.2 User management

SHIFT SYSTEM ALARM

Service display Service control

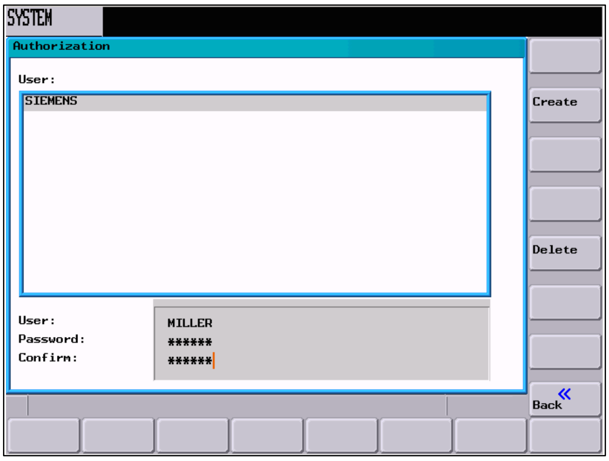
In the "System" operating area, select the **<Service display>** and **<Service control>** softkeys.

Service network
Authorization

Select the **<Service network > < Authorization>** softkeys to obtain access to the interactive screenform to input the network parameters.

The user accounts serve for saving personal settings of the users. To create a new account, type the user name and the log-in password in the input fields.

Use the **<Create>** softkey to insert a new user into the user management.



User:
SIEMENS

User: MILLER
Password: ******
Confirm: ******

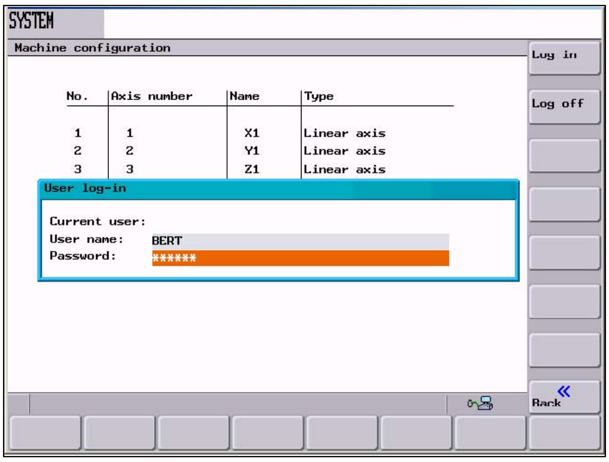
Fig. 1-10

Use the **<Delete>** softkey to delete the selected user from the user management.

1.5.3 User log-in – RCS log in

SHIFT
SYSTEM ALARM
RCS log–in

In the “System” operating area, select the **<RCS log-in>** softkey.
The interactive screen for the user log-in is opened.



No.	Axis number	Name	Type
1	1	X1	Linear axis
2	2	Y1	Linear axis
3	3	Z1	Linear axis

User log-in

Current user:
User name: BERT
Password: ******

Fig. 1-11 User log-in

Logon

Type user name and password into the appropriate input fields and select the **<Log in>** softkey to confirm.

After successful log-in, the user name is displayed in the **Current user** line.

Select the **<Back >** softkey to close the dialog box.

Note

This log-in simultaneously serves for user identification for remote connections.

Logoff

Press the **<Log-off >** softkey. This will log out the current user, all user-specific settings are saved, and any enables already granted are canceled.

1.5.4 Working with a network connection

The remote access (access to the control system from a PC or from a network) to the control system is disabled by default..

After log-in of a local user, the **RCS tool** are offered the following functions:

- Start-up functions
- Data transfer (transfer of part programs)
- Remote control for the control system

To grant access to a part of the file system, first share the relevant directories with other users.

Note:

If you share directories with other users, the authorized network nodes are granted access to the shared files in the control system. Depending on the sharting option, the user can modify or delete files.

1.5.5 Sharing directories

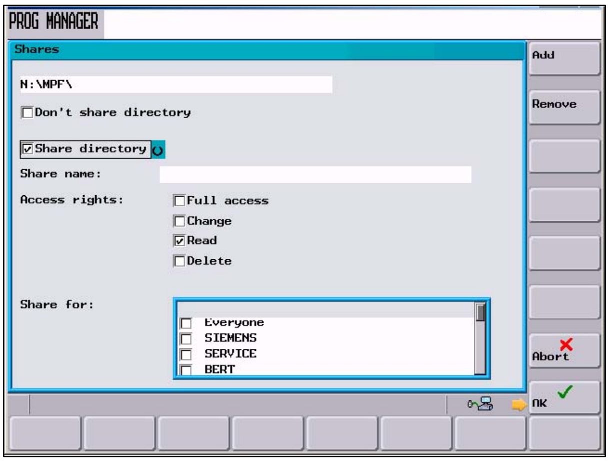
This function defines the rights for access of remote users to the file system of the control system.

PROGRAM MANAGER

Use the **Program Manager** to select the directory you want to share.

Enable

Use the **<Shares >** softkey to open the interactive screenform for sharing the selected directory.



PROG MANAGER

Shares

N:\MPF\

□ Don't share directory

☑ Share directory

Share name:

Access rights: □ Full access
□ Change
☑ Read
□ Delete

Share for: □ Everyone
□ SIEMENS
□ SERVICE
□ BERT

Fig. 1-12 Sharing status of the directory

- Select the sharing status for the selected directory:

– **Do not share this directory**	Directory will not be shared
– **Share this directory**	The directory will be shared and a share name must be entered.

- Type an identifier into the **Share name** field through which authorized users can access the files in the directory.
- Use the **<Add>** softkey to obtain access to the user list. Select the user. By selecting the **<Add>** softkey, the relevant information is entered in the **Shared for** field.
- Define the user rights (**Credentials**).

– **Full access**	User has full access
– **Change**	User may modify files
– **Read**	User may read files
– **Delete**	User may delete files

Select the **<OK >** softkey to set the specified properties. As in Windows, shared directories are marked with a “hand”.

1.5.6 Connecting / disconnecting network drives

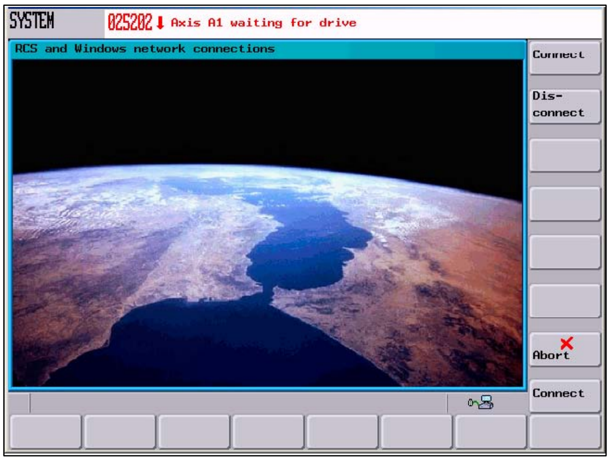
SHIFT SYSTEM ALARM

Service display Service control

In the “System” operating area, select the **<Service display>** and **<Service control>** softkeys.

Service network Connect Disconnect

Select **<Service network > <Connect Disconnect >** to obtain access to the network drive configuration area.


SYSTEM

025202 Axis A1 waiting for drive

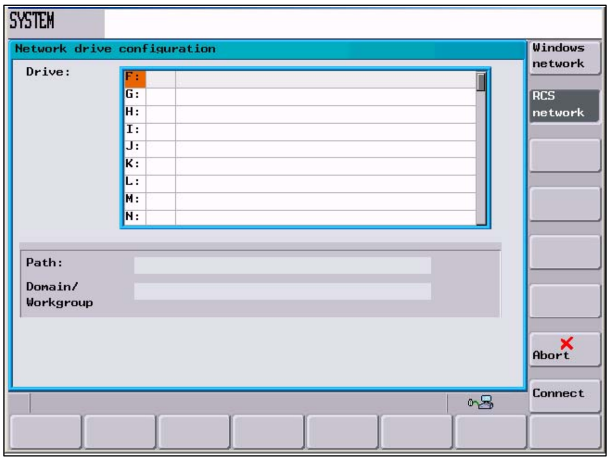
RCS and Windows network connections

Fig. 1-13

Network drives

Connect

The **<Connect >** function assigns a network drive a local drive letter.


SYSTEM

Network drive configuration

Drive:

F:
G:
H:
I:
J:
K:
L:
M:
N:

Path:

Domain/ Workgroup

Fig. 1-14 Network drive configuration

Place the cursor on an unused drive designator and use the TAB key to switch to the **Path** input field. Specify the IP address of the server and the sharing name.

Example: \\192.4.5.23\TEST\

The **<Connect>** softkey assigns a drive designator to the server connection.

Disconnecting network drives

Dis-connect

Use the **<Disconnect>** function to cancel an existing network connection.
To do so, position the cursor on the appropriate drive letter and select the **<Disconnect>** softkey. The appropriate drive is disconnected from the network.

1.6 RCS802 Tool

With the RCS tool (Remote Control System), you are provided with an Explorer tool for your PC/PG to assist you in your daily work with SINUMERIK 802D sl.

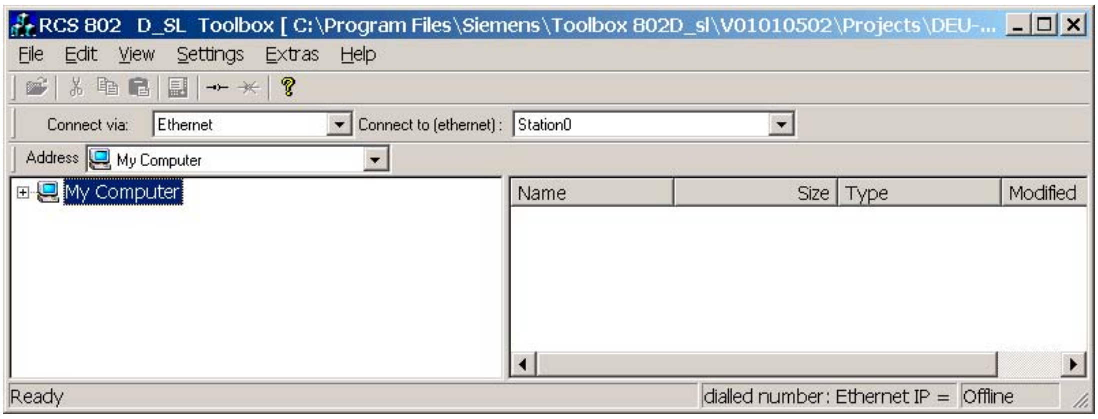
The connection between the control system and the PC/PG can be provided either via an RS232 cable or a network (option).

Attention

The full functionality of the RCS tool is only provided after loading of the RCS 802 license key.

This license key can be used to establish the connection to the control system via a local network (SINUMERIK 802D sl pro only); thus, it is also possible, for example, to use the remote control function.

Without license key, it is only possible to share local directories (on the PC/PG) for access by the control system.



RCS 802 D_SL Toolbox [C:\Program Files\Siemens\Toolbox 802D_sl\V01010502\Projects\DEU-...
File Edit View Settings Extras Help
My Computer
Name
Size
Type
Modified

Connect via: Ethernet

Connect to (ethernet): Station0

Address: My Computer

Ready | dialled number: Ethernet IP = | Offline

Fig. 1-15 Explorer window of the RCS tool

After starting, you are in the offline mode. This means that you can manage files on your PC only. In the online mode, the **Control 802D** directory additionally exists for data exchange with the control system. In addition, a remote control function is provided for process monitoring.

Note

The RCS tool provides a detailed online help function. For further details e.g. establishing a connection, project management etc., please refer to this help menu.

1.7 Coordinate systems

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

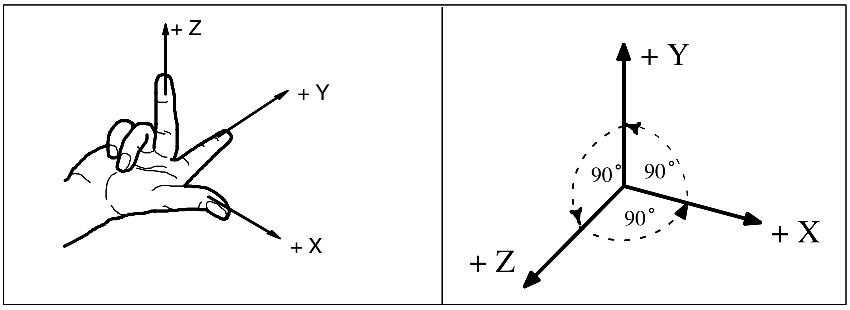


Fig. 1-16 Definition of the directions of the axes one to another; right-angled coordinate system

Machine coordinate system (MCS)

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

The axis directions follow the “Right-hand three-finger rule”. Seen from in front of the machine, the middle finger of the right hand points in the opposite direction to the infeed direction of the main spindle.

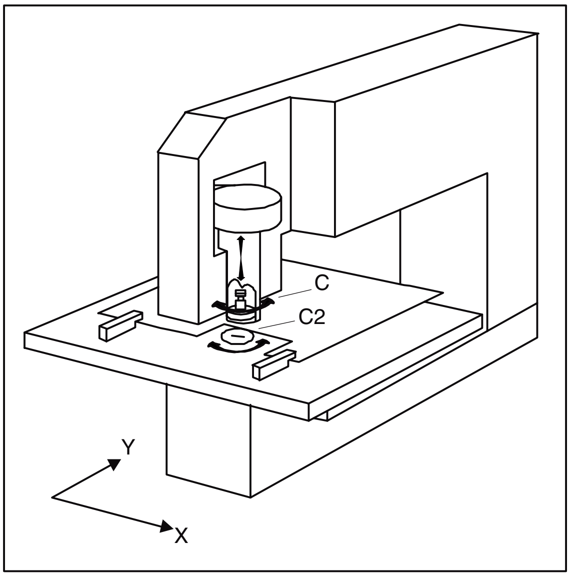


Fig. 1-17 Machine coordinates/machine axes

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

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

Workpiece coordinate system (WCS)

The coordinate system described in the beginning (see Fig. 1-16) is also used to describe the geometry of a workpiece in the workpiece program.
The **workpiece zero** can be freely selected by the programmer. The programmer need not to know the real motion relations on the machine, i.e. he need not to know whether the workpiece or the tool moves. Furthermore, it can be different from axis to axis. The directions are always defined such if the workpiece would be resting and the tool would move.

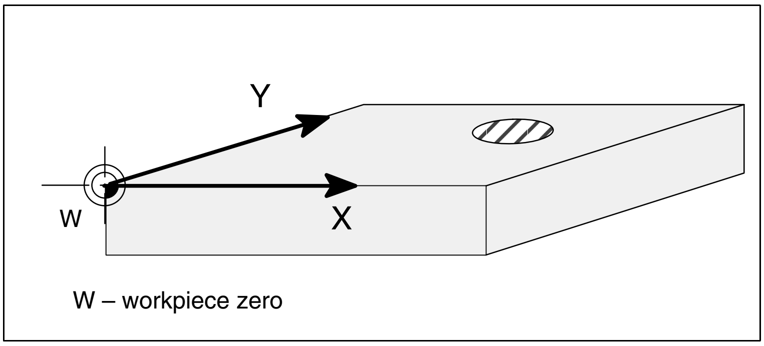


Fig. 1-18 Workpiece coordinate system

Relative coordinate system

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

Clamping the workpiece

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

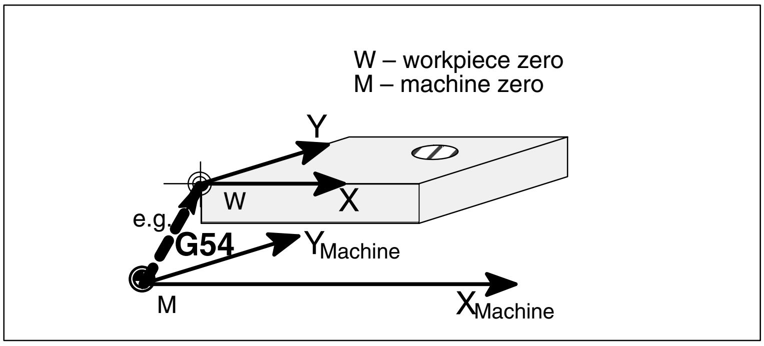


Fig. 1-19 Workpiece on the machine

Current workpiece coordinate system

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

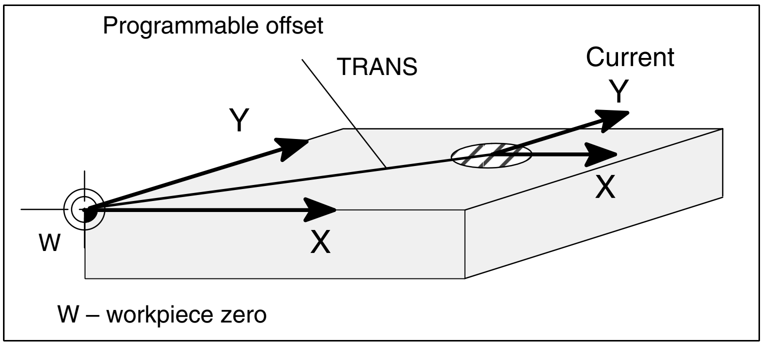


Fig. 1-20 Coordinates on the workpiece; current workpiece coordinate system

Turning On and Reference Point Approach 2

Note

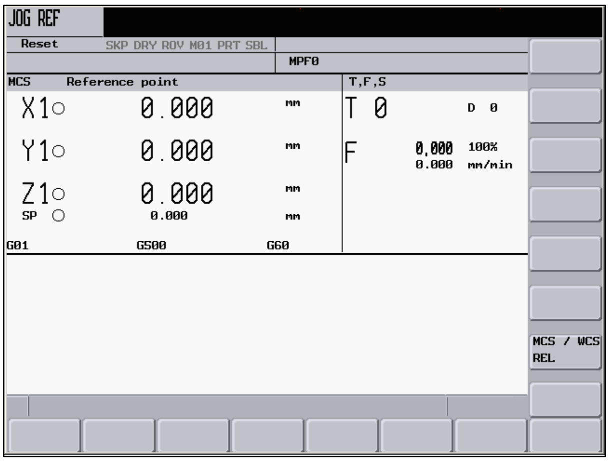
When you turn on the SINUMERIK 802D and the machine, please also observe the Machine Documentation, since turning on and reference point approach are machine-dependent functions.

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

Operating sequence

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

The *Reference point approach* window is active.



JOG REF

Reset

SKP DRY ROV M01 PRT SBL

MPF0

MCS Reference point

X1 ○ 0.000 mm

Y1 ○ 0.000 mm

Z1 ○ 0.000 mm

SP ○ 0.000 mm

G01 G500 G60

T,F,S

T 0 D 0

F 0.000 100%

0.000 mm/min

Fig. 2-1 The “Jog-Ref” start screen

The “Reference point approach” window (Fig. 2-1) displays whether or not the axes have been referenced (approached to their reference points).

○ Axis must be referenced

◐ Axis has reached its reference point

+X ... –Z

Press a direction key.

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

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

Note

“Reference point approach” is only possible in the Jog mode in the JOG REF window.

Setup 3

3.1 Punching and nibbling tools

Functionality

Which punching/nibbling tools are used depends on the particular case of application. The tool geometry results from the following standard geometry forms:

- Round
- Triangle
- Rectangle, square
- Elongated hole
- Double “D”
- Single “D”
- Long “D”
- Hexagon or octagon

Tools with several punches within the envelope are called multi-tools.

3.1.1 Tool list

Operating sequences

OFFSET PARAM

Tool list

You are in the **Offset parameters** operating area.

Use **<Tool list>** to open the list of tools created. Use the cursor keys and the **<Page Up> / <Page Down>**keys to navigate in this list.

Enter the offsets by positioning the

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

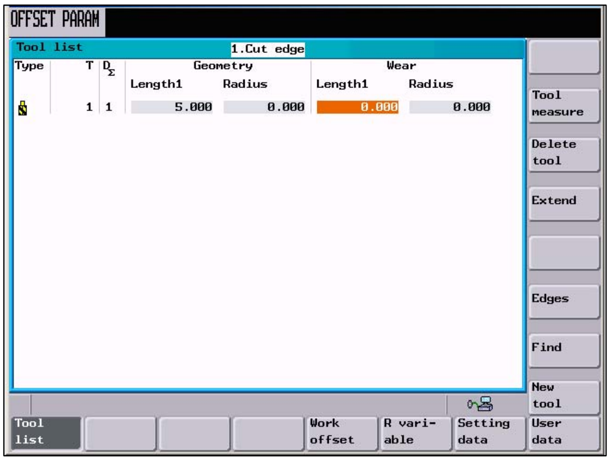
INPUT

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

Softkeys

Delete tool	Use this softkey to delete all tool offset data for all cutting edges of the tool.
Advanced	All parameters of a tool are displayed.
Find	Find tool number Type the term you are looking for in the input line and use the **<OK>** softkey to start the search. If the tool you are looking for exists, the cursor is positioned on the appropriate line.
New tool	Use this softkey to create tool compensation data for a new tool.

Parameters


OFFSET PARAM

Tool list 1.Cut edge

Type	T	D Σ	Geometry		Wear	
			Length1	Radius	Length1	Radius
	1	1	5.000	0.000	0.000	0.000

Fig. 3-1 Tool list

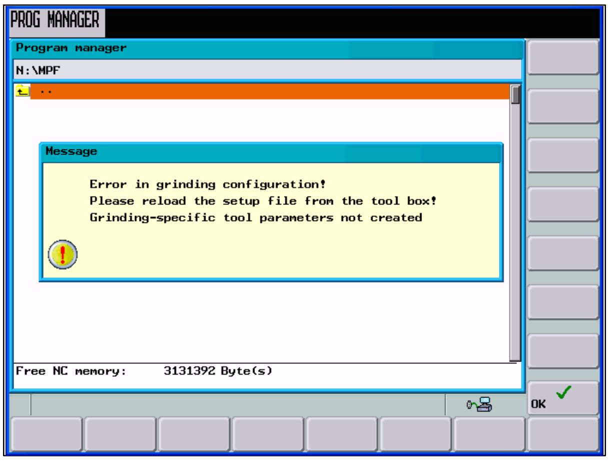
Table 3-1 Parameters of the tool list

Field	Description
✥	Bitmap for identification of a multi-tool
T	Tool number

Table 3-1 Parameters of the tool list, continued

Field	Description
Type	Describes the type of the tool geometry: 1 Round tool 2 Rectangular tool 3 Triangular tool 4 Double “D” 5 Single “D” 6 Long “D” 7 Elongated hole 8 Hexagon tool 9 Octagon tool 99 Free geometry 104 Multi-tool, 4 punches 105 Multi-tool, 5 punches 106 Multi-tool, 6 punches 108 Multi-tool, 8 punches 109 Multi-tool, 9 punches 110 Multi-tool, 10 punches 111 Multi-tool, 11 punches 112 Multi-tool, 12 punches
Dimension 1	Radius or length of the cutting edge
Dimension 2	Width of the cutting edge
Sheet thickness	Maximum sheet thickness which can be machined

If the technology setup has not been read in by the toolbox or the wrong technology setup has been read in, it might be that the supplementary tool parameters for the **Nibbling** configuration are not yet activated. This is displayed to the operator.

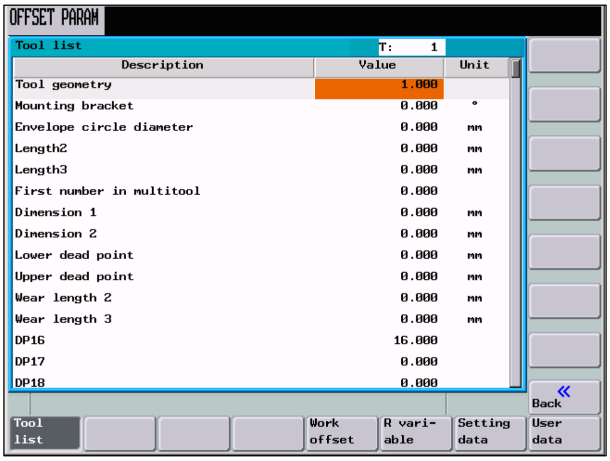


Error in grinding configuration!
Please reload the setup file from the tool box!
Grinding-specific tool parameters not created

Free NC memory: 3131392 Byte(s)

Fig. 3-2 Message in case of a configuration error

To open the complete parameter list, use <**Advanced**>.



OFFSET PARAM

Tool list T: 1

Description	Value	Unit
Tool geometry	1.000	
Mounting bracket	0.000	°
Envelope circle diameter	0.000	mm
Length2	0.000	mm
Length3	0.000	mm
First number in multitool	0.000	
Dimension 1	0.000	mm
Dimension 2	0.000	mm
Lower dead point	0.000	mm
Upper dead point	0.000	mm
Wear length 2	0.000	mm
Wear length 3	0.000	mm
DP16	16.000	
DP17	0.000	
DP18	0.000	

Fig. 3-3 Parameter list for a tool (Advanced)

Caution

The specified parameter list uses the parameters for the tool compensation. Therefore, the **G41 and G42** commands must **not** be used in the NC program.

Table 3-2 Parameter list for nibbling

Parameters	Field	Description
DP4	Length 2	Geometry of the Y axis
DP5	Length 3	Geometry of the X axis
DP6	Tool geometry	Describes the type of the tool geometry
DP7	First multi-tool number	First tool number of the punch on a multi-tool
DP8	Y – Length of the cutting edge	Length of the A axis cutting edge
DP9	X – Length of the cutting edge	Length of the X axis cutting edge
DP10	Bottom dead center	Bottom dead center of the tool
DP11	Top dead center	Top dead center of the tool
DP12	Mounting bracket	Angle at which the tool is mounted
DP13	Wear length 2	Y axis wear (fine compensation)
DP14	Wear length 3	X axis wear (fine compensation)
DP16	DP16	Free
DP17	DP17	Free
DP18	DP18	Free
DP19	DP19	Free
DP20	DP20	Free
DP21	Stroke counter	Stroke counter
DP22	Base length 2	Base length 2 (adapter), Y axis
DP23	Base length 3	Base length 3 (adapter), X axis
DP24	Envelope diameter	Diameter of the tool in the toolholder

Table 3-2 Parameter list for nibbling, continued

Parameters	Field	Description
DP25	Sheet thickness	Current sheet thickness
DPC1	Axis of the driven tool	Tool orientation can be changed by way of rotary axis
DPC2	Location	Tool location
DPC3	Maximum sheet thickness	Maximum sheet thickness
DPC4	No. of the multi-tool	Number of the toolholder
DPC5	DPC5	Free
DPC6	DPC6	Free
DPC7	DPC7	Free
DPC8	DPC8	Free
DPC9	DPC9	Free
DPC10	DPC10	Free

Important

The meanings of the **DP2**, **DP6**, **DP12** and **DP24** parameters is defined in conjunction with part program simulation as follows:

- DP2 Reserved for part program simulation
- DP6 Tool geometry
- DP12 Angle of tool orientation
- DP24 Envelope diameter

and cannot be overwritten.

Interrelation between geometry data and cutting edge parameters

The geometry data required for the simulation of punching/nibbling tools can be mapped to the cutting edge parameters as follows:

Table 3-3 Interrelation ’geometry data – cutting edge parameters’

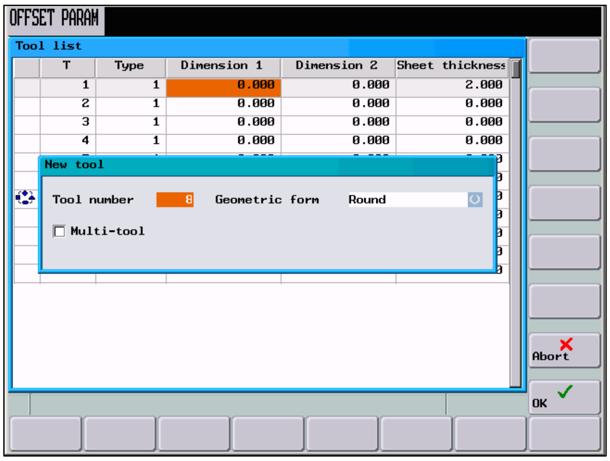
Geometry	Tool geometry	Cutting edge parameters
Round, hexagon or octagon, triangle, double “D”, single “D”, long “D”, rectangle, square, elongated hole	Envelope diameter Orientation angle	DP24 DP12
Rectangle, square	Y Length of the tool edge X Length of the tool edge	DP8 DP9

3.1.2 Create new tool

Operating sequence

New tool

The interactive screenform for creating a new tool is opened.



Tool list

	T	Type	Dimension 1	Dimension 2	Sheet thickness
	1	1	0.000	0.000	2.000
	2	1	0.000	0.000	0.000
	3	1	0.000	0.000	0.000
	4	1	0.000	0.000	0.000

New tool

Tool number 8 Geometric form Round

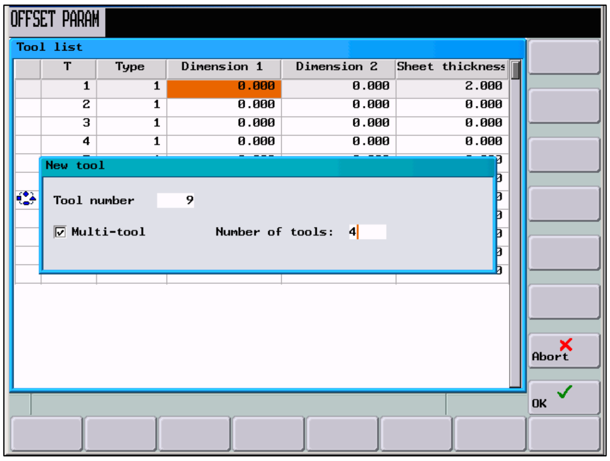
□ Multi-tool

Fig. 3-4 New tool

Type the tool number and the tool form in the interactive screenform.

OK ✓

Select **<OK>** to confirm your input. The tool is displayed in the tool list.



Tool list

	T	Type	Dimension 1	Dimension 2	Sheet thickness
	1	1	0.000	0.000	2.000
	2	1	0.000	0.000	0.000
	3	1	0.000	0.000	0.000
	4	1	0.000	0.000	0.000

New tool

Tool number 9

☑ Multi-tool Number of tools: 4

Fig. 3-5 New tool: Multi-tool

When creating a multi-tool, check the "**Multi-tool**" checkbox and type the number of punches in the displayed input field. Use **<OK>** to create a tool for the punch holder and to specify the number of subtools.

Note

The following tool range can be used: Tool numbers from 1 to 99.
Multi-tools are created using the formula *Tool number by 100*.

3.2 Entering / modifying a work offset

Functionality

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

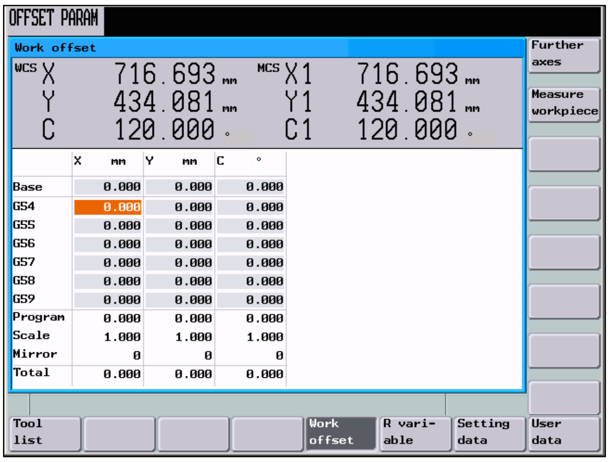
Operating sequences

OFFSET PARAM

Work offset

You are in the **“Offset parameters”** operating area.

Selecting **<Work offset>** opens an overview of the settable work offsets. The screenform additionally contains the values of the programmed work offset, of the active scaling factors, the “Mirroring active” status display and the total of all active work offsets.



	X mm	Y mm	C °
Base	0.000	0.000	0.000
G54	0.000	0.000	0.000
G55	0.000	0.000	0.000
G56	0.000	0.000	0.000
G57	0.000	0.000	0.000
G58	0.000	0.000	0.000
G59	0.000	0.000	0.000
Program	0.000	0.000	0.000
Scale	1.000	1.000	1.000
Mirror	0	0	0
Total	0.000	0.000	0.000

Fig. 3-6 The “Work offset” window

← ↑ → ↓

Position the cursor bar on the input field to be changed

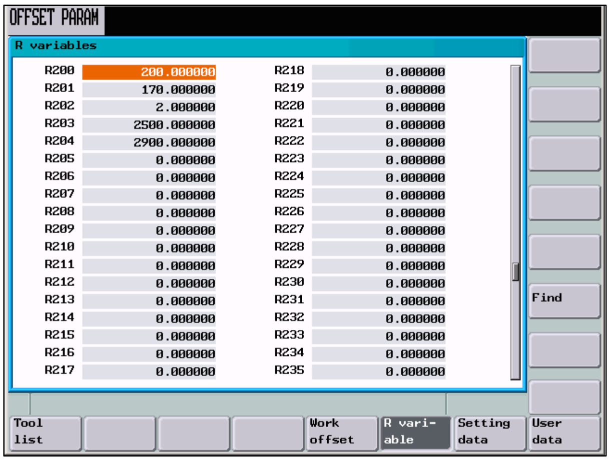
0 9

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

3.3 Arithmetic parameters R

Functionality

The **R parameters** start screen displays all R parameters existing in the control system (see also Section 8.7 “R parameters”).
These can be changed if necessary.



OFFSET PARAM

R variables

R200	200.000000	R218	0.000000
R201	170.000000	R219	0.000000
R202	2.000000	R220	0.000000
R203	2500.000000	R221	0.000000
R204	2900.000000	R222	0.000000
R205	0.000000	R223	0.000000
R206	0.000000	R224	0.000000
R207	0.000000	R225	0.000000
R208	0.000000	R226	0.000000
R209	0.000000	R227	0.000000
R210	0.000000	R228	0.000000
R211	0.000000	R229	0.000000
R212	0.000000	R230	0.000000
R213	0.000000	R231	0.000000
R214	0.000000	R232	0.000000
R215	0.000000	R233	0.000000
R216	0.000000	R234	0.000000
R217	0.000000	R235	0.000000

Fig. 3-7 The “R parameters” window

Operating sequence

OFFSET PARAM

R para-meters

INPUT

You are in the **Offset/parameters** operating area.
Use the **<R parameters>** softkey to obtain access to the input area. Position the cursor bar on the input field to be modified and enter the value.

Either press the **<Input>** key or move the cursor to confirm. The values are accepted.

3.4 Programming the setting data

Functionality

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

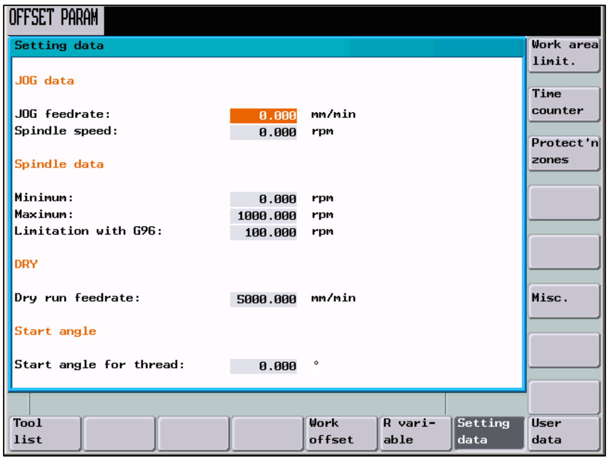
Operating sequences

OFFSET PARAM

Setting data

You are in the **"Offset parameters"** operating area.

Press the **<Setting data>** softkey. The "Setting data" start screen is opened. It provides further softkey functions which can be used to set various options in the control system.



JOG data

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

Spindle data

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

DRY

Dry run feedrate:	5000.000	mm/min

Start angle

Start angle for thread:	0.000	°

Fig. 3-8 The *Setting data* start screen

Jog feedrate

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

Dry run feed (DRY)

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

Note: No further setting data are relevant.

← ↑ → ↓

0 9

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

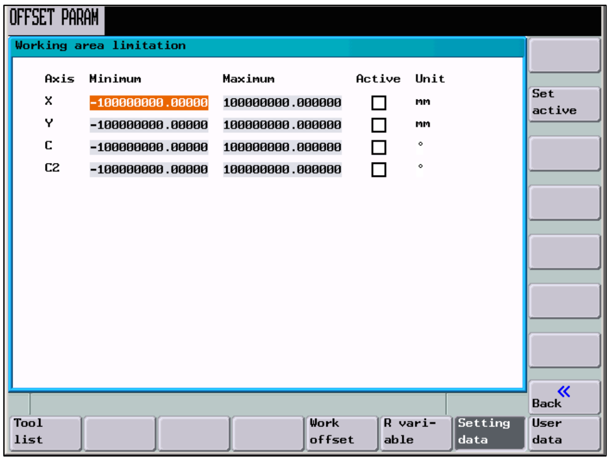
INPUT

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

Softkeys

Set active

The working area limitation is active with geometry and additional axes. Enter the values for the work area limitation. Selecting the **<Set active>** softkey enables / disables the values for the axis highlighted by the cursor.



OFFSET PARAM

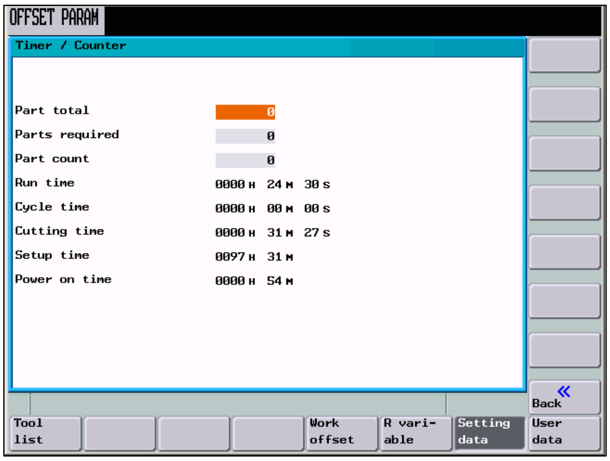
Working area limitation

Axis	Minimum	Maximum	Active	Unit
X	-100000000.00000	100000000.000000	□	mm
Y	-100000000.00000	100000000.000000	□	mm
C	-100000000.00000	100000000.000000	□	°
C2	-100000000.00000	100000000.000000	□	°

Fig. 3-9

Times Counters

Timers Counters



OFFSET PARAM

Timer / Counter

Part total	0
Parts required	0
Part count	0
Run time	0000 H 24 M 30 s
Cycle time	0000 H 00 M 00 s
Cutting time	0000 H 31 M 27 s
Setup time	0097 H 31 M
Power on time	0000 H 54 M

Fig. 3-10

Explanation:

- Total parts: Number of workpieces produced in total (total actual value)
- Parts requested: Number of workpieces required (number of workpieces setpoint)
- Number of parts: This counter registers the number of all workpieces produced since the starting time.
- Total runtime: Total execution time of NC programs in the Automatic mode (in seconds): In the AUTOMATIC mode, the execution times of all programs are summed between NC start and program end / reset. The timer is zeroed with each power-up of the control system. Time of execution of the selected NC program (in seconds)

- Time of program execution: Tool action time (in seconds)
  The runtime between NC START and end of program / RESET is measured in the selected NC program. The timer is reset with starting a new NC program.
- Feedrate runtime
  The runtime of the path axes is measured in all NC programs between NC START and end of program / RESET without rapid traverse active and with the tool active. The measurement is interrupted when a dwell time is active.

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

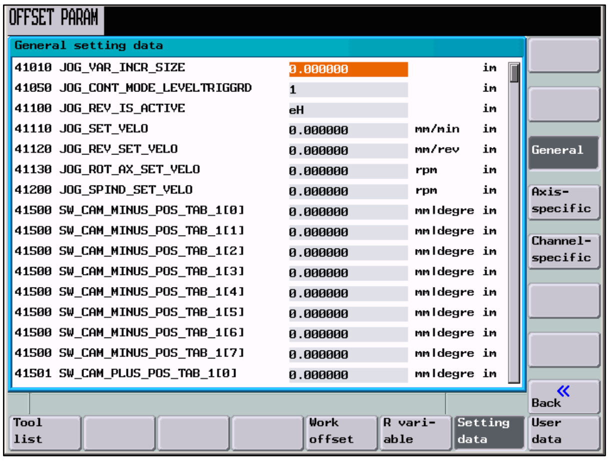
Protection zones

This function can be used to define protective areas for max. four clamps (clamp protection).

Miscellaneous

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

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



General setting data

```
41010 JOG_VAR_INCR_SIZE                 0.000000             im
41050 JOG_CONT_MODE_LEVELTRIGGRD        1                    im
41100 JOG_REV_IS_ACTIVE                 eH                   im
41110 JOG_SET_VELO                      0.000000   mm/min    im
41120 JOG_REV_SET_VELO                  0.000000   mm/rev    im
41130 JOG_ROT_AX_SET_VELO               0.000000   rpm       im
41200 JOG_SPIND_SET_VELO                0.000000   rpm       im
41500 SW_CAM_MINUS_POS_TAB_1[0]         0.000000   mm|degre  im
41500 SW_CAM_MINUS_POS_TAB_1[1]         0.000000   mm|degre  im
41500 SW_CAM_MINUS_POS_TAB_1[2]         0.000000   mm|degre  im
41500 SW_CAM_MINUS_POS_TAB_1[3]         0.000000   mm|degre  im
41500 SW_CAM_MINUS_POS_TAB_1[4]         0.000000   mm|degre  im
41500 SW_CAM_MINUS_POS_TAB_1[5]         0.000000   mm|degre  im
41500 SW_CAM_MINUS_POS_TAB_1[6]         0.000000   mm|degre  im
41500 SW_CAM_MINUS_POS_TAB_1[7]         0.000000   mm|degre  im
41501 SW_CAM_PLUS_POS_TAB_1[0]          0.000000   mm|degre  im
```

Fig. 3-11

3.5 Clamp protection

OFFSET PARAM

Setting data

You are in the **Offset parameters** operating area and have selected the **<Setting data>** softkey.

Protection zones

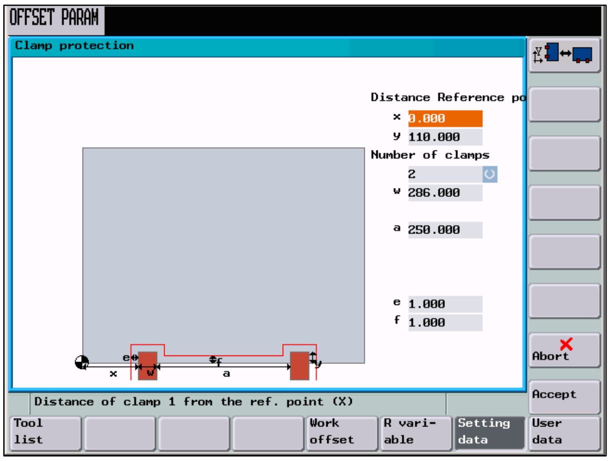
Use this interactive screenform to define protection zones for max. four clamps.

Type the appropriate values in the interactive screenform and press **<Enter>** to confirm. You will be prompted to press NC START.

If you quit the screenform and no NC error message is displayed, the protection zones are defined.

Caution

Subsequently, activate the defined protection zones according to the number of clamps and holder in the NC program using the CPROT command.



OFFSET PARAM

Clamp protection

Distance Reference po

x 0.000

y 110.000

Number of clamps

2

w 286.000

a 250.000

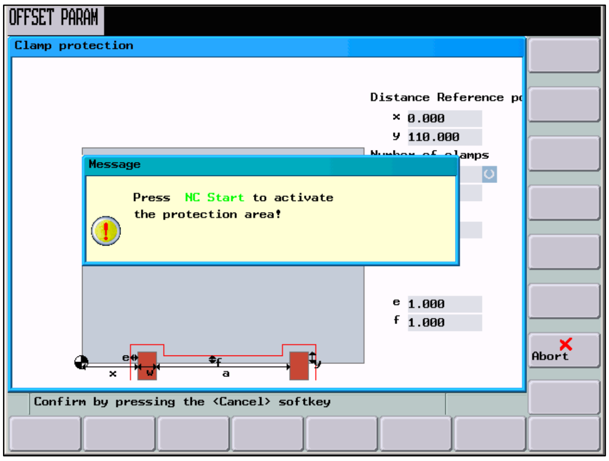
e 1.000

f 1.000

Distance of clamp 1 from the ref. point (X)

Fig. 3-12 Clamp protection

Use this softkey function to switch the arrangement of the claws from ’parallel to the X axis’ to ’parallel to the Y axis’.



OFFSET PARAM

Clamp protection

Distance Reference po

x 0.000

y 110.000

Number of clamps

Message

Press NC Start to activate
the protection area!

e 1.000

f 1.000

Confirm by pressing the <Cancel> softkey

Fig. 3-13 Defining protection zones

Use the **<Abort>** softkey to cancel.

Depending on the number of clamps, the following NC machine data must be changed: **MD 28200, MD 28210, MD 28212, MD 18190, MD 18890**.

- Number of protection areas used

  2 clamps: 3

  3 clamps: 4

  4 clamps: 6

- Number of elements used

  2 clamps: 12 elements

  3 clamps: 16 elements

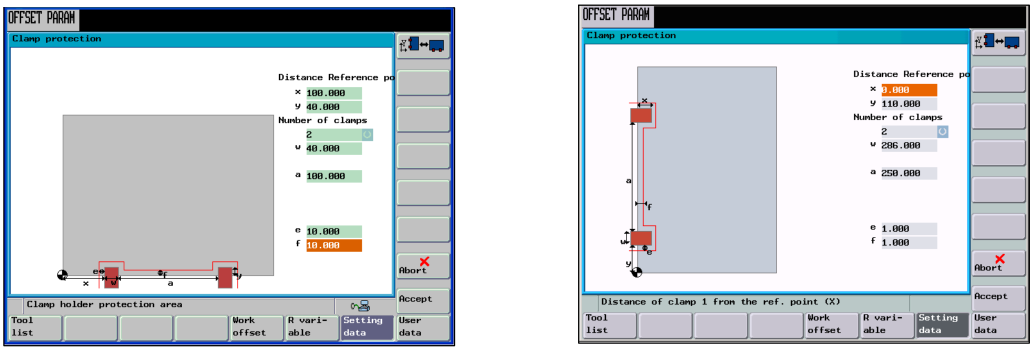
  4 clamps: 24 elements

3.5.1 Protection zone for two clamps

The parameters refer to a defined reference point (workpiece zero).

Table 3-4 Parameters for two clamps

Parameter	Meaning
x	X position of the clamp
y	Y position of the clamp
w	Clamp width
a	Clearance between clamps 1 and 2
e	Clamp safety clearance
f	Clamp holder safety clearance



OFFSET PARAM – Clamp protection

Distance Reference po
x 100.000
y 40.000
Number of clamps
2
w 40.000
a 100.000
e 10.000
f 10.000

Clamp holder protection area

OFFSET PARAM – Clamp protection

Distance Reference po
x 0.000
y 110.000
Number of clamps
2
w 286.000
a 250.000
e 1.000
f 1.000

Distance of clamp 1 from the ref. point (X)

Fig. 3-14 Interactive screenform for two clamps

The protection zones are assigned as follows:

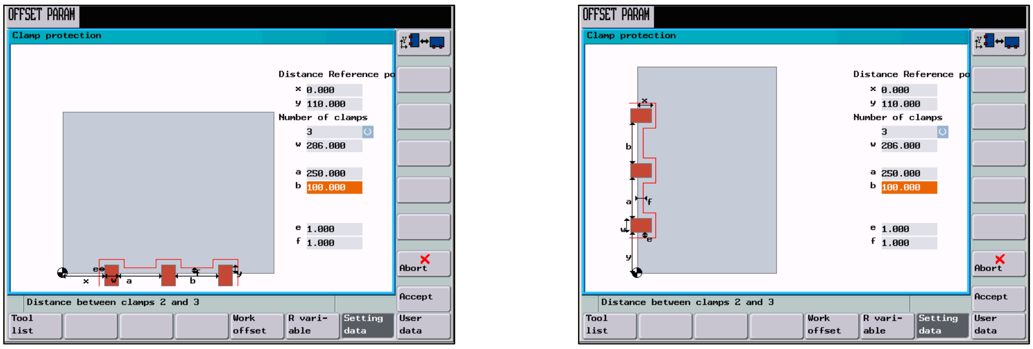
Protection zone number	Area
1	Clamp 1
2	Clamp 2
3	Clamp holder

3.5.2 Protection zones for three clamps

The parameters refer to a defined reference point (workpiece zero).

Table 3-5 Parameters for three clamps

Parameter	Meaning
x	X position of the clamp
y	Y position of the clamp
w	Clamp width
a	Clearance between clamps 1 and 2
b	Clearance between clamps 2 and 3
e	Clamp safety clearance
f	Clamp holder safety clearance



OFFSET PARAM – Clamp protection

Distance Reference po
x 0.000
y 110.000
Number of clamps
3
w 286.000
a 250.000
b 100.000
e 1.000
f 1.000

Distance between clamps 2 and 3

OFFSET PARAM – Clamp protection

Distance Reference po
x 0.000
y 110.000
Number of clamps
3
w 286.000
a 250.000
b 100.000
e 1.000
f 1.000

Distance between clamps 2 and 3

Fig. 3-15 Interactive screenform for three clamps

The protection zones are assigned as follows:

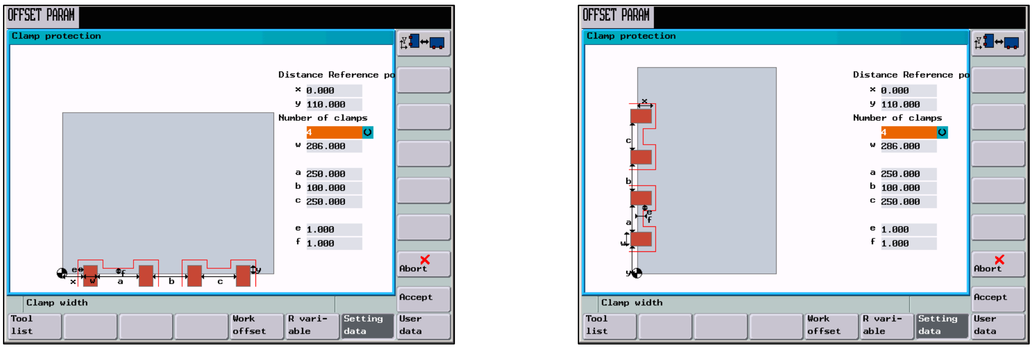
Protection zone number	Area
1	Clamp 1
2	Clamp 2
3	Clamp 3
4	Clamp holder

3.5.3 Protection zone for four clamps

The parameters refer to a defined reference point (workpiece zero).

Table 3-6 Parameters for four clamps

Parameter	Meaning
x	X position of the clamp
y	Y position of the clamp
w	Clamp width
a	Clearance between clamps 1 and 2
b	Clearance between clamps 2 and 3
c	Clearance between clamps 3 and 4
e	Clamp safety clearance
f	Clamp holder safety clearance


OFFSET PARAM

Clamp protection

Distance Reference po

x 0.000
y 110.000

Number of clamps

4

w 286.000

a 250.000
b 100.000
c 250.000

e 1.000
f 1.000

Clamp width

OFFSET PARAM

Clamp protection

Distance Reference po

x 0.000
y 110.000

Number of clamps

4

w 286.000

a 250.000
b 100.000
c 250.000

e 1.000
f 1.000

Clamp width

Fig. 3-16 Interactive screenform for four clamps

The protection zones are assigned as follows:

Protection zone number	Area
1	Clamp 1
2	Clamp 2
3	Clamp holder 1
4	Clamp 3

Protection zone number	Area
5	Clamp 4
6	Clamp holder 2

Manually Controlled Mode 4

Preliminary remark

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

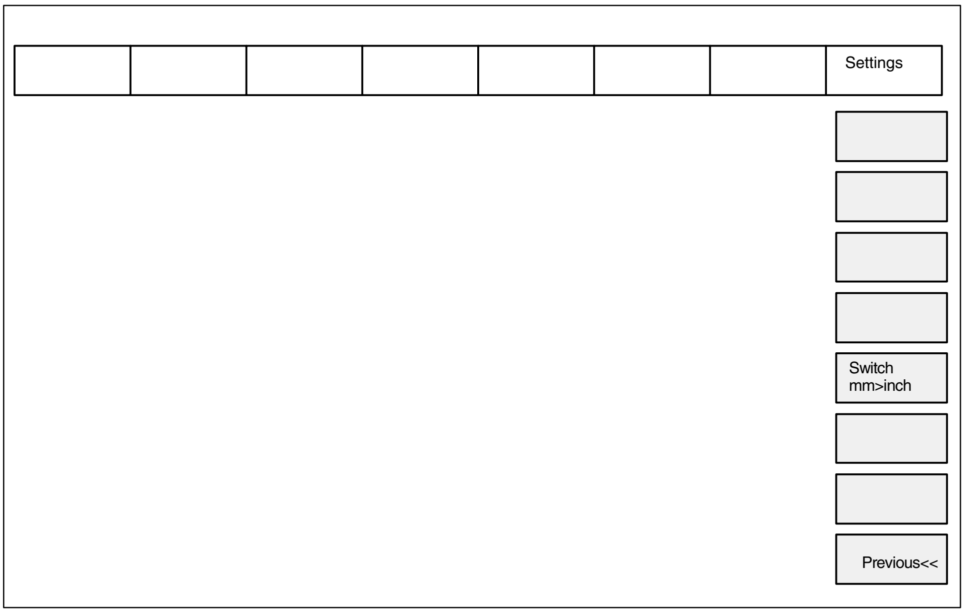


Fig. 4-1 Menu tree for the JOG mode, “Position” operating area

4.1 JOG mode – “Position” operating area

Operating sequences

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

+X ... −Y

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

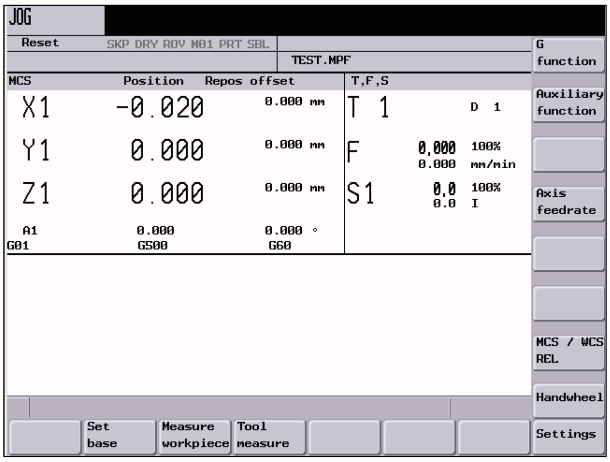
The axes will traverse continuously at the velocity stored in the setting data until the key is released. If the value of the setting data is zero, the value stored in the machine data is used.

If necessary set the velocity using the override switch.

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

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

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



```
JOG
Reset    SKP DRY ROV M01 PRT SBL
                                  TEST.MPF
MCS        Position   Repos offset   T,F,S
X1         -0.020     0.000 mm       T  1            D  1
Y1          0.000     0.000 mm       F      0,000  100%
                                            0.000  mm/min
Z1          0.000     0.000 mm       S1       0,0  100%
                                              0.0  I
A1          0.000     0.000 °
G01         G500      G60
```

Fig. 4-2 JOG main screen

Parameters

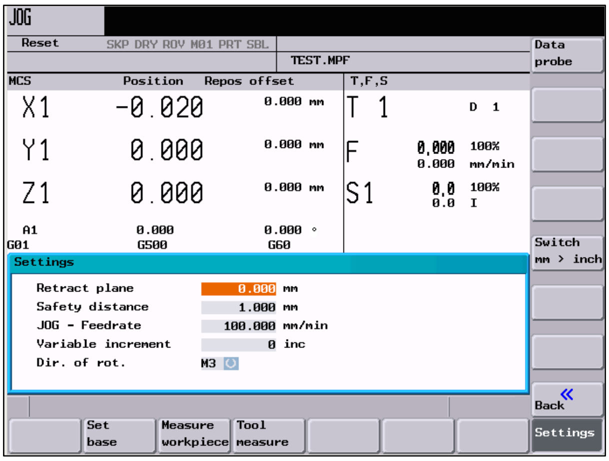
Table 4-1 Description of the parameters in the *JOG* start screen

Parameter	Explanation
MCS X Y C1 C2	Displays the address of the axes existing in the machine coordinate system (MCS)
+X .... –Y	If you traverse an axis in the positive (+) or negative () direction, a plus or minus sign will appear in the relevant field. If the axis is already in the required position, no sign is displayed.
Position mm	These fields display the current position of the axes in the MCS or WCS.
Repos. offset	If the axes are traversed in the “Program interrupted” condition in the *JOG* mode, the distance traversed by each axis is displayed referred to the interruption point.
G function	Displays important G functions
Feed F mm/ min	Displays the path feedrate actual value and setpoint
Tool	Displays the currently active tool with the current edge number

Softkeys

Settings

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



```
JOG
Reset        SKP DRY ROV M01 PRT SBL
                                 TEST.MPF
MCS          Position   Repos offset      T,F,S
X1           -0.020     0.000 mm          T 1          D 1
Y1            0.000     0.000 mm          F    0,000   100%
                                               0.000   mm/min
Z1            0.000     0.000 mm          S1     0,0   100%
                                                 0.0   I
A1            0.000     0.000 °
G01           G500      G60
Settings
   Retract plane            0.000 mm
   Safety distance          1.000 mm
   JOG - Feedrate         100.000 mm/min
   Variable increment           0 inc
   Dir. of rot.            M3
```

Fig. 4-3

Safety clearance: Safety clearance to the workpiece surface
This value defines the minimum clearance between the workpiece surface and the workpiece.

JOG feedrate: Feedrate value in the JOG mode

Switch mm>inch

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

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

Functionality

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

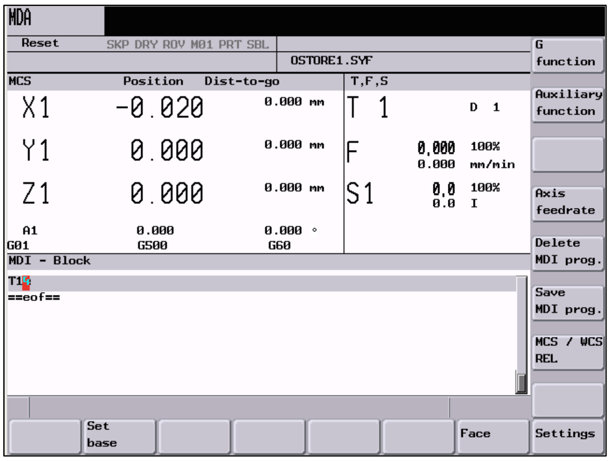
Caution

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

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

Operating sequences

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



```
MDA
Reset      SKP DRY ROV M01 PRT SBL
                                  OSTORE1.SYF
MCS         Position   Dist-to-go      T,F,S
X1          -0.020     0.000 mm        T 1            D 1
Y1           0.000     0.000 mm        F     0,000   100%
                                             0.000   mm/min
Z1           0.000     0.000 mm        S1      0,0   100%
                                               0.0   I
 A1          0.000     0.000 °
G01          G500      G60
MDI - Block
T1
==eof==
```

Fig. 4-4 MDA start screen

Enter one or several blocks using the keyboard.

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

The contents are stored even after editing. After restarting the NC, the axes can be traversed.

Parameters

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

Parameter	Explanation
MCS X Y C1 C2	Displays the existing axes in the MCS or WCS
+X ... –Y	If you traverse an axis in the positive (+) or negative () direction, a plus or minus sign will appear in the relevant field. If the axis is already in the required position, no sign is displayed.
Position mm	These fields display the current position of the axes in the MCS or WCS.
Distance-to-go	This field displays the distance to go of the axes in the MCS or WCS.
G function	Displays important G functions
Feedrate F	Displays the path feedrate actual value and setpoint in mm/min or mm/rev.
Tool	Displays the currently active tool with the current edge number (T..., D...).
Editing window	In the “Reset” program state, an editing window serves to input a part program block.

Softkeys

Settings

Refer to Section 4.1

G function

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

All G funct.

Displays all G functions.

Auxiliary function

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

Axis feedrate

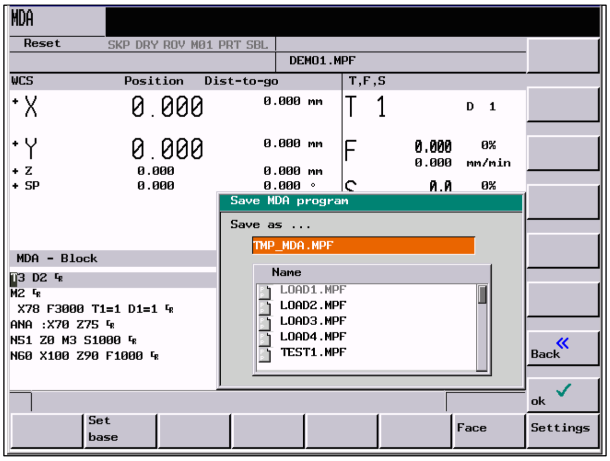
Use this softkey to display the Axis feedrate window. Selecting the softkey repeatedly will close the window.

Delete MDA progr.

Use this function to delete blocks from the program window.

Save MDA progr.

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



MDA

Reset SKP DRY ROV M01 PRT SBL

DEMO1.MPF

WCS	Position	Dist-to-go
+ X	0.000	0.000 mm
+ Y	0.000	0.000 mm
+ Z	0.000	0.000 mm
+ SP	0.000	0.000 °

T,F,S

T 1 D 1

F 0.000 0%
0.000 mm/min

S 0.0 0%

MDA - Block

```
T3 D2 ¶
M2 ¶
 X78 F3000 T1=1 D1=1 ¶
ANA :X70 Z75 ¶
N51 Z0 M3 S1000 ¶
N60 X100 Z90 F1000 ¶
```

Save MDA program

Save as ...

TMP_MDA.MPF

Name

- LOAD1.MPF
- LOAD2.MPF
- LOAD3.MPF
- LOAD4.MPF
- TEST1.MPF

Fig. 4-5

MCS-WCS-REL

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

AUTOMATIC Mode 5

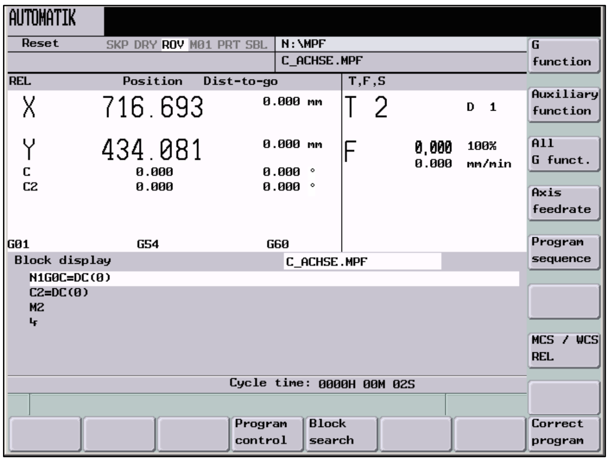
Prerequisite

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

Operating sequence

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

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



```
AUTOMATIK
Reset          SKP DRY ROV M01 PRT SBL  N:\MPF
                                        C_ACHSE.MPF
REL          Position   Dist-to-go      T,F,S
 X      716.693       0.000 mm          T 2            D  1
 Y      434.081       0.000 mm          F       0,000  100%
 C        0.000       0.000 °                   0.000  mm/min
 C2       0.000       0.000 °
G01            G54            G60
 Block display                    C_ACHSE.MPF
   N1G0C=DC(0)
   C2=DC(0)
   M2
   ↳
                    Cycle time: 0000H 00M 02S
```

Fig. 5-1 The *AUTOMATIC* start screen

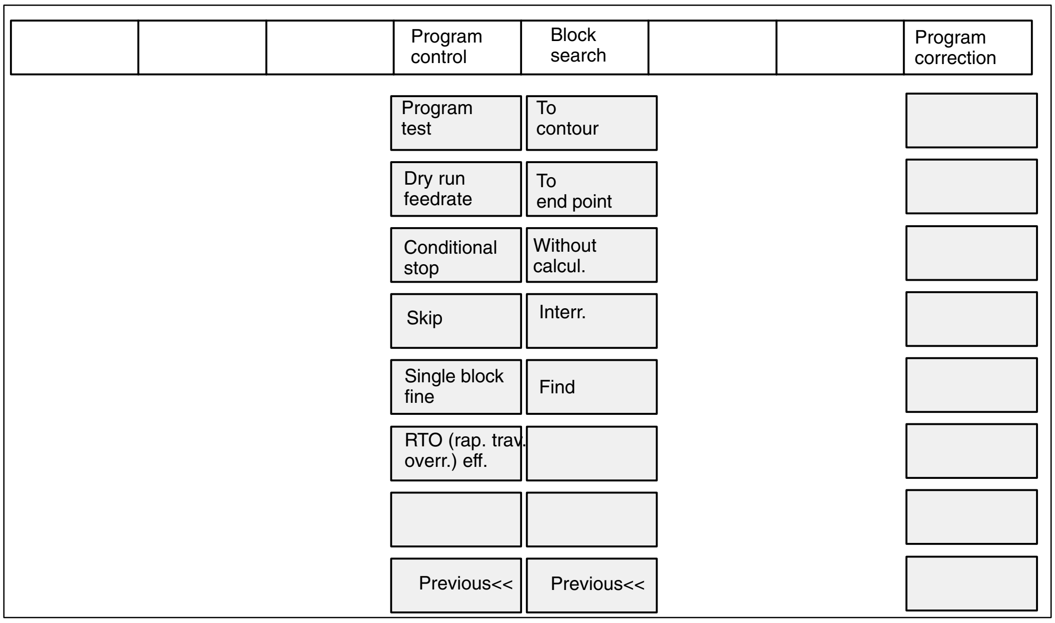


Fig. 5-2 The *AUTOMATIC* menu tree

Parameters

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

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

Softkeys

Program control

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

Program test

If “Program test” (PRT) is selected, the output of setpoints to axes and spindles is disabled. The set point display “simulates” the traversing movements.

Dry run feedrate

If you select this softkey, all traversing movements are performed with the feedrate setpoint specified via the “Dry run feed” setting data. Instead of the programmed motion commands, the dry run feedrate will be effective.

Conditional stop

If this function is active, the program execution is stopped at the blocks in which the miscellaneous function M01 is programmed.

Skipping

Program blocks marked with a slash in front of the block number are skipped during the program execution (e.g. “/N100”).

Single block fine

If this function is enabled, the part program blocks are executed separately as follows: Each block is decoded separately and execution is interrupted after every block. “Single Block fine” can only be selected in the RESET status.

RTO effective

The feedrate override switch also acts on the rapid traverse override.

Previous<<

Use this softkey to quit the screenform.

Block search

Use the block search function to go to the desired place in the program.

To contour

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

To end point

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

Without calcul.

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

Interr.

The cursor is positioned on the interruption point.

Find

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

Program correction

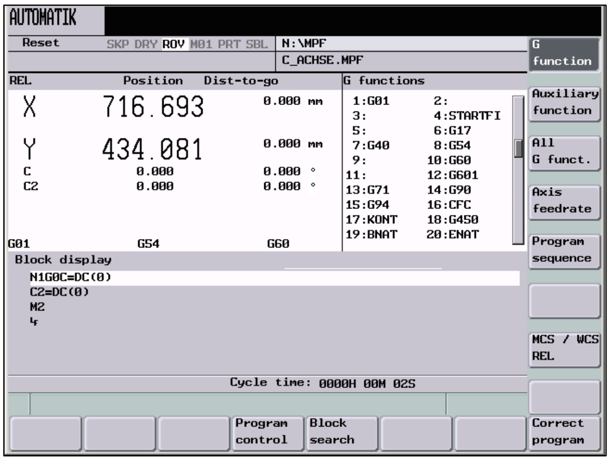
Use this softkey to correct a faulty program passage. Any changes are stored immediately.

G function

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

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

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



```
AUTOMATIK
Reset      SKP DRY ROV M01 PRT SBL   N:\MPF
                                     C_ACHSE.MPF
REL        Position   Dist-to-go     G functions
X     716.693         0.000 mm        1:G01      2:
                                      3:         4:STARTFI
Y     434.081         0.000 mm        5:         6:G17
C       0.000         0.000 °         7:G40      8:G54
C2      0.000         0.000 °         9:        10:G60
                                     11:        12:G601
                                     13:G71     14:G90
                                     15:G94     16:CFC
                                     17:KONT    18:G450
                                     19:BNAT    20:ENAT
G01           G54         G60
Block display
  N1G0C=DC(0)
  C2=DC(0)
  M2
  ↳
                 Cycle time: 0000H 00M 02S
```

Fig. 5-3 The *Active G function* window

All G funct.

Displays all G functions.

Auxiliary function

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

Axis feedrate

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

Program execution

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

MCS/WCS-REL

The values of the machine, workpiece or relative coordinate system are selected.

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

Functionality

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

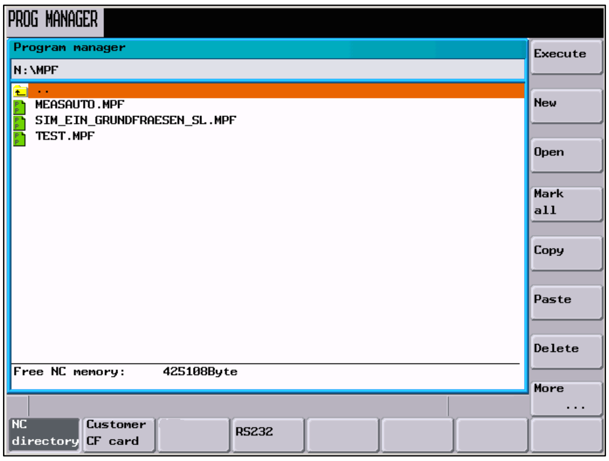
Operating sequence

⇥

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

PROGRAM MANAGER

The Program Manager is opened. Use the **<NC directory>** (default selection) or the **<Customer CF card>** softkey to obtain access to the appropriate directories.



PROG MANAGER

Program manager

N:\MPF

```
..
MEASAUTO.MPF
SIM_EIN_GRUNDFRAESEN_SL.MPF
TEST.MPF
```

Free NC memory: 425108Byte

Fig. 5-4 The “Program Manager” start screen

↑ ↓

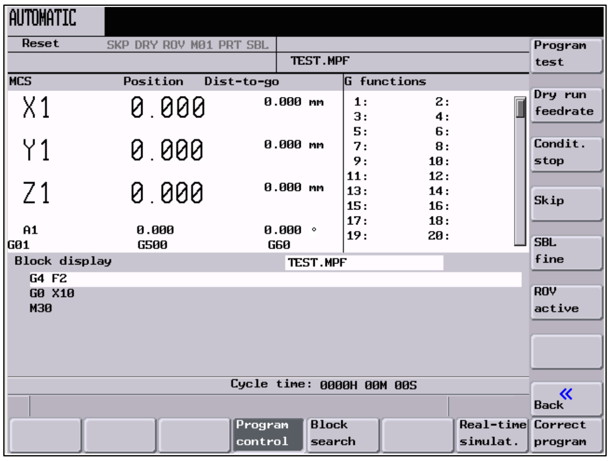
Position the cursor bar on the desired program.

Run

Use the **<Run>** (NC directory) or **<Run on external device>** softkey (with CF card) to select the program for execution. The name of the selected program is displayed in the “Program name” screen line.

Program control

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



MCS	Position	Dist-to-go
X1	0.000	0.000 mm
Y1	0.000	0.000 mm
Z1	0.000	0.000 mm
A1	0.000	0.000 °
G01	G500	G60

```
G4 F2
G0 X10
M30
```

Cycle time: 0000H 00M 00S

Fig. 5-5 Program control

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

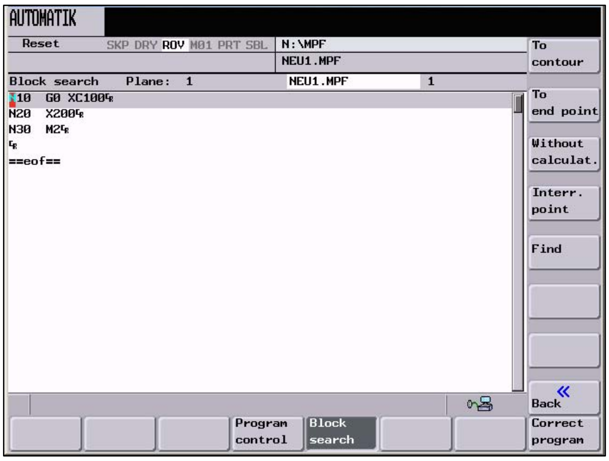
5.2 Block search “Machine” operating area

Operating sequence

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

Block search

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



Block search Plane: 1 NEU1.MPF 1

```
N10  G0 XC100
N20  X200
N30  M2

==eof==
```

Fig. 5-6 Block search

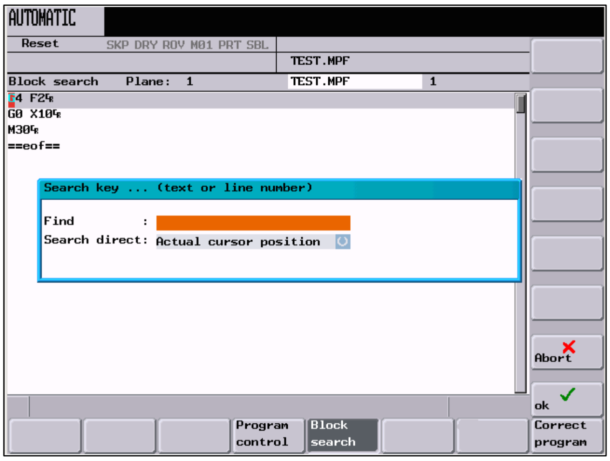
To contour — Block search to the block start

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

Without calcul. — Block search without calculation

Interr. — The interruption point is loaded.

Find — This softkey opens a dialog box where you can enter the number of the line or searched terms.



AUTOMATIC

Reset SKP DRY ROV M01 PRT SBL

TEST.MPF

Block search Plane: 1 TEST.MPF 1

```
G4 F2¶
G0 X10¶
M30¶
==eof==
```

Search key ... (text or line number)

Find :

Search direct: Actual cursor position

Fig. 5-7 Entering the searched term

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

Search result

The required block is displayed in the “Current block” window.

5.3 Stopping / canceling a part program

Operating sequence

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

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

5.4 Reapproach after cancellation

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

Operating sequence

Select the **AUTOMATIC** mode.

Block search

Use this softkey to open the “Block search” window for loading the interruption point.

Interr.

The interruption point is loaded.

To contour

Selecting this softkey starts the block search to the interruption point. An adjustment to the start position of the interrupted block is carried out.

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

5.5 Repositioning after interruption

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

Operating sequence

Select the **AUTOMATIC** mode.

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

Caution

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

5.6 Execution from external

Functionality

Use this softkey to transmit an external program to the control system via the CF card; pressing **<NC START>** will start this program immediately.

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

Operating sequence when executing a program from the CF card

Prerequisite: The control system is in the RESET condition.

→ PROGRAM MANAGER

Select the **AUTOMATIC** mode and the Program Manager using the appropriate keys on the machine control panel.

Customer CF card

Press the softkey and select the program to be executed using the cursor.

Execution from external

softkey.

The program is transmitted into the buffer memory and selected and displayed in the "Program Selection" automatically.

◇

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

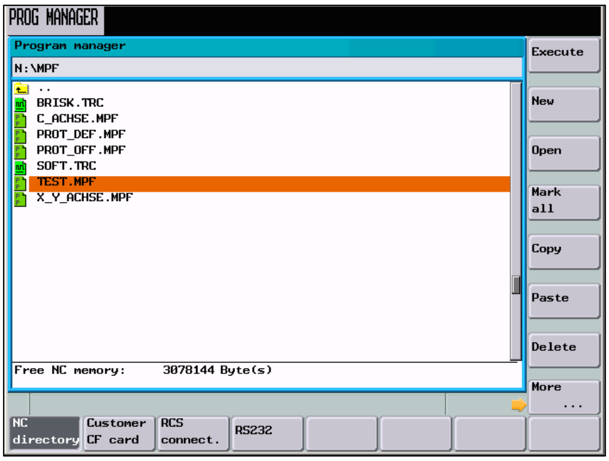
At the end of the program or in case of **RESET**, the program is automatically removed from the control system.

Part Programming 6

Operating sequence

PROGRAM MANAGER

To call the Program Manager, use the **<Program Manager>** key.



PROG MANAGER

Program manager

N:\MPF

```
..
BRISK.TRC
C_ACHSE.MPF
PROT_DEF.MPF
PROT_OFF.MPF
SOFT.TRC
TEST.MPF
X_Y_ACHSE.MPF
```

Free NC memory: 3078144 Byte(s)

Fig. 6-1 The *“Program Manager”* start screen

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

Softkeys

NC directory

Use this softkey to display the directories of the NC.

Run

Use this softkey to select the program on which the cursor is positioned for execution. The control system will switch to the position display. With the next **NC START**, the program is started.

New

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

Open

Use the “Open” softkey to open the file highlighted by the cursor for editing.

Select All

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

Note

Selecting individual files:

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

Copy

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

Paste

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

Delete

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

Use the **OK** softkey to execute the deletion order and **Abort** to discard.

Next ...

Use this softkey to branch to further functions.

Rename

Selecting the **Rename** softkey opens a window where you can rename the program you have selected beforehand using the cursor.

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

Preview window

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

Customer CF card

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

Execution from external

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

RCS connect.

RS232

Selecting this softkey provides the functions required to read out / read in files via the RS232 interface and the function “Program execution from external”.

Send

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

Receive

Use this softkey to load files via the RS232 interface.

For the settings of the interface, please refer to the **System** operating area (Chapter 7). The part programs must be transmitted using the text format.

Error log

Error log

6.1 Entering a new program “Program” operating area

Operating sequences

PROGRAM MANAGER

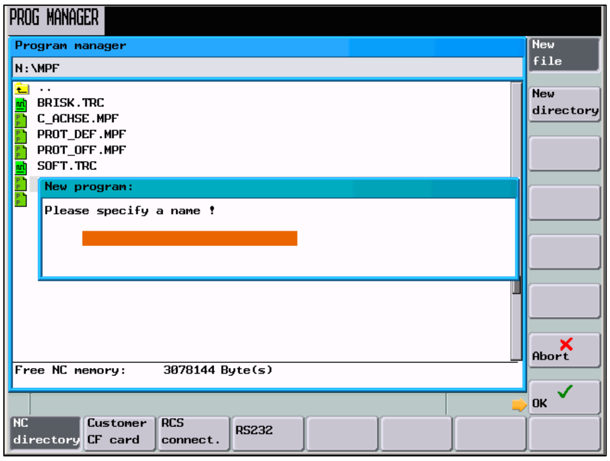
You have selected the Program Manager.

NC directory

Use either **<NC directory>** softkey or the Customer CF card softkey to select the location where you want to store the new program.

New file

Select the **<New file>** softkey to obtain a new dialog window into which the new main program or subroutine names are entered. The .MPF extension for main programs is entered automatically. The .SPF extension for subroutines must be entered along with the program name.


PROG MANAGER

Program manager

N:\MPF

..
BRISK.TRC
C_ACHSE.MPF
PROT_DEF.MPF
PROT_OFF.MPF
SOFT.TRC

New program:

Please specify a name !

Free NC memory: 3078144 Byte(s)

Fig. 6-2 The *New program* interactive screenform

A J W Z

Enter the name for the new program.

OK ✓

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

Abort ✗

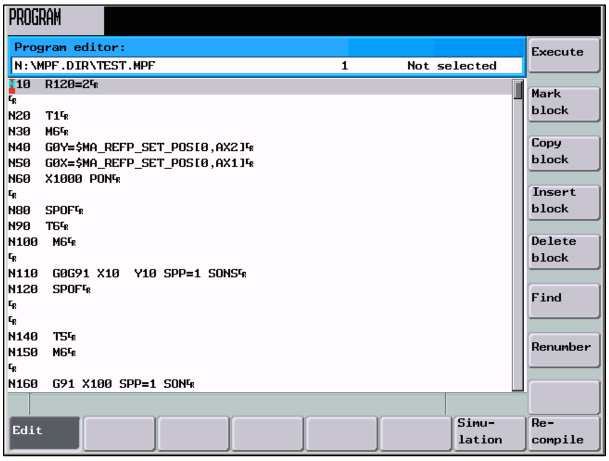
Use **<Abort>** to cancel the creation of the program; the window will be closed.

6.2 Editing part programs “Program” operating area

Functionality

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

Any modifications to the part program are stored immediately.



Program editor:
N:\MPF.DIR\TEST.MPF 1 Not selected

```
N10  R120=2␊
␊
N20  T1␊
N30  M6␊
N40  G0Y=$MA_REFP_SET_POS[0,AX2]␊
N50  G0X=$MA_REFP_SET_POS[0,AX1]␊
N60  X1000 PON␊
␊
N80  SPOF␊
N90  T6␊
N100  M6␊
␊
N110  G0G91 X10  Y10 SPP=1 SONS␊
N120  SPOF␊
␊
␊
N140  T5␊
N150  M6␊
␊
N160  G91 X100 SPP=1 SON␊
```

Fig. 6-3 The “Program editor” start screen

Menu tree

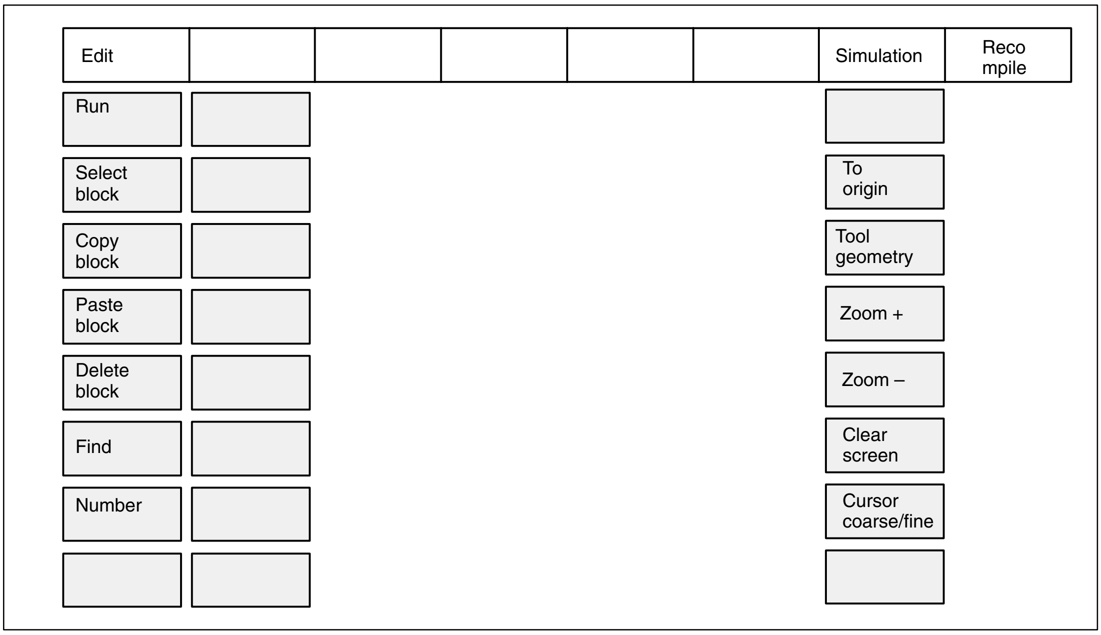


Fig. 6-4 The “Program” menu tree

Softkeys

Edit

Use this softkey to edit text.

Run

Use this softkey to execute the selected file.

Select block

Use this softkey to select a text segment up to the current cursor position using the arrow keys.

Copy block

Use this softkey to copy a selected block to the clipboard.

Paste block

Use this softkey to paste a text from the clipboard at the current cursor position.

Delete block

Use this softkey to delete a selected text.

Find

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

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

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

Number

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

Simulation

The simulation is described in Section 6.3.

Recompile

For recompilation, position the cursor on the cycle calling line in the program. This function decodes the cycle name and prepares the screenform with the relevant parameters. If there are any parameters beyond the range of validity, the function will automatically use the default values. After closing the screenform, the original parameter block is replaced by the corrected block.

Note: Only automatically generated blocks can be recompiled.

6.3 Simulation

Functionality

Generally, simulation means that the programmed tool path of the selected program can be traced using broken-line graphics.

The display behavior of the part simulation is different for nibbling. Instead of traversing movements, the simulation displays the **clamp protection zones** and the **performed strokes** with the appropriate tool profile. Graphical representation is only provided with stroke initiation selected.

Attention

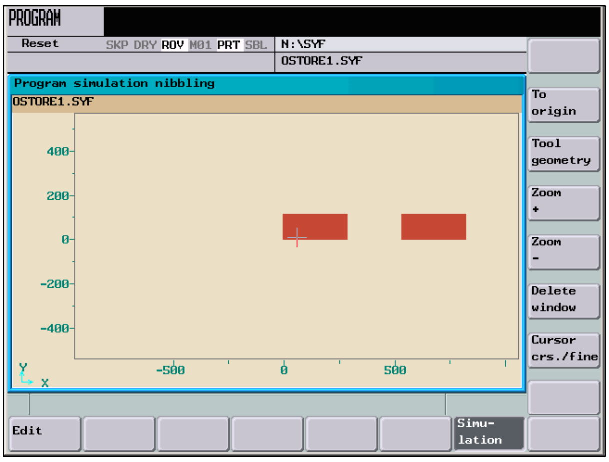
If the protection zone was specified via the active NC program, it is not represented in the simulation.

Operating sequence

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

PROGRAM | Simulation

The start screen is displayed.



PROGRAM
Reset | SKP DRY ROV M01 PRT SBL | N:\SYF
OSTORE1.SYF

Program simulation nibbling
OSTORE1.SYF

Fig. 6-5 The “Simulation” start screen

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

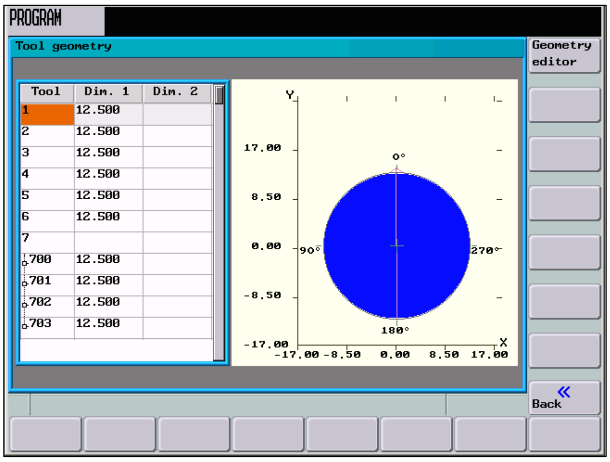
Softkeys

To origin

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

Tool geometry

Use this function to display all setting data for the control system in the form of a list. The geometrical form and the orientation of the tool highlighted by the cursor are displayed. Newly set-up tools are automatically assigned a round form.



Tool	Dim. 1	Dim. 2
1	12.500	
2	12.500	
3	12.500	
4	12.500	
5	12.500	
6	12.500	
7		
700	12.500	
701	12.500	
702	12.500	
703	12.500	

Fig. 6-6 Display of the tool geometry

Geometry editor

The geometry editor is opened.

Zoom +

Use this softkey to zoom out the displayed section.

Zoom –

Use this softkey to zoom in the displayed section.

Clear screen

Use this softkey to delete the visible image.

Cursor coarse/fine

Use this softkey to change the cursor increment.

6.3.1 Geometry editor

Functionality

The geometry editor can be used to define the form of the punch of a tool. To do so, select the geometrical form using the softkey functions offered.

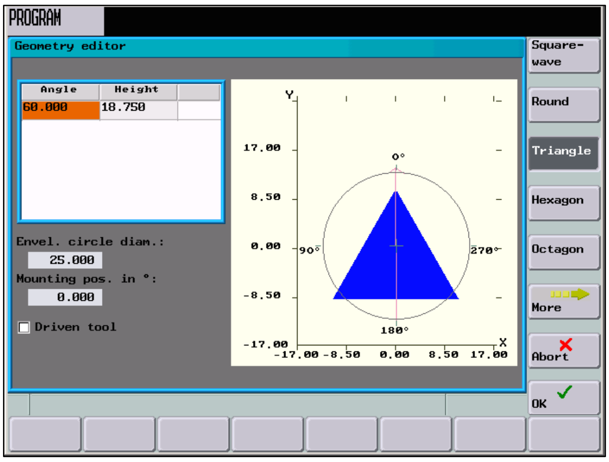
The control system offers the following standard geometries:

- Rectangle
- Round
- Triangle
- Hexagon
- Octagon
- Elongated hole
- Double “D”
- Long “D”

Subsequently, you can modify the punch size using the relevant input fields. The resulting geometrical form is displayed on the right-hand side of the interactive screen at completion of your input. All punches are displayed rotational-symmetrically.

The “Envelope circle diameter” field displays the appropriate envelope circle diameter.

To define the tool orientation, use the **Angle** input field.



Geometry editor

Angle	Height	
60.000	18.750	

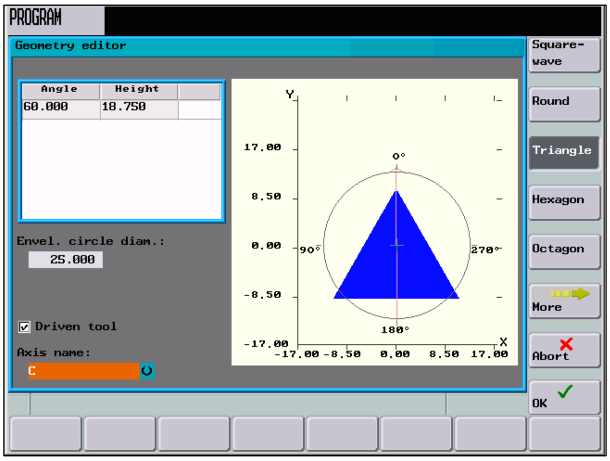
Envel. circle diam.: 25.000

Mounting pos. in °: 0.000

☐ Driven tool

Fig. 6-7 Changing the tool orientation

If the toolholder is coupled to a rotary axis, a this rotary axis can be assigned to a tool. Activate the **Driven tool** checkbox and select the axis name.



PROGRAM

Geometry editor

Angle	Height
60.000	18.750

Envel. circle diam.: 25.000

☑ Driven tool

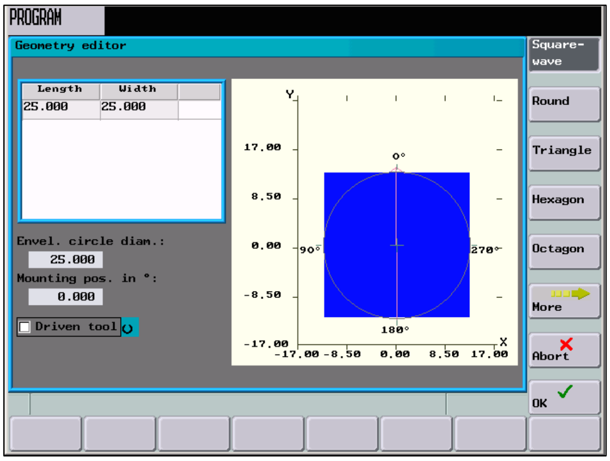
Axis name: C

Fig. 6-8 Assigning the rotary axis to the tool

Selecting the **<OK>** softkey function accepts the data and lets you return to the tool geometry list.

Rectangular punch form

Use the **Length** and **Width** input fields to define the dimensions of the rectangle.



PROGRAM

Geometry editor

Length	Width
25.000	25.000

Envel. circle diam.: 25.000

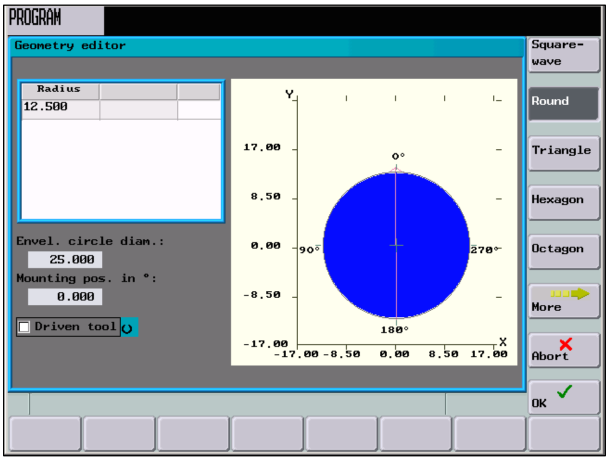
Mounting pos. in °: 0.000

□ Driven tool

Fig. 6-9 Rectangular punch form

Round punch form

Use the **Radius** input field to vary the punch diameter.



Radius
12.500

Envel. circle diam.: 25.000

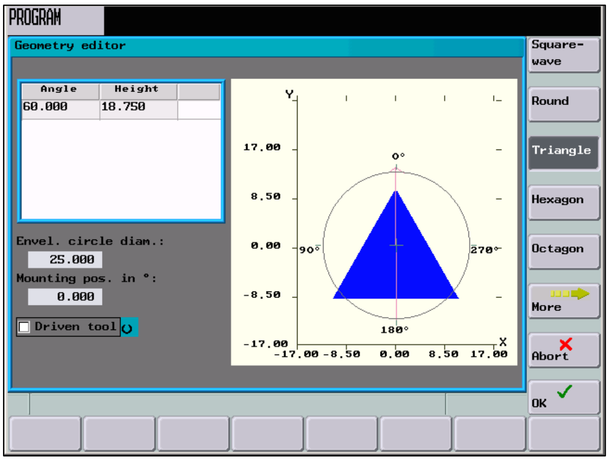
Mounting pos. in °: 0.000

□ Driven tool

Fig. 6-10 Round punch form

Triangular punch form

Use the **Angle** input field to define the aperture angle of the triangle. The height of the triangle is determined by the **Height** field. The resulting triangle is centered to the center point of the envelope circle.



Angle	Height
60.000	18.750

Envel. circle diam.: 25.000

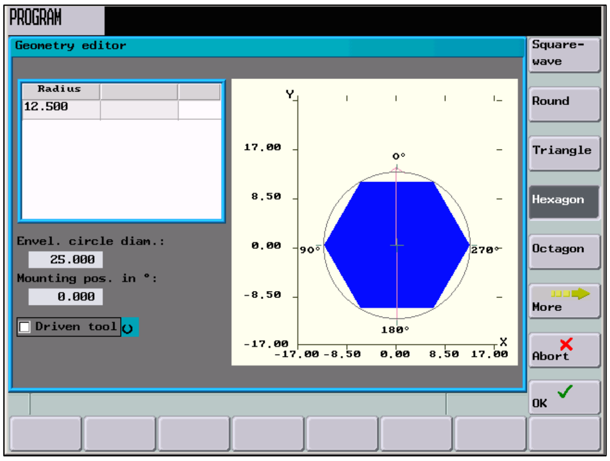
Mounting pos. in °: 0.000

□ Driven tool

Fig. 6-11 Triangular punch form

Hexagon punch form

Use the **Radius** input field to vary the punch diameter.



Radius
12.500

Envel. circle diam.: 25.000

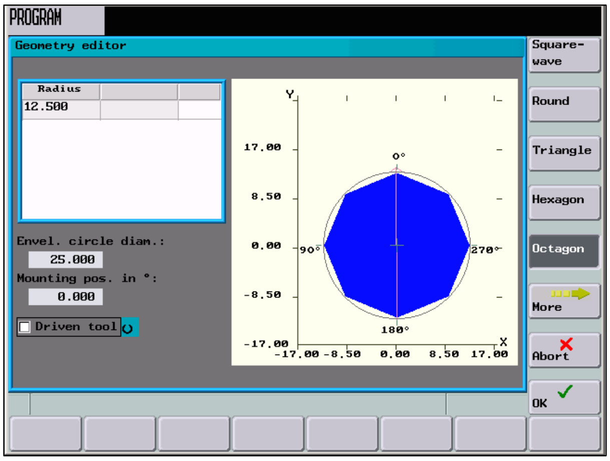
Mounting pos. in °: 0.000

□ Driven tool

Fig. 6-12 Hexagon punch form

Octagon punch form

Use the **Radius** input field to vary the punch diameter.



Radius
12.500

Envel. circle diam.: 25.000

Mounting pos. in °: 0.000

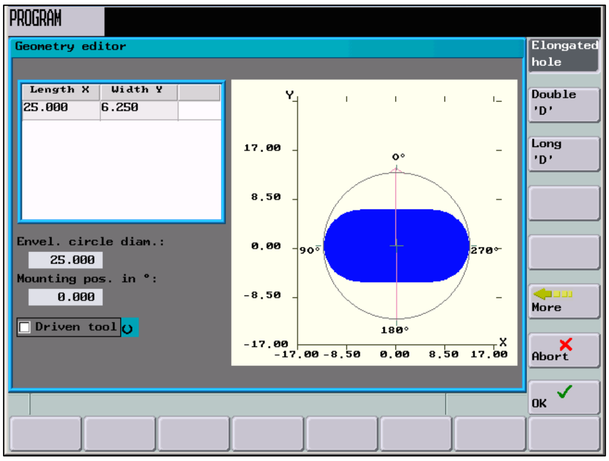
□ Driven tool

Fig. 6-13 Octagon punch form

Elongated hole punch form

Use the **Length** and **Width** input fields to define the dimensions of the elongated hole.

The rounding radius results from the ratio between length and width.



PROGRAM

Geometry editor

Length X	Width Y
25.000	6.250

Envel. circle diam.: 25.000

Mounting pos. in °: 0.000

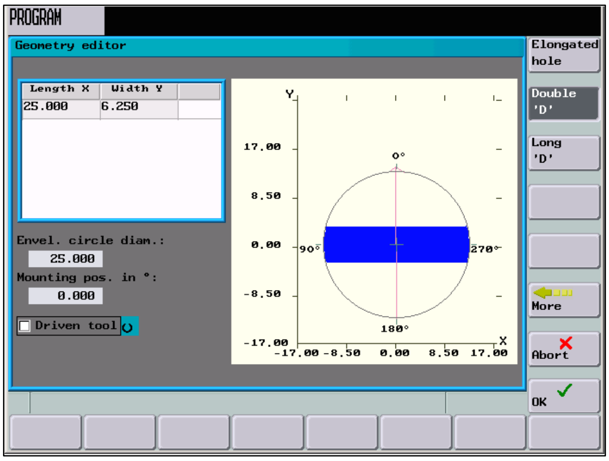
□ Driven tool

Fig. 6-14 Elongated hole punch form

Double “D” punch form

Use the **Width** input field to define the dimensions of the **Double “D”** punch form.

The rounding radius always corresponds to the radius of the envelope circle.



PROGRAM

Geometry editor

Length X	Width Y
25.000	6.250

Envel. circle diam.: 25.000

Mounting pos. in °: 0.000

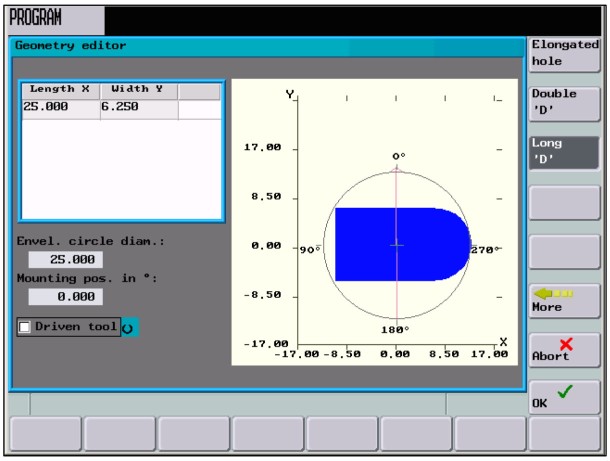
□ Driven tool

Fig. 6-15 Double “D” punch form

Long “D” punch form

Use the **Width** input field to define the dimensions of the **Long “D”** punch form.

The length is determined by the envelope circle diameter.



Length X	Width Y
25.000	6.250

Envel. circle diam.: 25.000

Mounting pos. in °: 0.000

□ Driven tool

Fig. 6-16 Long “D” punch form

6.4 Data transfer via the RS232 interface

Functionality

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

Operating sequence

PROGRAM MANAGER

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

Select the data to be transmitted using either the cursor or the **<Select all>** softkey.

Copy

Thereafter, copy the data to the clipboard.

RS232

Select the **<RS232>** softkey and select the desired transfer mode.

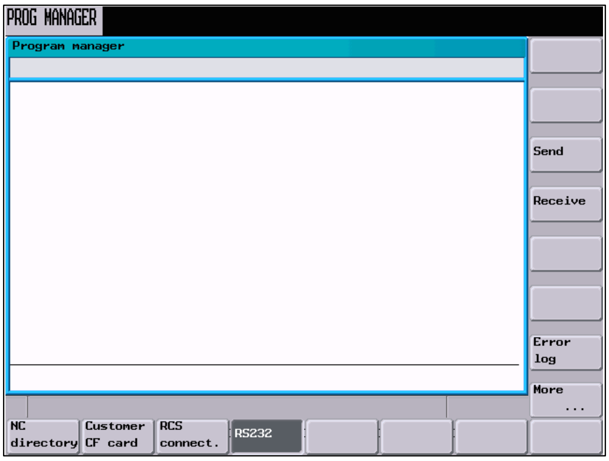


Fig. 6-17 Reading out a program

Send

Use **<Send>** to start the data transfer. All data copied to the clipboard will be transmitted.

Further softkeys

Receive

Use this softkey to load files via the RS232 interface.

Error log

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

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

Transmission messages:

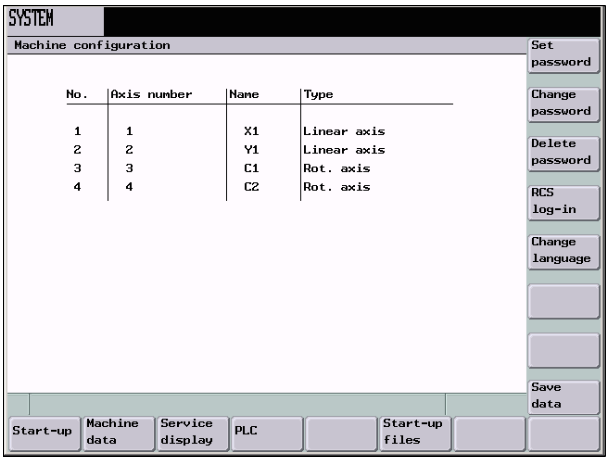
OK	Transmission completed successfully
ERR EOF	End-of-text character received, but archive file incomplete
Time Out	The time monitoring is reporting an interruption of the data transfer
User Abort	Data transfer aborted by the **<Stop>** softkey
Error Com	Error at the COM 1 port
NC / PLC Error	Error message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

System

7

Functionality

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



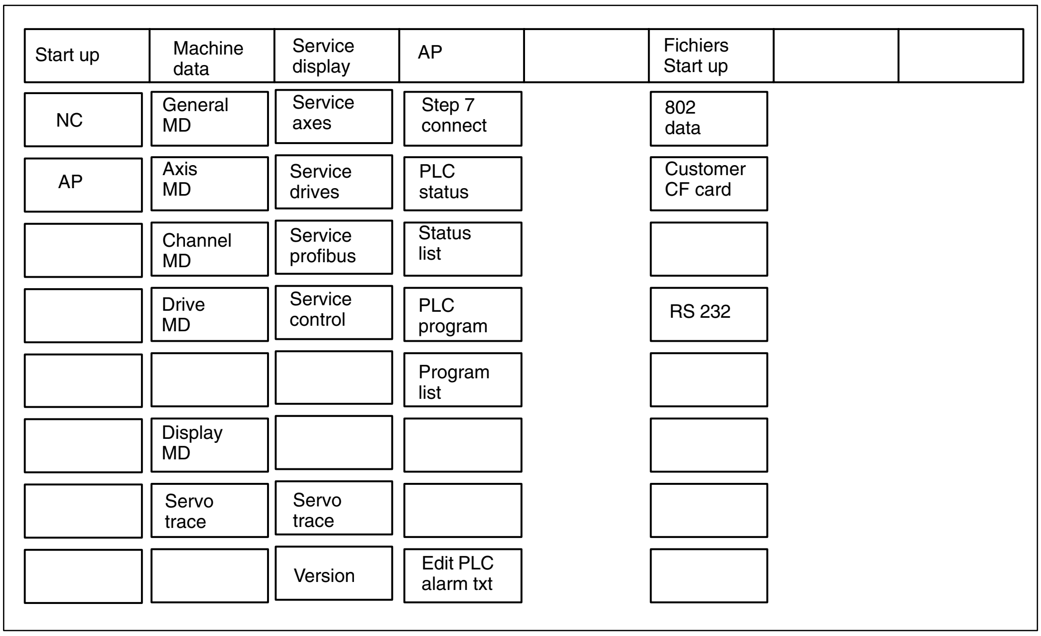
SYSTEM

Machine configuration

No.	Axis number	Name	Type
1	1	X1	Linear axis
2	2	Y1	Linear axis
3	3	C1	Rot. axis
4	4	C2	Rot. axis

Fig. 7-1 The “System” start screen

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



Start up	Machine data	Service display	AP		Fichiers Start up		
NC	General MD	Service axes	Step 7 connect		802 data		
AP	Axis MD	Service drives	PLC status		Customer CF card		
	Channel MD	Service profibus	Status list				
	Drive MD	Service control	PLC program		RS 232		
			Program list				
	Display MD						
	Servo trace	Servo trace					
		Version	Edit PLC alarm txt				

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

Softkey

Set password

Setting the password

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

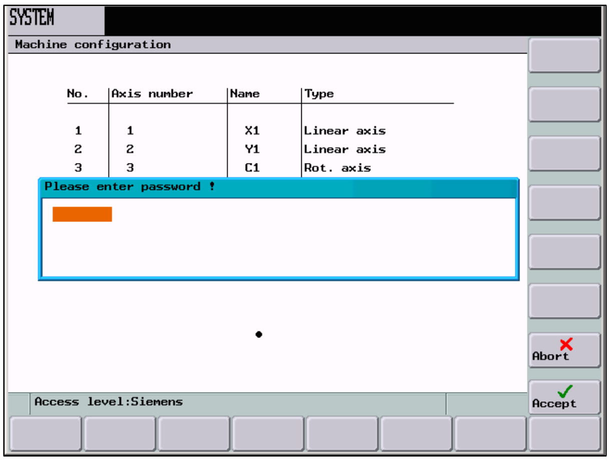
- System password
- Manufacturer password
- User password

According to the access level selected, certain data can be changed.

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

Note for the Reader

See also: SINUMERIK 802D sl “Lists”



SYSTEM

Machine configuration

No.	Axis number	Name	Type
1	1	X1	Linear axis
2	2	Y1	Linear axis
3	3	C1	Rot. axis

Please enter password !

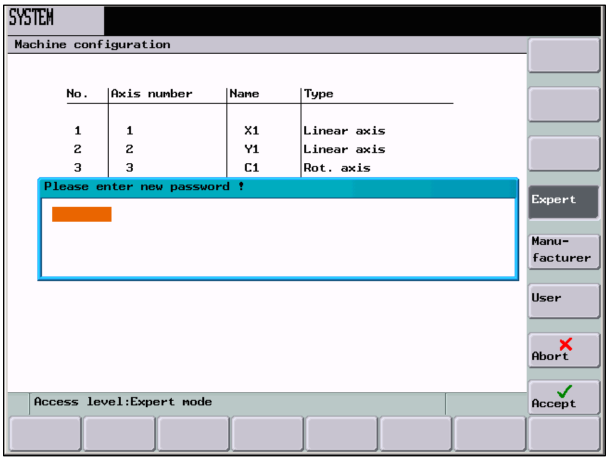
Access level:Siemens

Fig. 7-3 Entering the password

Selecting the **<OK>** softkey sets the password.
Use **<Abort>** to return to the *System* main screen without performing any action.

Change password

Changing the password



Machine configuration

No.	Axis number	Name	Type
1	1	X1	Linear axis
2	2	Y1	Linear axis
3	3	C1	Rot. axis

Please enter new password !

Access level:Expert mode

Fig. 7-4 Changing the password

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

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

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

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

Use **<Abort>** to return without any action to the “Start-up” main screen.

Delete password

Resetting the access right

RCS log-in

User log-in to the network (see Section 1.5)

Change language

Changing the language

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

Save data

Making a data backup

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

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

Startup

Startup

NC

Select the power-up mode of the NC.
Use the cursor to select the appropriate mode.

- Normal power-up
  The system is restarted.
- Power-up with default data
  Cold restart with the default values (restores the default condition as on delivery)
- Power-up with saved data
  Cold restart with the data saved last (see “Data backup”)

PLC

The PLC can be started in the following modes:

- **Restart** Restart
- **Overall reset** Overall reset

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

OK

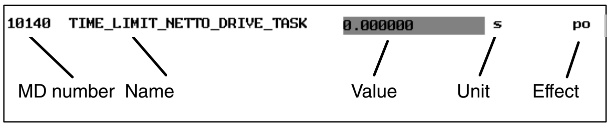
Use **<OK>** to RESET the control system and to carry out a restart in the mode selected.

Use **<RECALL>** to return without any action to the main screen.

Machine data

Machine data

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



```
10140  TIME_LIMIT_NETTO_DRIVE_TASK     0.000000    s       po
```

Fig. 7-5 Structure of a machine data line

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

Caution

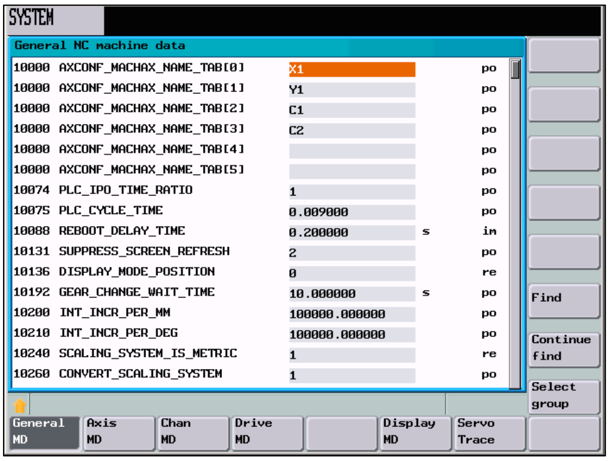
Faulty parameterization may result in destruction of the machine.

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

General MD

General machine data

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



General NC machine data

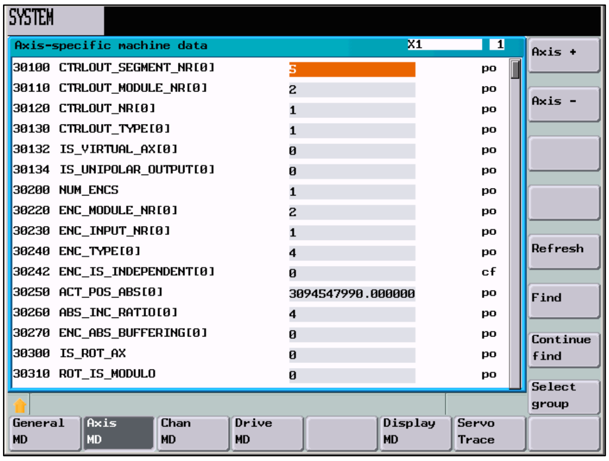
10000	AXCONF_MACHAX_NAME_TAB[0]	X1		po
10000	AXCONF_MACHAX_NAME_TAB[1]	Y1		po
10000	AXCONF_MACHAX_NAME_TAB[2]	C1		po
10000	AXCONF_MACHAX_NAME_TAB[3]	C2		po
10000	AXCONF_MACHAX_NAME_TAB[4]			po
10000	AXCONF_MACHAX_NAME_TAB[5]			po
10074	PLC_IPO_TIME_RATIO	1		po
10075	PLC_CYCLE_TIME	0.009000		po
10088	REBOOT_DELAY_TIME	0.200000	s	im
10131	SUPPRESS_SCREEN_REFRESH	2		po
10136	DISPLAY_MODE_POSITION	0		re
10192	GEAR_CHANGE_WAIT_TIME	10.000000	s	po
10200	INT_INCR_PER_MM	100000.000000		po
10210	INT_INCR_PER_DEG	100000.000000		po
10240	SCALING_SYSTEM_IS_METRIC	1		re
10260	CONVERT_SCALING_SYSTEM	1		po

Fig. 7-6 The “Machine data” start screen

Axis MD

Axis-specific machine data

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



Axis-specific machine data

30100	CTRLOUT_SEGMENT_NR[0]	5	po
30110	CTRLOUT_MODULE_NR[0]	2	po
30120	CTRLOUT_NR[0]	1	po
30130	CTRLOUT_TYPE[0]	1	po
30132	IS_VIRTUAL_AX[0]	0	po
30134	IS_UNIPOLAR_OUTPUT[0]	0	po
30200	NUM_ENCS	1	po
30220	ENC_MODULE_NR[0]	2	po
30230	ENC_INPUT_NR[0]	1	po
30240	ENC_TYPE[0]	4	po
30242	ENC_IS_INDEPENDENT[0]	0	cf
30250	ACT_POS_ABS[0]	3094547990.000000	po
30260	ABS_INC_RATIO[0]	4	po
30270	ENC_ABS_BUFFERING[0]	0	po
30300	IS_ROT_AX	0	po
30310	ROT_IS_MODULO	0	po

Fig. 7-7 Axis-specific machine data

The data of axis 1 are displayed.

Axis +

Axis –

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

Find

Find

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

The cursor will jump to the data searched.

Continue search

Use this softkey to continue searching for the next match.

Select group

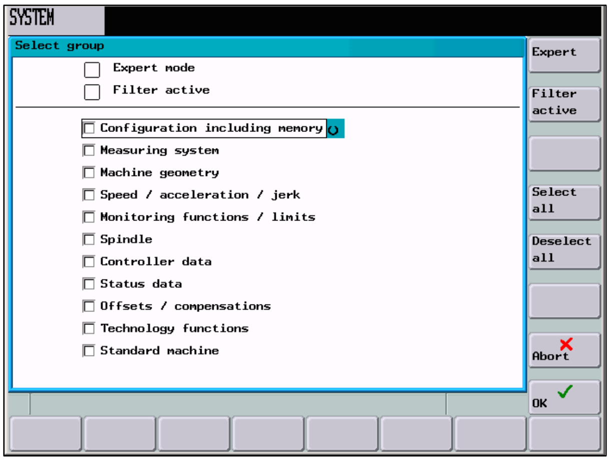
This function provides various display filters for the active machine data group. Further softkeys are provided:

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

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

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

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



Select group

□ Expert mode
□ Filter active

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

Fig. 7-8 Display filter

Channel MD

Channel-specific machine data

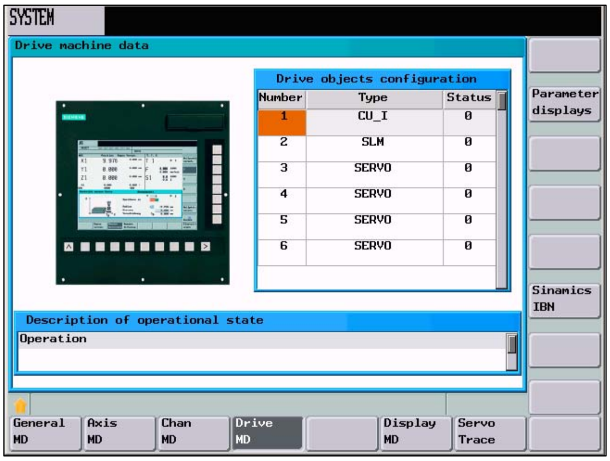
Open the “Channel-specific machine data” window. Use the PageUp / PageDown keys to browse forward / backward.

Drive MD

SINAMICS drive machine data

Open the “Drive machine data” dialog box.

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



Drive machine data

Drive objects configuration

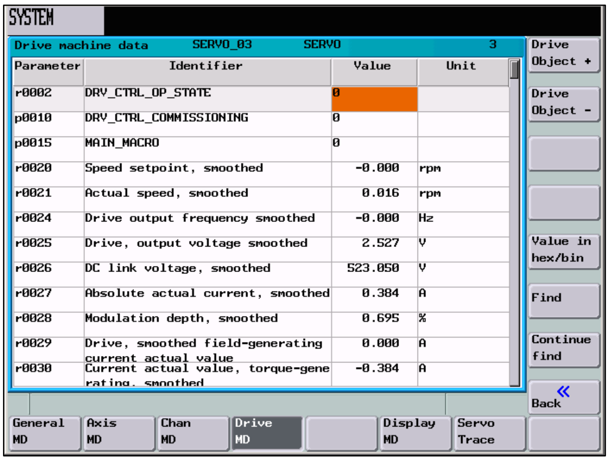
Number	Type	Status
1	CU_I	0
2	SLM	0
3	SERVO	0
4	SERVO	0
5	SERVO	0
6	SERVO	0

Description of operational state

Operation

Fig. 7-9 Configuration overview

To display all parameters, position the cursor on the appropriate unit and select the **<Parameter display>** softkey. For a description of the parameters, please refer to the documentation of SINAMICS drives.



Drive machine data SERVO_03 SERVO 3

Parameter	Identifier	Value	Unit
r0002	DRV_CTRL_OP_STATE	0	
p0010	DRV_CTRL_COMMISSIONING	0	
p0015	MAIN_MACRO	0	
r0020	Speed setpoint, smoothed	-0.000	rpm
r0021	Actual speed, smoothed	0.016	rpm
r0024	Drive output frequency smoothed	-0.000	Hz
r0025	Drive, output voltage smoothed	2.527	V
r0026	DC link voltage, smoothed	523.050	V
r0027	Absolute actual current, smoothed	0.384	A
r0028	Modulation depth, smoothed	0.695	%
r0029	Drive, smoothed field-generating current actual value	0.000	A
r0030	Current actual value, torque-generating, smoothed	-0.384	A

Fig. 7-10 Parameter list

Display MD

Display machine data

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

Note for the Reader

For a description of the machine data, please refer to the
SINUMERIK 802D sl “Lists” Manual.

Changing colors

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

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

Select the **<OK>** softkey to accept your settings and to quit the dialog. Selecting the **<Abort>** softkey will quit the dialog box without saving your changes.

Softkey color

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

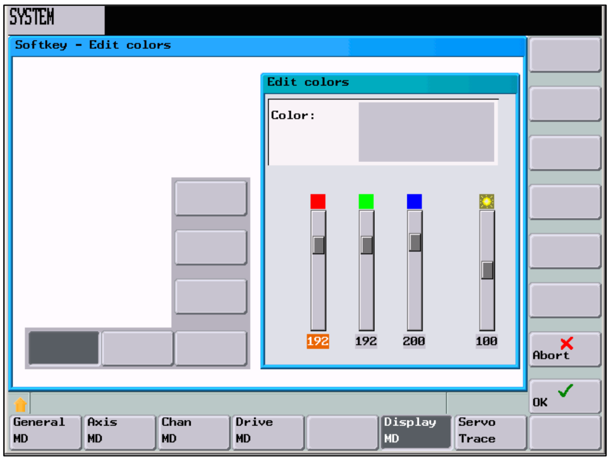


Fig. 7-11 Editing the softkey colors

Window color

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

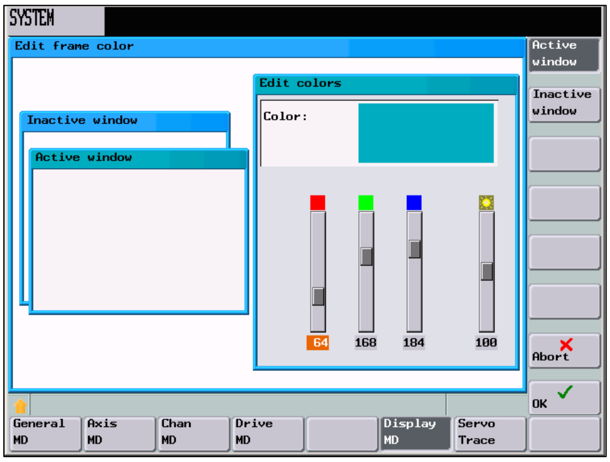


Fig. 7-12 Editing the frame color

Service display

Selecting this softkey displays the *Service axes* window.

Service axes

This window displays information in respect of the axis.

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

Service drives

This window displays information in respect of the digital drive.

Service profibus

This window displays information in respect of the PROFIBUS settings.

Service control

Use this softkey to activate the action log.

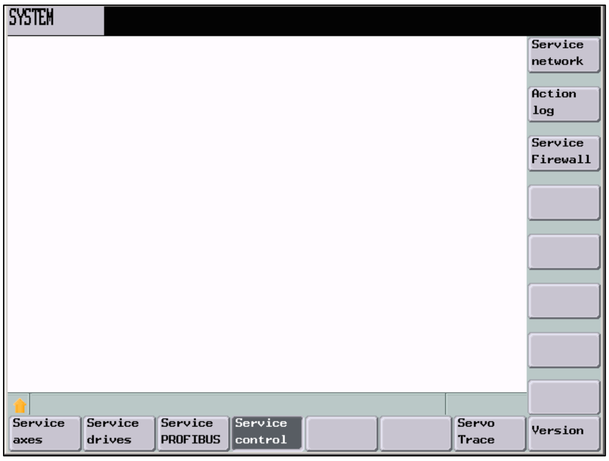


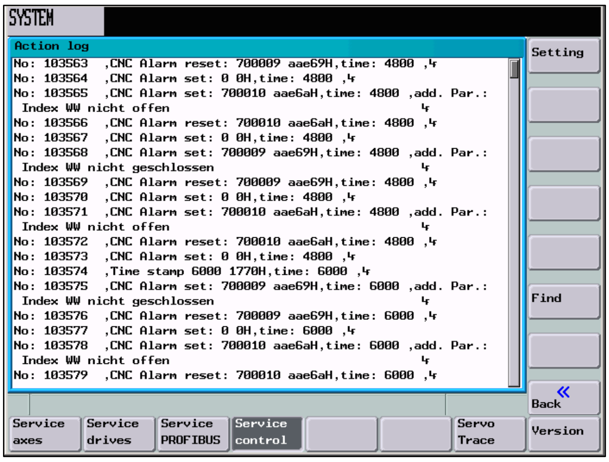
Fig. 7-13 *The “Service control” screen*

Service network

Network configuration

Action log

The **<Action log>** function is intended for servicing and displays all recorded events in the form of a list.



```
Action log
No: 103563  ,CNC Alarm reset: 700009 aae69H,time: 4800 ,␊
No: 103564  ,CNC Alarm set: 0 0H,time: 4800 ,␊
No: 103565  ,CNC Alarm set: 700010 aae6aH,time: 4800 ,add. Par.:
 Index WW nicht offen                                     ␊
No: 103566  ,CNC Alarm reset: 700010 aae6aH,time: 4800 ,␊
No: 103567  ,CNC Alarm set: 0 0H,time: 4800 ,␊
No: 103568  ,CNC Alarm set: 700009 aae69H,time: 4800 ,add. Par.:
 Index WW nicht geschlossen                               ␊
No: 103569  ,CNC Alarm reset: 700009 aae69H,time: 4800 ,␊
No: 103570  ,CNC Alarm set: 0 0H,time: 4800 ,␊
No: 103571  ,CNC Alarm set: 700010 aae6aH,time: 4800 ,add. Par.:
 Index WW nicht offen                                     ␊
No: 103572  ,CNC Alarm reset: 700010 aae6aH,time: 4800 ,␊
No: 103573  ,CNC Alarm set: 0 0H,time: 4800 ,␊
No: 103574  ,Time stamp 6000 1770H,time: 6000 ,␊
No: 103575  ,CNC Alarm set: 700009 aae69H,time: 6000 ,add. Par.:
 Index WW nicht geschlossen                               ␊
No: 103576  ,CNC Alarm reset: 700009 aae69H,time: 6000 ,␊
No: 103577  ,CNC Alarm set: 0 0H,time: 6000 ,␊
No: 103578  ,CNC Alarm set: 700010 aae6aH,time: 6000 ,add. Par.:
 Index WW nicht offen                                     ␊
No: 103579  ,CNC Alarm reset: 700010 aae6aH,time: 6000 ,␊
```

Fig. 7-14 *Action log*

Set-tings

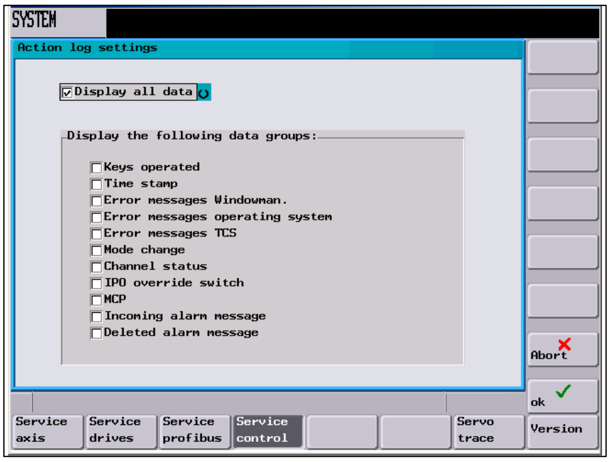
This dialog can be used to select certain events for display.
To switch between the “Display all data” and Display data groups“ fields, use the TAB key.

Table 7-1 Data groups

Group	Meaning
Actuated keys	Keyboard input
Time stamp	Time stamp
Window manager error message	Error messages issued by the Windows manager (only system-internal meaning)
Operating system error message	Error messages issued by the QW operating system (only system internal meaning)

Table 7-1 Data groups, continued

Group	Meaning
TKS error message	Error messages issued by the object request broker (only system-internal meaning)
Mode change	Selected mode
Channel status	Channel status
IPO override switch	Override value set
MCP	Machine control panel
Incoming alarm message	NC / PLC alarms
Cleared alarm message	Canceled NC / PLC alarms



Action log settings

☑ Display all data

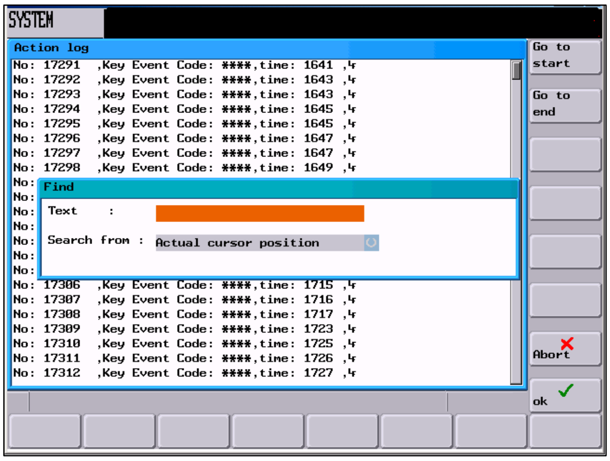
Display the following data groups:

- □ Keys operated
- □ Time stamp
- □ Error messages Windowman.
- □ Error messages operating system
- □ Error messages TCS
- □ Mode change
- □ Channel status
- □ IPO override switch
- □ MCP
- □ Incoming alarm message
- □ Deleted alarm message

Fig. 7-15

Find

Use this function to browse the event list for the entered term you are looking for. Searching can be started either from the current cursor position or from the start of the list.



Action log

```
No: 17291  ,Key Event Code: ****,time: 1641 ,↵
No: 17292  ,Key Event Code: ****,time: 1643 ,↵
No: 17293  ,Key Event Code: ****,time: 1643 ,↵
No: 17294  ,Key Event Code: ****,time: 1645 ,↵
No: 17295  ,Key Event Code: ****,time: 1645 ,↵
No: 17296  ,Key Event Code: ****,time: 1647 ,↵
No: 17297  ,Key Event Code: ****,time: 1647 ,↵
No: 17298  ,Key Event Code: ****,time: 1649 ,↵
No:
No:
No:
No:
No:
No:
No:
No: 17306  ,Key Event Code: ****,time: 1715 ,↵
No: 17307  ,Key Event Code: ****,time: 1716 ,↵
No: 17308  ,Key Event Code: ****,time: 1717 ,↵
No: 17309  ,Key Event Code: ****,time: 1723 ,↵
No: 17310  ,Key Event Code: ****,time: 1725 ,↵
No: 17311  ,Key Event Code: ****,time: 1726 ,↵
No: 17312  ,Key Event Code: ****,time: 1727 ,↵
```

Find

Text :

Search from : Actual cursor position

Fig. 7-16

Service firewall

Configuring the firewall (cf. Section 1.5)

Servo trace

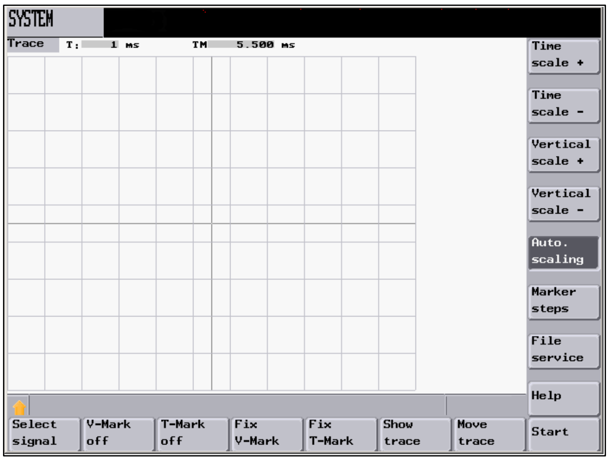
To optimize the drives, an oscilloscope function is provided for graphical representation

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

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

To analyze the result, the following functions are provided:

- Changing and scaling of abscissa and ordinate;
- Measuring of a value using the horizontal or vertical marker;
- Measuring of abscissa and ordinate values as a difference between two marker positions;
- Storing of the result as a file in the part program directory. Thereafter, it is possible to export the file using either RCS802 or the CF card and to process the data in MS Excel.

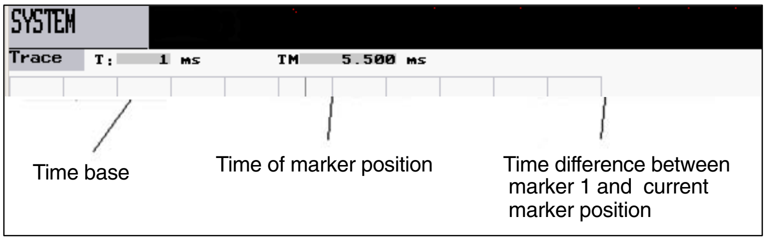


Trace T: 1 ms TM 5.500 ms

Fig. 7-17 The *Servo trace*start screen

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

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



SYSTEM

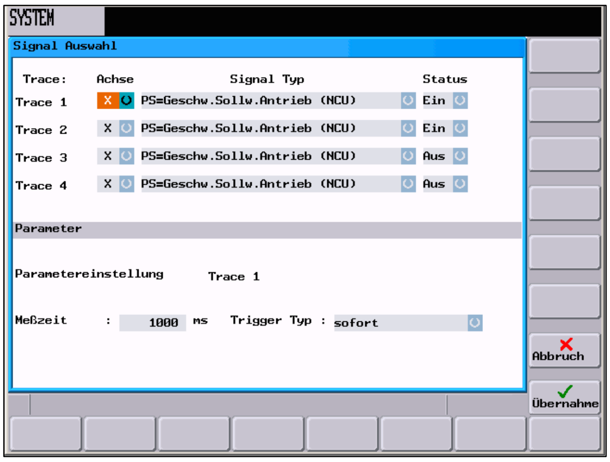
Trace T: 1 ms TM 5.500 ms

Time base
Time of marker position
Time difference between marker 1 and current marker position

Fig. 7-18 Meaning of the fields

Select signal

Use this menu to parameterize the measuring channel.



SYSTEM

Signal Auswahl

Trace:	Achse	Signal Typ	Status
Trace 1	X	PS=Geschw.Sollw.Antrieb (NCU)	Ein
Trace 2	X	PS=Geschw.Sollw.Antrieb (NCU)	Ein
Trace 3	X	PS=Geschw.Sollw.Antrieb (NCU)	Aus
Trace 4	X	PS=Geschw.Sollw.Antrieb (NCU)	Aus

Parameter

Parametereinstellung Trace 1

Meßzeit : 1000 ms Trigger Typ : sofort

Fig. 7-19

- **Selecting the axis:** To select the axis, use the “Axis” toggle field.
- **Signal type:**

  Following error

  System deviation
  Contour deviation
  Actual position value
  Speed actual value
  Speed setpoint
  Compensation value
  Set of parameters
  Controller input position setpoint
  Controller input speed setpoint
  Controller input acceleration setpoint
  Speed forward control value
  Exact stop fine signal
  Exact stop coarse signal
- **Status:** On The recording is performed in this channel.
  Off The channel is inactive.

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

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

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

V mark OFF

T mark OFF

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

Fix V mark

Fix T mark

Use the markers to determine the differences in the horizontal or vertical direction. To do this, position the marker on the start point and select the **<Fix**, **H mark>** or **<Fix T mark>** softkey. The difference between the starting point and the current marker position is now displayed in the status bar. The softkey labels will change to **<Free V mark >** or **<Free T mark>**.

Trace display

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

Time scale +

Time scale –

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

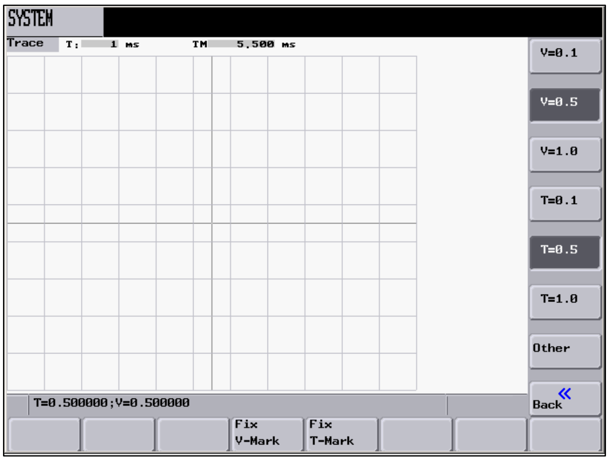
Vertical scale +

Vertical scale –

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

Marker steps

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



T: 1 ms TM 5.500 ms

T=0.500000;V=0.500000

Fig. 7-20

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

File

Use this softkey to save or load trace data.

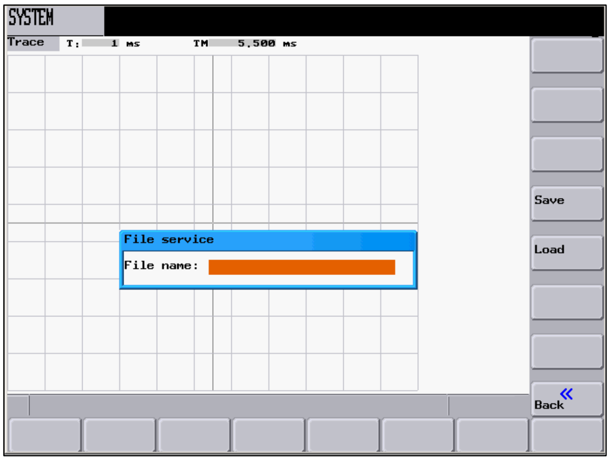


Fig. 7-21

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

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

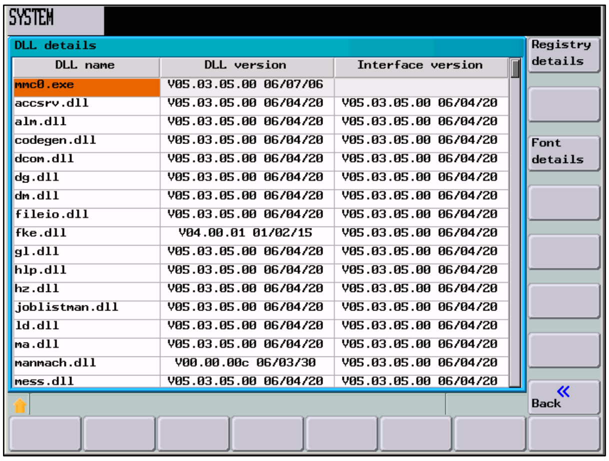
Use the **<Load** > softkey to load the specified file and to display the data graphically.

Version

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

HMI details

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



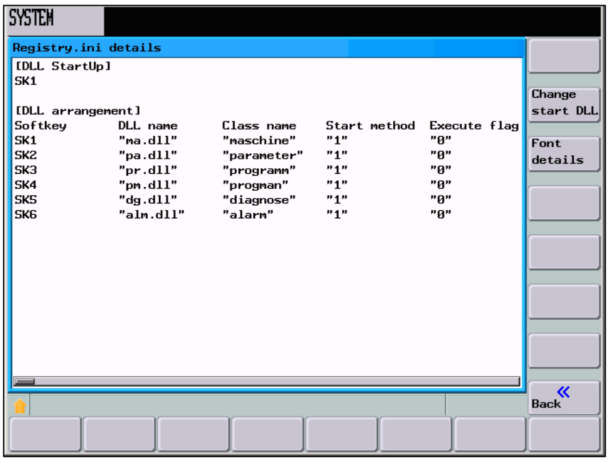
DLL details

DLL name	DLL version	Interface version
mmc0.exe	V05.03.05.00 06/07/06	
accsrv.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
alm.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
codegen.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
dcom.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
dg.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
dm.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
fileio.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
fke.dll	V04.00.01 01/02/15	V05.03.05.00 06/04/20
gl.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
hlp.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
hz.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
joblistman.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
ld.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
ma.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20
manmach.dll	V00.00.00c 06/03/30	V05.03.05.00 06/04/20
mess.dll	V05.03.05.00 06/04/20	V05.03.05.00 06/04/20

Fig. 7-22 The “HMI version” menu area

Registry details

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



Registry.ini details

```
[DLL StartUp]
SK1

[DLL arrangement]
Softkey        DLL name       Class name      Start method   Execute flag
SK1            "ma.dll"       "maschine"      "1"            "0"
SK2            "pa.dll"       "parameter"     "1"            "0"
SK3            "pr.dll"       "programm"      "1"            "0"
SK4            "pm.dll"       "progman"       "1"            "0"
SK5            "dg.dll"       "diagnose"      "1"            "0"
SK6            "alm.dll"      "alarm"         "1"            "0"
```

Fig. 7-23

Table 7-2 Meanings of the entries under [DLL arrangement]

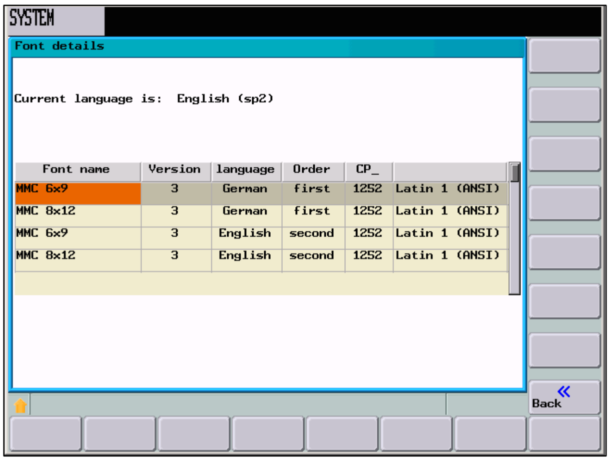
Description	Meaning
Softkey	SK1 to SK7 Hardkey assignment 1 to 7
DLL name	Name of the program to be executed
Class name	The identifier for receiving messages is defined in this column.
Start method	Number of the function executed after starting the program
Execute flag (kind of execution)	0 The program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

Table 7-2 Meanings of the entries under [DLL arrangement], continued

Description	Meaning
Password level	The execution of the program depends on the password level.
Class SK	reserved
SK file	reserved

Font details

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



Font details

Current language is: English (sp2)

Font name	Version	language	Order	CP_	
MMC 6x9	3	German	first	1252	Latin 1 (ANSI)
MMC 8x12	3	German	first	1252	Latin 1 (ANSI)
MMC 6x9	3	English	second	1252	Latin 1 (ANSI)
MMC 8x12	3	English	second	1252	Latin 1 (ANSI)

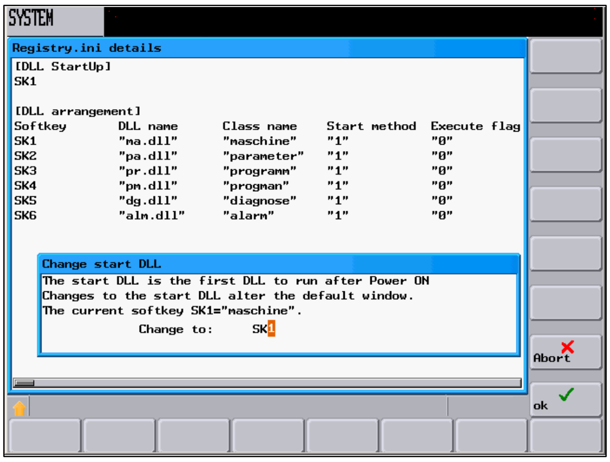
Fig. 7-24

Modifying the start DLL

Defining the start program

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

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



Registry.ini details

```
[DLL StartUp]
SK1

[DLL arrangement]
Softkey       DLL name      Class name     Start method   Execute flag
SK1           "ma.dll"      "maschine"     "1"            "0"
SK2           "pa.dll"      "parameter"    "1"            "0"
SK3           "pr.dll"      "programm"     "1"            "0"
SK4           "pm.dll"      "progman"      "1"            "0"
SK5           "dg.dll"      "diagnose"     "1"            "0"
SK6           "alm.dll"     "alarm"        "1"            "0"
```

Change start DLL

```
The start DLL is the first DLL to run after Power ON
Changes to the start DLL alter the default window.
The current softkey SK1="maschine".
             Change to:      SK1
```

Fig. 7-25 Modifying the startup DLL

PLC

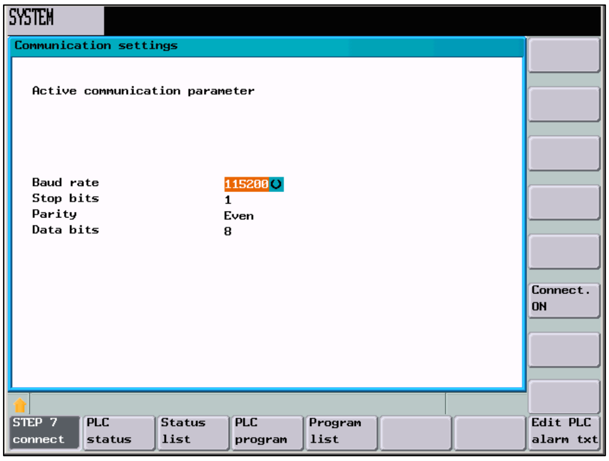
This softkey offers further functions for diagnostics and start–up of the PLC.

Connect to Step 7

Use this softkey to open the configuring dialog for the interface parameters for the STEP 7 connection.

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

The RS232 interface is initialized with activation of the connection.



Active communication parameter

Baud rate	115200
Stop bits	1
Parity	Even
Data bits	8

Fig. 7-26 Activating/deactivating RS232 for the Programming Tool

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

Connect on

Connect off

Use this softkey to activate the connection between the control system and the PC/PG. It is waited for the call of the Programming Tool. Any modifications to the settings are not possible in this status.
The softkey labeling changes to **<Connect off>**.
You can cancel the transmission from the control system at any point by selecting **<Connect off >** . Now it is possible again to make changes in the settings.

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

Press **<RECALL>** to quit the menu.

PLC status

Use this function to display and change the current states of the memory areas listed in Table 7-3.

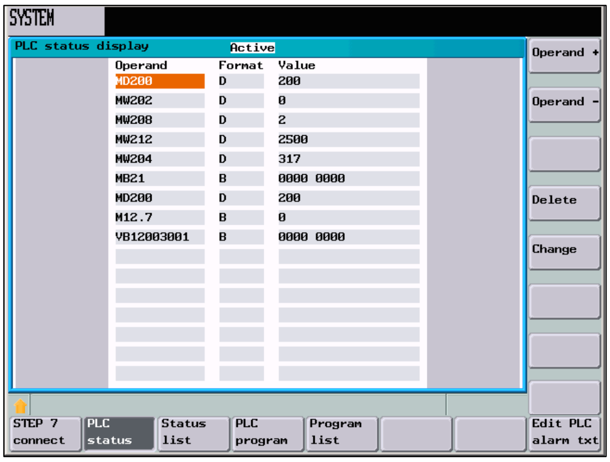
It is possible to display 16 operands at the same time.

Table 7-3 Memory areas

Inputs	I	Input byte (IBx), input word (Iwx), input double-word (IDx)
Outputs	Q	Output byte (Qbx), output word (Qwx), output double-word (QDx)
Bit memory	M	Flag byte (Mx), flag word (Mw), flag double-word (MDx)
Times	T	Time (Tx)

Table 7-3 Memory areas, continued

Counters	C	Counter (Cx)
Data	V	Data byte (Vbx), data word (Vwx), data double-word (VDx)
Format	B H D	Binary Hexadecimal decimal The binary representation is not possible with double words. Counters and timers are represented decimally.



Operand	Format	Value
MD200	D	200
MW202	D	0
MW208	D	2
MW212	D	2500
MW204	D	317
MB21	B	0000 0000
MD200	D	200
M12.7	B	0
VB12003001	B	0000 0000

Fig. 7-27 PLC status display

Operand +

The address of the operand is incremented each time by 1.

Operand –

The operand address displays the value decremented by 1.

Delete

Use this softkey to delete all operands.

Modify

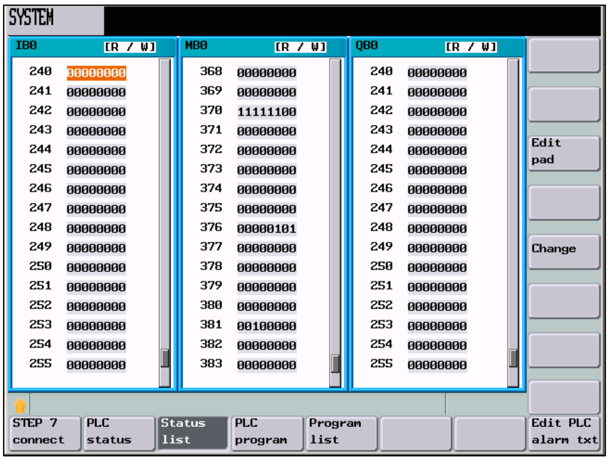
This softkey will cancel the cyclic update of the values. Then you can change the values of the operands.

Status list

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

There are 3 lists to choose from:

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



IB0	[R / W]	MB0	[R / W]	QB0	[R / W]
240	00000000	368	00000000	240	00000000
241	00000000	369	00000000	241	00000000
242	00000000	370	11111100	242	00000000
243	00000000	371	00000000	243	00000000
244	00000000	372	00000000	244	00000000
245	00000000	373	00000000	245	00000000
246	00000000	374	00000000	246	00000000
247	00000000	375	00000000	247	00000000
248	00000000	376	00000101	248	00000000
249	00000000	377	00000000	249	00000000
250	00000000	378	00000000	250	00000000
251	00000000	379	00000000	251	00000000
252	00000000	380	00000000	252	00000000
253	00000000	381	00100000	253	00000000
254	00000000	382	00000000	254	00000000
255	00000000	383	00000000	255	00000000

Fig. 7-28 The *PLC status list* start screen

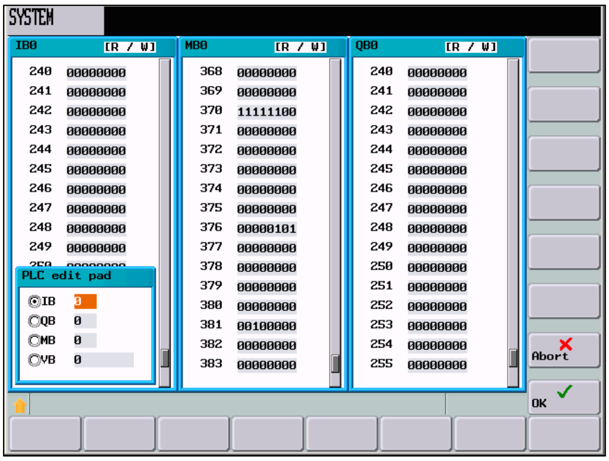
To change the settings, use the **<Edit pad> function.**

Modify

Use this softkey to change the value of the highlighted variable. Select the **<Accept >** softkey to confirm your changes.

Editing block

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



IB0	[R / W]	MB0	[R / W]	QB0	[R / W]
240	00000000	368	00000000	240	00000000
241	00000000	369	00000000	241	00000000
242	00000000	370	11111100	242	00000000
243	00000000	371	00000000	243	00000000
244	00000000	372	00000000	244	00000000
245	00000000	373	00000000	245	00000000
246	00000000	374	00000000	246	00000000
247	00000000	375	00000000	247	00000000
248	00000000	376	00000101	248	00000000
249	00000000	377	00000000	249	00000000
		378	00000000	250	00000000
		379	00000000	251	00000000
		380	00000000	252	00000000
		381	00100000	253	00000000
		382	00000000	254	00000000
		383	00000000	255	00000000

PLC edit pad

● IB 0
○ QB 0
○ MB 0
○ VB 0

Fig. 7-29 The “Data type” selection screenform

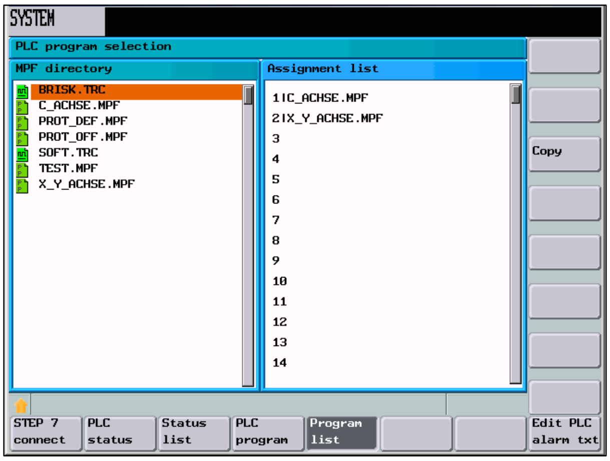
Use the cursor keys and the PageUp / PageDown keys to navigate in and between the columns.

PLC program

PLC diagnosis using a ladder diagram (see Section 7.3)

Program list

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



SYSTEM

PLC program selection

MPF directory

```
BRISK.TRC
C_ACHSE.MPF
PROT_DEF.MPF
PROT_OFF.MPF
SOFT.TRC
TEST.MPF
X_Y_ACHSE.MPF
```

Assignment list

```
1|C_ACHSE.MPF
2|X_Y_ACHSE.MPF
3
4
5
6
7
8
9
10
11
12
13
14
```

Fig. 7-30

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

Copy

... writes the selected file name to the clipboard

Paste

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

Delete

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

Structure of the reference list (PLCPROG.LST file)

It is divided into 3 areas:

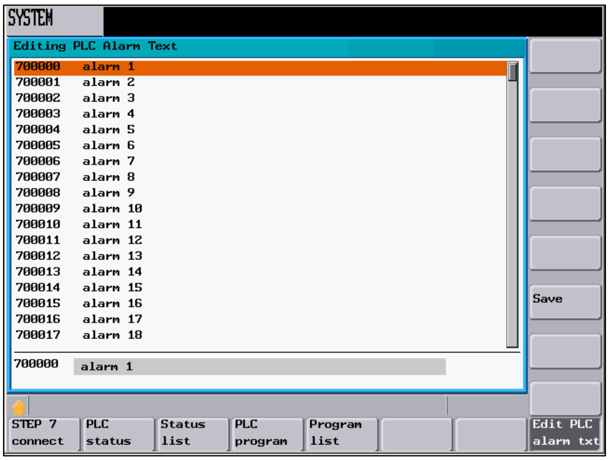
Number	Area	Protection level
1 to 100	User area	User
101 to 200	Machine manufacturer	Machine manufacturer
201 to 255	Siemens	Siemens

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

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

Edit PLC alarm txt

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



Editing PLC Alarm Text

```
700000   alarm 1
700001   alarm 2
700002   alarm 3
700003   alarm 4
700004   alarm 5
700005   alarm 6
700006   alarm 7
700007   alarm 8
700008   alarm 9
700009   alarm 10
700010   alarm 11
700011   alarm 12
700012   alarm 13
700013   alarm 14
700014   alarm 15
700015   alarm 16
700016   alarm 17
700017   alarm 18

700000   alarm 1
```

Fig. 7-31 Editing the PLC alarm text

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

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

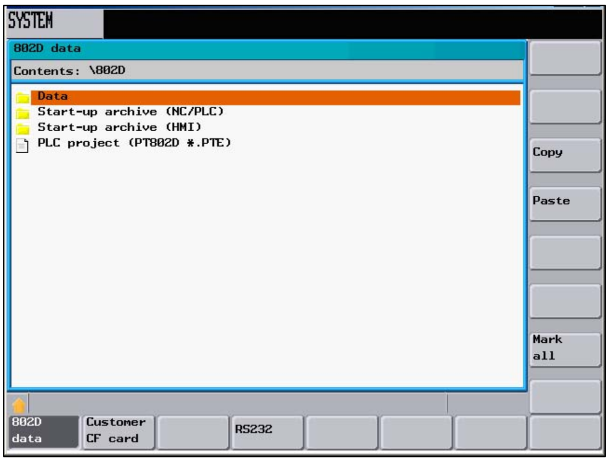
Start-up files

Use this function to create, read out/read in start-up archives and PLC projects (see also Section 7.1).
The window displays the contents of the selected drive in a tree structure. The horizontal softkeys display the available drives for selection in the form of a list. The vertical softkeys provide the control functions possible for the drive in question.

There are the following fixed default assignments:

• 802D data	Start-up data
• Customer CF card	Customer data on the CF card
• RS232	Serial interface

The data are handled using the “Copy & Paste” principle.



802D data

Contents: \802D

```
Data
Start-up archive (NC/PLC)
Start-up archive (HMI)
PLC project (PT802D *.PTE)
```

Fig. 7-32

802D data

The individual data groups in the “802D data” area have the following meanings:

- Data: Machine data
  Setting data
  Tool data
  R parameters
  Work offset
  Compensation: Leadscrew error (LSC)
  Global user data (user data)

  These data are special initialization data and are transferred in an ASCII file.

- Start-up archive (NC/PLC): NC data
  NC directories
  Display machine data
  Compensation: Leadscrew error
  PLC user alarm texts
  PLC project
  Drive machine data

  These data constitute a start-up file for NC and PLC data and are transferred in the binary format using the HMI archive format.

- Start-up archive (HMI) User cycles
  User directories
  Language files SP1
  Language files SP2
  Start screen
  Online help
  HMI bitmaps

  These data constitute a start-up file for HMI data and are transferred in the binary format using the HMI archive format.

- PLC project (PT802D *.PTE)

  A direct exchange between the control system and Programming Tool without conversion is possible by assisting the handling of a PLC project.

Customer CF card

Use this function to display the contents of the CF card. The following functions are available:

Rename

Use this function to rename a file selected beforehand using the cursor.

Create directory

Use this softkey to create a new directory on the CF card.

Copy

Use this softkey to copy one or several files to the clipboard.

Paste

Use this softkey to paste files or directories from the clipboard to the current directory.

Delete

Deletes the selected file name from the assignment list

Select All

Use this softkey to select all files for subsequent operations.

RS232

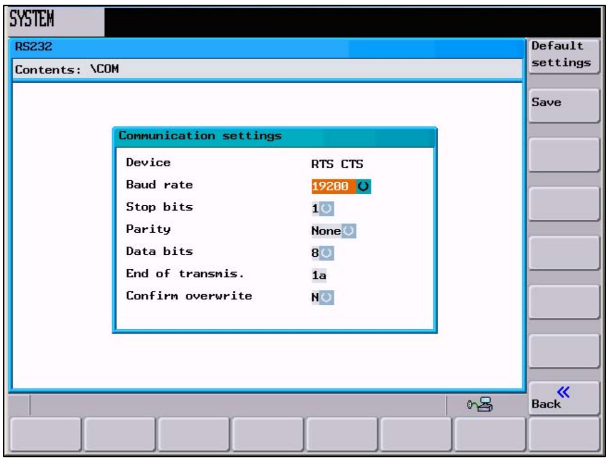
Use this softkey to read in / read out data via the RS232 interface.

Settings

Use this function to display and change the interface parameters. Any changes in the settings come into effect immediately.

Selecting the **<Save >** softkey will save the selected settings even beyond switching off.

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



Contents: \COM

Communication settings

Device	RTS CTS
Baud rate	19200
Stop bits	1
Parity	None
Data bits	8
End of transmis.	1a
Confirm overwrite	N

Fig. 7-33 Parameters of the RS232 interface

Interface parameters

Table 7-4 Interface parameters

Parameter	Description
Device type	**RTS/CTS** The signal RTS (Request to Send) controls the Send mode of the data transfer device. Active: Data are to be sent. Passive: The Send mode is only quit after all data have been transmitted. The CTS signal indicates the readiness to transmit data as the acknowledgment signal for RTS.
Baud rate	Setting the baud rate. 300 Baud 600 Baud 1,200 Baud 2,400 Baud 4,800 Baud 9,600 Baud 19,200 Baud 38,400 Baud 57,600 Baud 115,200 Baud
Stop bits	Number of stop bits with asynchronous transmission Input: 1 stop bit (default setting)2 stop bits

Table 7-4 Interface parameters, continued

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

7.1 Creating / reading in / reading out a start-up archive

Note for the Reader

/BA1/ SINUMERIK 802D sl “Instruction Manual”, Section “Data backup and machine series start-up”

Operating sequence

SHIFT SYSTEM ALARM

Start-up files

Select the **<Start-up files>** softkey which is to be found in the “System” operating area.

Creating a start-up archive

A start-up archive can be created either with all components or with some selected components.

To create an archive with selected components, the following operator actions are required:

802D data

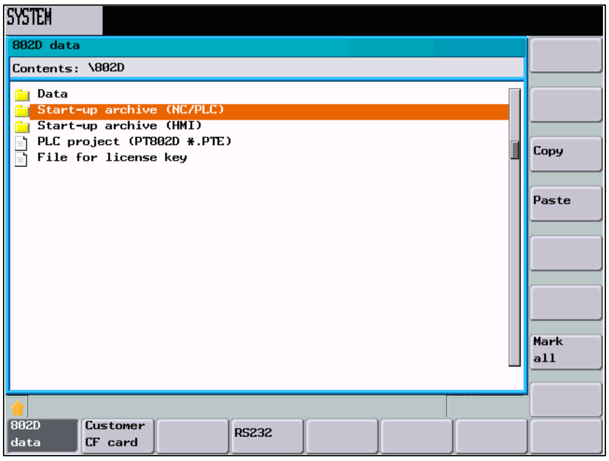
Select **<802D data>**. Use the direction keys to select the **Start-up archive (NC/PLC)** line.

INPUT

Press ENTER to open the directory and select the desired files using the cursor keys.

Copy

Select the **<Copy> softkey.** Files are copied to the clipboard.



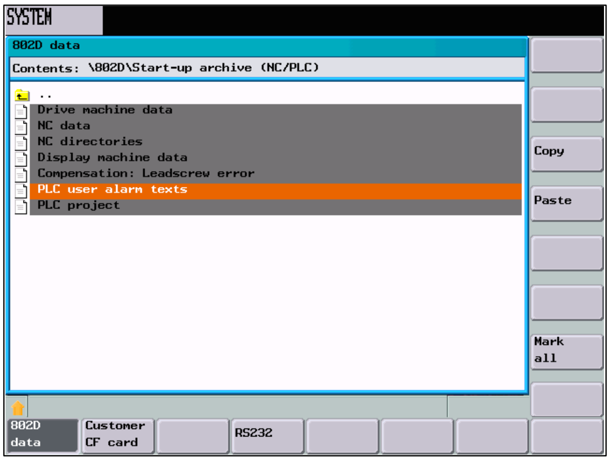
SYSTEM

802D data

Contents: \802D

- Data
- Start-up archive (NC/PLC)
- Start-up archive (HMI)
- PLC project (PT802D *.PTE)
- File for license key

Fig. 7-34 Copying a complete start-up archive



802D data

Contents: \802D\Start-up archive (NC/PLC)

```
..
Drive machine data
NC data
NC directories
Display machine data
Compensation: Leadscrew error
PLC user alarm texts
PLC project
```

Fig. 7-35 Contents of the start-up backup

Writing the start-up archive to a CompactFlash card

Prerequisite: The CF card is inserted, and the start-up archive has been copied to the clipboard.

Operating sequence:

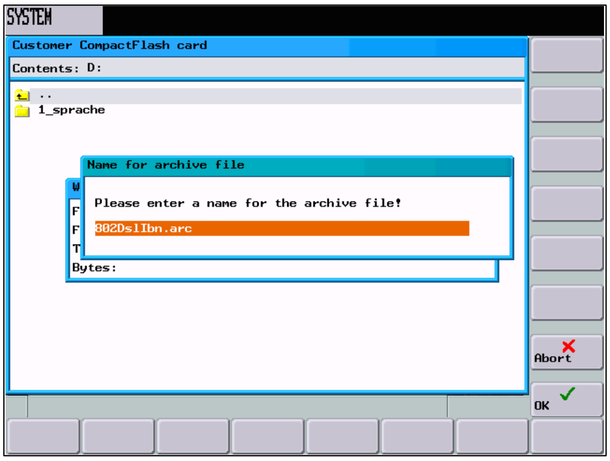
Customer CF card

Select the **<Customer CF card> softkey.** Select the archiving location from the directory.

Paste

Select the **<Paste>** softkey to start writing of the start-up archive.

In the dialog which is now displayed, either confirm the name offered or type a new name. Clicking on **<OK>** quits the dialog box.



Customer CompactFlash card

Contents: D:

```
..
1_sprache
```

Name for archive file

Please enter a name for the archive file!

802DslIbn.arc

Bytes:

Fig. 7-36

Reading in the start-up archive from the CompactFlash card

To import a start-up archive, perform the following operator actions:

1. Insert the CF card.
2. Select the **<Customer CF card>** softkey and select the line with the desired archive file.
3. Use **<Copy>** to copy the file to the clipboard.
4. Select the **<802D data>** softkey and place the cursor on the **Start-up archive (NC/PLC)** line.
5. Select **<Paste>** to start the start-up.
6. Acknowledge the start dialog on the control system.

7.2 Reading in / reading out PLC projects

When reading in a project, this will be transferred to the file system of the PLC and then activated. To complete the activation, the control system is restarted (warm start).

Reading in the start-up archive from the CompactFlash card

To import a PLC project, perform the following operator actions:

1. Insert the CF card.
2. Select the **<Customer CF card>** softkey and select the line with the desired project file in the PTE format.
3. Use **<Copy>** to copy the file to the clipboard.
4. Select the **<802D data>** softkey and position the cursor on the **PLC project (PT802D *.PTE)** line.
5. Select the **<Paste>** softkey to start reading in and archiving.

Writing the project to the CompactFlash card

Perform the following operator actions:

1. Insert the CF card.
2. Select the **<802D data>** softkey and position the direction keys on the **PLC project (PT802D *.PTE)** line.
3. Use **<Copy>** to copy the file to the clipboard.
4. Select the **<Customer CF card>** softkey and select the archiving location for the file.
5. Select the **<Paste>** softkey to start writing.

7.3 PLC diagnosis represented as a ladder diagram

Functionality

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

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

Note

It is not possible here to edit the program.

Operating sequence

SHIFT SYSTEM ALARM

PLC

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

PLC program

The project stored in the permanent memory is opened.

7.3.1 Screen layout

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

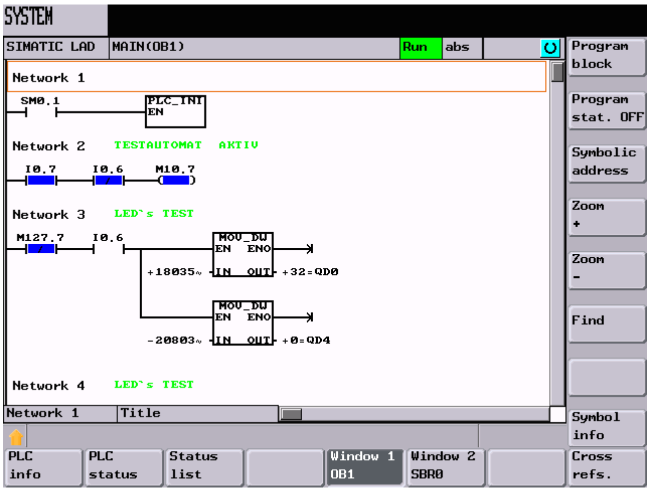


Fig. 7-37 Screen layout

Screen control	Display	Meaning
①	**Application area**	
②	**Supported PLC program language**	
③	**Name of the active program block** Representation: Symbolic name (absolute name)	
④	**Program status**	
	RUN	Program is running
	STOP	Program stopped
	Status of the application area	
	Sym	Symbolic representation
	abs	Absolute representation
⑤	⇨↻	**Display of the active keys**
⑥	**Focus** Performs the tasks of the cursor	
⑦	**Tip line** Contains notes for searching	

7.3.2 Operating options

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

Hotkeys

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

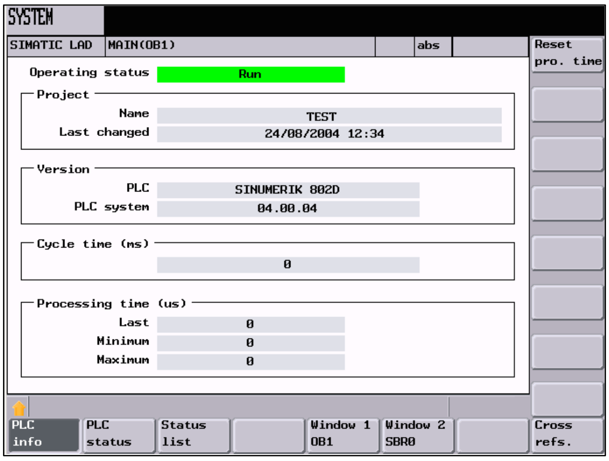
Table 7-5 Key combinations

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

Softkeys

PLC info

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



Operating status: Run

Project
Name: TEST
Last changed: 24/08/2004 12:34

Version
PLC: SINUMERIK 802D
PLC system: 04.00.04

Cycle time (ms)
0

Processing time (us)
Last: 0
Minimum: 0
Maximum: 0

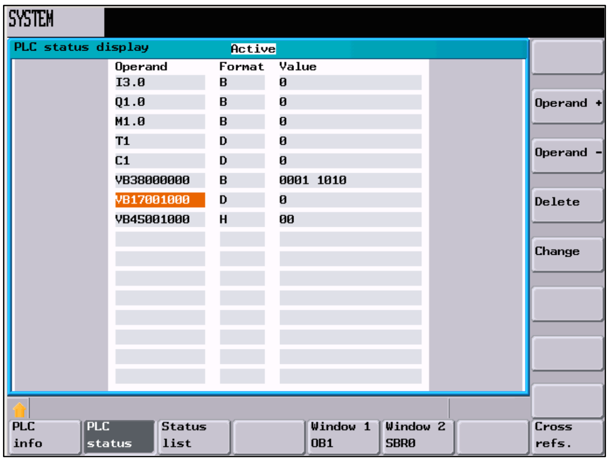
Fig. 7-38 PLC info

Reset pro. time

Use this softkey to refresh the data in the window.

PLC status

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

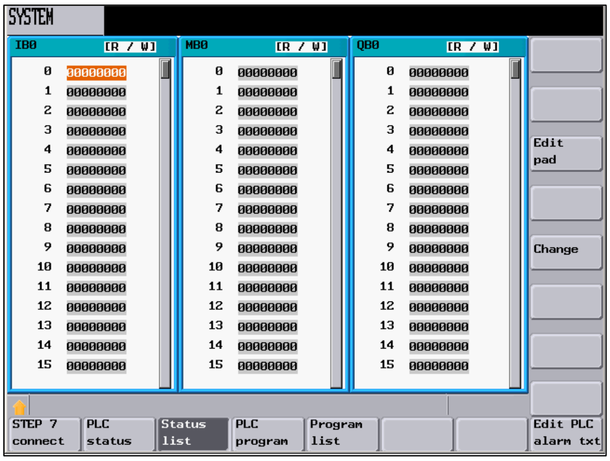


Operand	Format	Value
I3.0	B	0
Q1.0	B	0
M1.0	B	0
T1	D	0
C1	D	0
VB38000000	B	0001 1010
VB17001000	D	0
VB45001000	H	00

Fig. 7-39 PLC status display

Status list

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



IB0	[R / W]	MB0	[R / W]	QB0	[R / W]
0	00000000	0	00000000	0	00000000
1	00000000	1	00000000	1	00000000
2	00000000	2	00000000	2	00000000
3	00000000	3	00000000	3	00000000
4	00000000	4	00000000	4	00000000
5	00000000	5	00000000	5	00000000
6	00000000	6	00000000	6	00000000
7	00000000	7	00000000	7	00000000
8	00000000	8	00000000	8	00000000
9	00000000	9	00000000	9	00000000
10	00000000	10	00000000	10	00000000
11	00000000	11	00000000	11	00000000
12	00000000	12	00000000	12	00000000
13	00000000	13	00000000	13	00000000
14	00000000	14	00000000	14	00000000
15	00000000	15	00000000	15	00000000

Fig. 7-40 Status list

Window 1
xxxx

Window2
xxxx

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

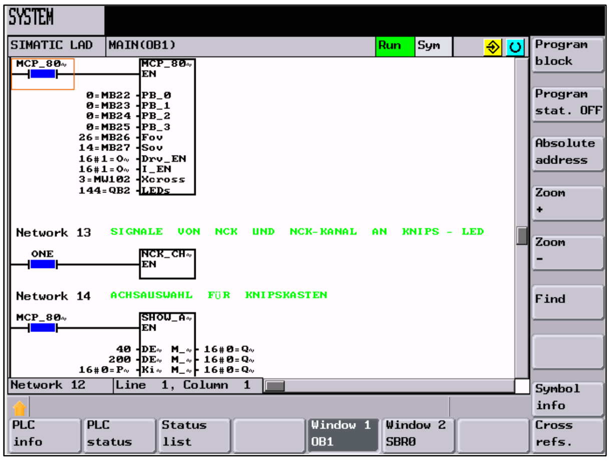
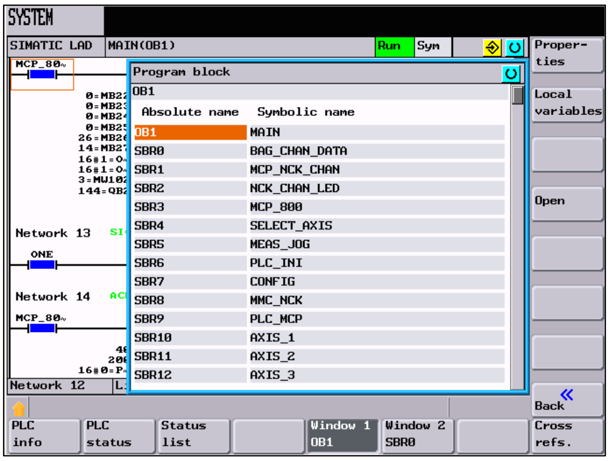


Fig. 7-41 Window 1

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

Program
block

This softkey can be used to display the list of the PLC program blocks. Use the direction or the PageUp/PageDown keys to select the PLC program block to be opened. The current program block is displayed in the Info line of the list box.



Program block

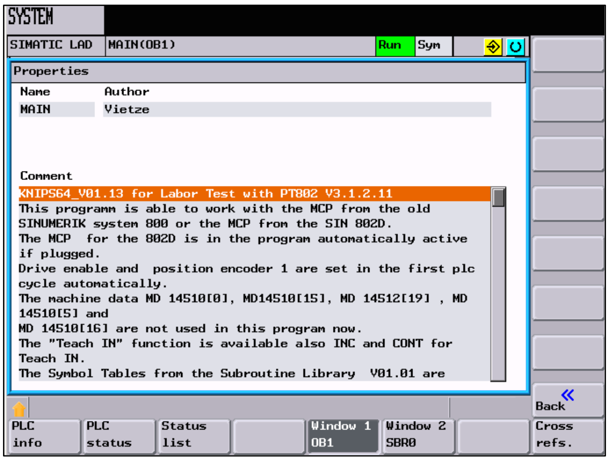
OB1

Absolute name	Symbolic name
OB1	MAIN
SBR0	BAG_CHAN_DATA
SBR1	MCP_NCK_CHAN
SBR2	NCK_CHAN_LED
SBR3	MCP_800
SBR4	SELECT_AXIS
SBR5	MEAS_JOG
SBR6	PLC_INI
SBR7	CONFIG
SBR8	MMC_NCK
SBR9	PLC_MCP
SBR10	AXIS_1
SBR11	AXIS_2
SBR12	AXIS_3

Fig. 7-42 PLC block selection

Properties

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



Properties

Name	Author
MAIN	Vietze

Comment

```
KNIPS64_V01.13 for Labor Test with PT802 V3.1.2.11
This programm is able to work with the MCP from the old
SINUMERIK system 800 or the MCP from the SIN 802D.
The MCP  for the 802D is in the program automatically active
if plugged.
Drive enable and  position encoder 1 are set in the first plc
cycle automatically.
The machine data MD 14510[0], MD14510[15], MD 14512[19] , MD
14510[5] and
MD 14510[16] are not used in this program now.
The "Teach IN" function is available also INC and CONT for
Teach IN.
The Symbol Tables from the Subroutine Library  V01.01 are
```

Fig. 7-43 Properties of the selected PLC program block

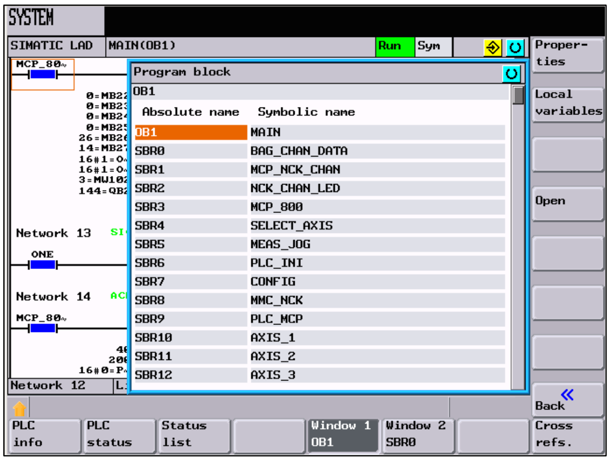
Local variables

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

There are two types of program blocks.

- OB1 Only temporary local variable
- SBRxx Temporary local variable

A table of variables exists for each program block.



SIMATIC LAD | MAIN(OB1) | Run | Sym

Program block

OB1

Absolute name	Symbolic name
OB1	MAIN
SBR0	BAG_CHAN_DATA
SBR1	MCP_NCK_CHAN
SBR2	NCK_CHAN_LED
SBR3	MCP_800
SBR4	SELECT_AXIS
SBR5	MEAS_JOG
SBR6	PLC_INI
SBR7	CONFIG
SBR8	MMC_NCK
SBR9	PLC_MCP
SBR10	AXIS_1
SBR11	AXIS_2
SBR12	AXIS_3

Fig. 7-44 Table of local variables for the selected program block

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

Protection

For protected program blocks, it is possible to enter the password assigned in the PLC802 Programming Tool. Subsequently, the relevant program block is released for display in the ladder diagram.

Open

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

Program stat. ON

Program stat. OFF

Use this softkey to activate / deactivate the program status display. Here you can monitor the current states of the network from the PLC cycle end. The states of all operands are displayed in the “Program status” ladder diagram. This LAD acquires the values for the status display in several PLC cycles and then refreshes the status display.

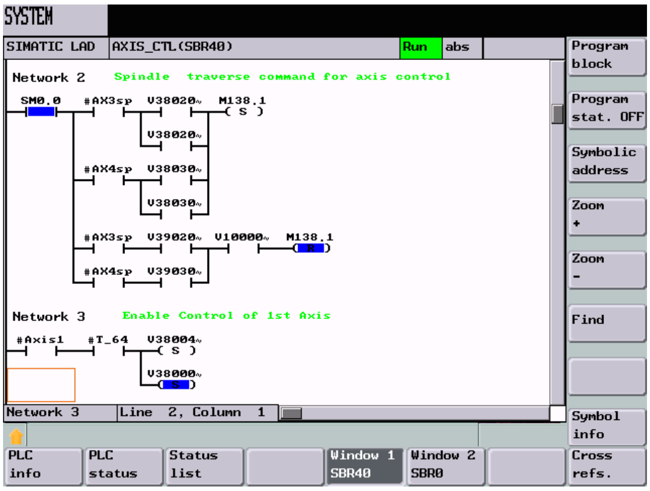


Fig. 7-45 “Program status” ON – absolute representation

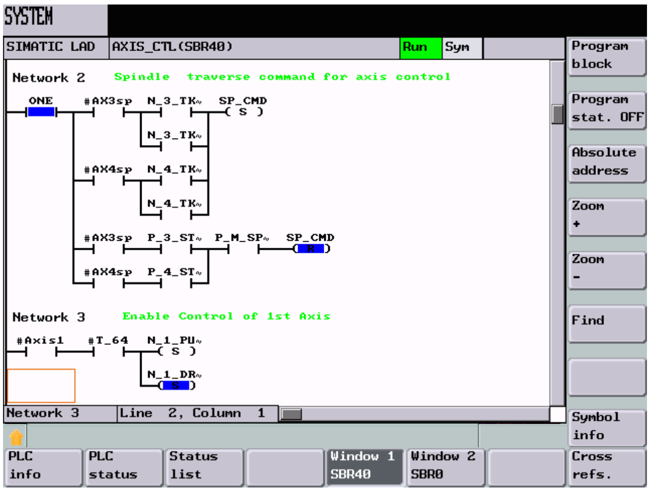


Fig. 7-46 “Program status” ON – symbolic representation

Symbol. address

Absolute address

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

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

Zoom +

Zoom –

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

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

Find

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

A dialog box is displayed from which various search criteria can be selected. The **<Absolute/symbol. address>** softkey can be used to search for a certain operand matching this criterion in both PLC windows. When searching, uppercase and lowercase letters are ignored.

Selection in the upper toggle field:

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

Further search criteria:

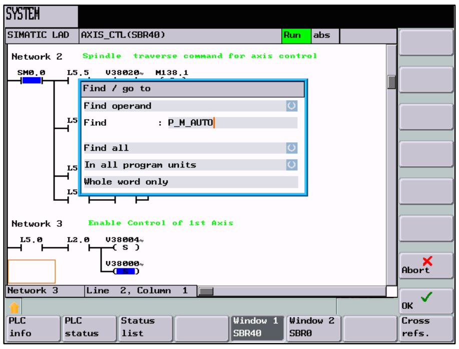
- Search direction down (from the current cursor position)
- Whole program block (from the beginning)
- In one program block
- Over all program blocks

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

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

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

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



Find / go to

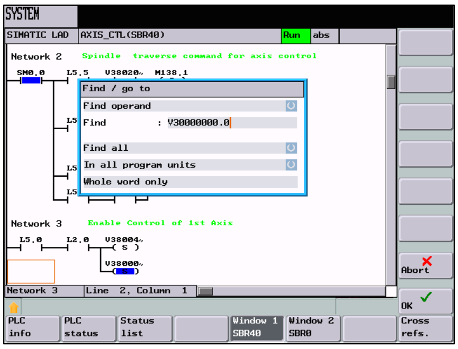
Find operand

Find : P_M_AUTO

Find all

In all program units

Whole word only



Find / go to

Find operand

Find : V30000000.0

Find all

In all program units

Whole word only

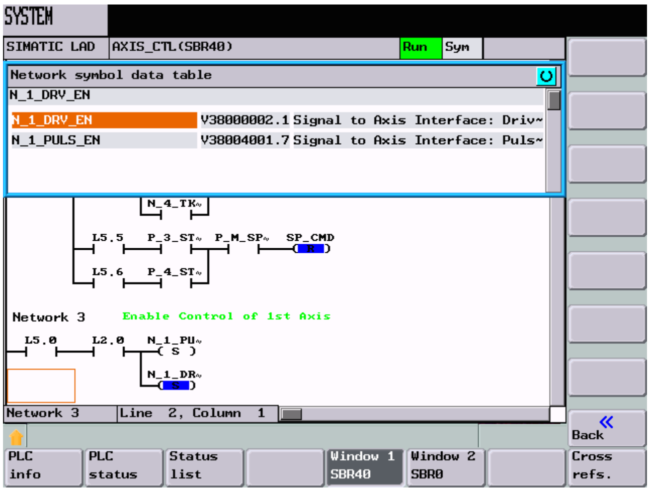
Fig. 7-47 Searching for symbolic operands

Searching for absolute operands

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

Symbol info

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



Network symbol data table

N_1_DRV_EN

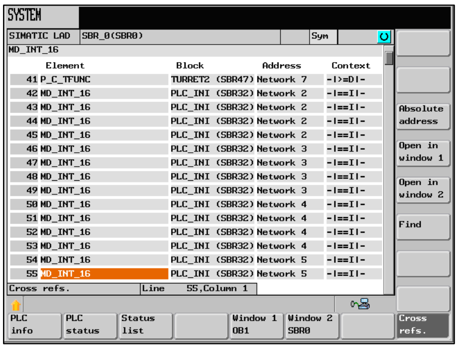
N_1_DRV_EN	V38000002.1	Signal to Axis Interface: Driv~
N_1_PULS_EN	V38004001.7	Signal to Axis Interface: Puls~

Fig. 7-48 Network symbolic

Cross reference

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

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

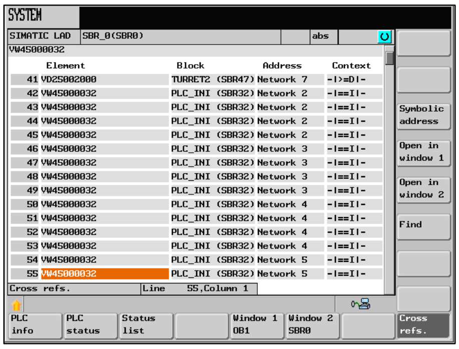


SIMATIC LAD | SBR_0(SBR0) | Sym

MD_INT_16

	Element	Block	Address	Context
41	P_C_TFUNC	TURRET2 (SBR47)	Network 7	-|>=D|-
42	MD_INT_16	PLC_INI (SBR32)	Network 2	-|==I|-
43	MD_INT_16	PLC_INI (SBR32)	Network 2	-|==I|-
44	MD_INT_16	PLC_INI (SBR32)	Network 2	-|==I|-
45	MD_INT_16	PLC_INI (SBR32)	Network 2	-|==I|-
46	MD_INT_16	PLC_INI (SBR32)	Network 3	-|==I|-
47	MD_INT_16	PLC_INI (SBR32)	Network 3	-|==I|-
48	MD_INT_16	PLC_INI (SBR32)	Network 3	-|==I|-
49	MD_INT_16	PLC_INI (SBR32)	Network 3	-|==I|-
50	MD_INT_16	PLC_INI (SBR32)	Network 4	-|==I|-
51	MD_INT_16	PLC_INI (SBR32)	Network 4	-|==I|-
52	MD_INT_16	PLC_INI (SBR32)	Network 4	-|==I|-
53	MD_INT_16	PLC_INI (SBR32)	Network 4	-|==I|-
54	MD_INT_16	PLC_INI (SBR32)	Network 5	-|==I|-
55	MD_INT_16	PLC_INI (SBR32)	Network 5	-|==I|-

Cross refs. | Line 55,Column 1



SIMATIC LAD | SBR_0(SBR0) | abs

VW45000032

	Element	Block	Address	Context
41	VD25002000	TURRET2 (SBR47)	Network 7	-|>=D|-
42	VW45000032	PLC_INI (SBR32)	Network 2	-|==I|-
43	VW45000032	PLC_INI (SBR32)	Network 2	-|==I|-
44	VW45000032	PLC_INI (SBR32)	Network 2	-|==I|-
45	VW45000032	PLC_INI (SBR32)	Network 2	-|==I|-
46	VW45000032	PLC_INI (SBR32)	Network 3	-|==I|-
47	VW45000032	PLC_INI (SBR32)	Network 3	-|==I|-
48	VW45000032	PLC_INI (SBR32)	Network 3	-|==I|-
49	VW45000032	PLC_INI (SBR32)	Network 3	-|==I|-
50	VW45000032	PLC_INI (SBR32)	Network 4	-|==I|-
51	VW45000032	PLC_INI (SBR32)	Network 4	-|==I|-
52	VW45000032	PLC_INI (SBR32)	Network 4	-|==I|-
53	VW45000032	PLC_INI (SBR32)	Network 4	-|==I|-
54	VW45000032	PLC_INI (SBR32)	Network 5	-|==I|-
55	VW45000032	PLC_INI (SBR32)	Network 5	-|==I|-

Cross refs. | Line 55,Column 1

Fig. 7-49 The “Cross references” main menu (symbolic)

(absolute)

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

Symbol. address

Absolute address

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

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

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

Open in window 1

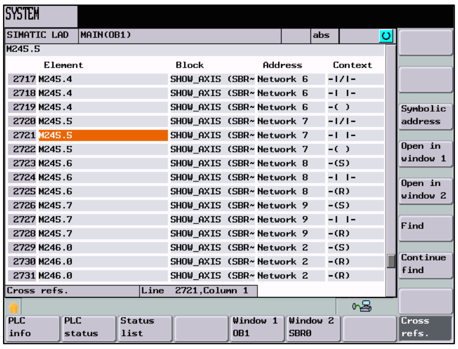
Open in window 2

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

Example:

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

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

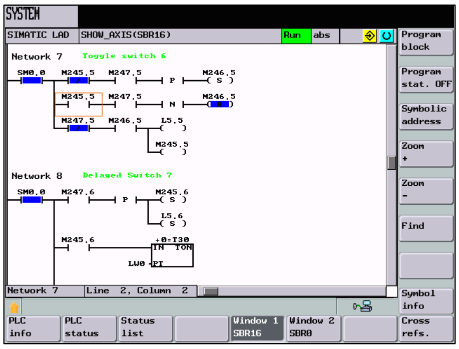


SIMATIC LAD | MAIN(OB1) | abs

M245.5

	Element	Block	Address	Context
2717	M245.4	SHOW_AXIS (SBR~	Network 6	-|/|-
2718	M245.4	SHOW_AXIS (SBR~	Network 6	-| |-
2719	M245.4	SHOW_AXIS (SBR~	Network 6	-( )
2720	M245.5	SHOW_AXIS (SBR~	Network 7	-|/|-
2721	M245.5	SHOW_AXIS (SBR~	Network 7	-| |-
2722	M245.5	SHOW_AXIS (SBR~	Network 7	-( )
2723	M245.6	SHOW_AXIS (SBR~	Network 8	-(S)
2724	M245.6	SHOW_AXIS (SBR~	Network 8	-| |-
2725	M245.6	SHOW_AXIS (SBR~	Network 8	-(R)
2726	M245.7	SHOW_AXIS (SBR~	Network 9	-(S)
2727	M245.7	SHOW_AXIS (SBR~	Network 9	-| |-
2728	M245.7	SHOW_AXIS (SBR~	Network 9	-(R)
2729	M246.0	SHOW_AXIS (SBR~	Network 2	-(S)
2730	M246.0	SHOW_AXIS (SBR~	Network 2	-(R)
2731	M246.0	SHOW_AXIS (SBR~	Network 2	-(R)

Cross refs. | Line 2721,Column 1



SIMATIC LAD | SHOW_AXIS(SBR16) | Run | abs

Network 7 | Line 2, Column 2

Fig. 7-50 Cursor “M251.0 in OB1 network 2)

M251.0 in OB1 network 2 in window 1

Find

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

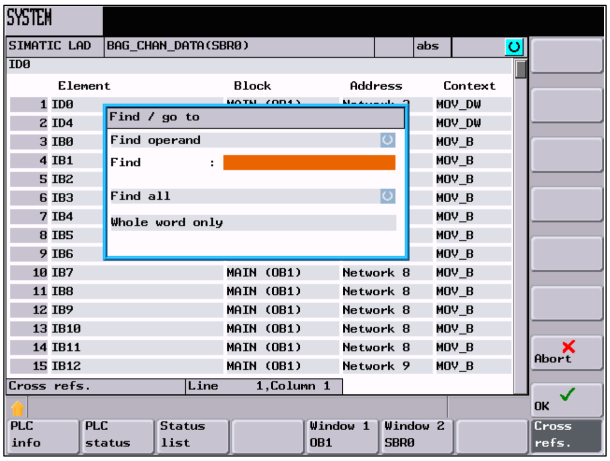
You can search for the operands as whole words (identifiers). When searching, uppercase and lowercase letters are ignored.

Search options:

- Search for absolute and symbolic operands
- Go to line

Search criteria:

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



SIMATIC LAD BAG_CHAN_DATA(SBR0)

ID0

	Element	Block	Address	Context
1	ID0	MAIN (OB1)	[illegible]	MOV_DW
2	ID4			MOV_DW
3	IB0			MOV_B
4	IB1			MOV_B
5	IB2			MOV_B
6	IB3			MOV_B
7	IB4			MOV_B
8	IB5			MOV_B
9	IB6			MOV_B
10	IB7	MAIN (OB1)	Network 8	MOV_B
11	IB8	MAIN (OB1)	Network 8	MOV_B
12	IB9	MAIN (OB1)	Network 8	MOV_B
13	IB10	MAIN (OB1)	Network 8	MOV_B
14	IB11	MAIN (OB1)	Network 8	MOV_B
15	IB12	MAIN (OB1)	Network 9	MOV_B

Find / go to

Find operand

Find :

Find all

Whole word only

Cross refs. Line 1,Column 1

Fig. 7-51 Searching for operands in cross references

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

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

7.4 Alarm display

Operating sequence

SYSTEM ALARM

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

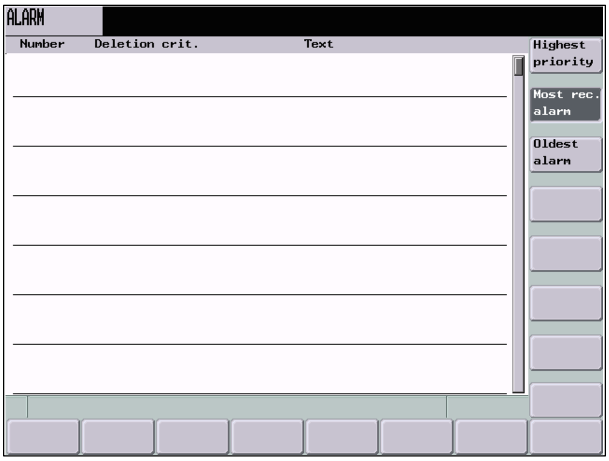


Fig. 7-52 Alarm window

Softkeys

Highest priority

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

Most recent alarm

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

Oldest alarm

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

Notes

Programming 8

8.1 Fundamental principles of NC programming

8.1.1 Program names

Each program has its own program name. The name can be freely chosen during program creation, taking the following conventions into account:

- The first two characters must be letters;
- Use only letters, digits or underscore.
- Do not use delimiters (see Section “Character set”).
- The decimal point must only be used for separation of the file extension.
- Do not use more than 25 characters.

Example: **FRAME52**

8.1.2 Program structure

Structure and contents

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

Each block represents a machining step.

Instructions are written in the blocks in the form of **words**.

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

Table 8-1 NC program structure

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

8.1.3 Word structure and address

Functionality/structure

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

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

  A positive sign (+) can be omitted.

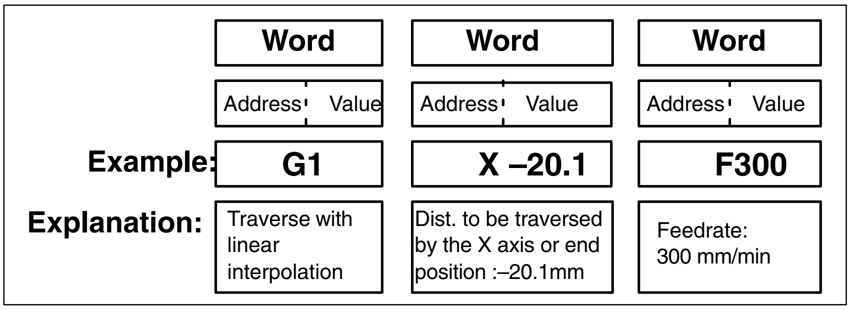


Fig. 8-1 Word structure (example)

Several address characters

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

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

Extended address

With the addresses

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

the address is extended by 1 to 4 figures to obtain a greater number of addresses. In this case, the value must be assigned using an equality sign "=" (see also Section "List of instructions").
Examples: **R10=6.234 H5=12.1 I1=32.67**

8.1.4 Block structure

Functionality

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

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

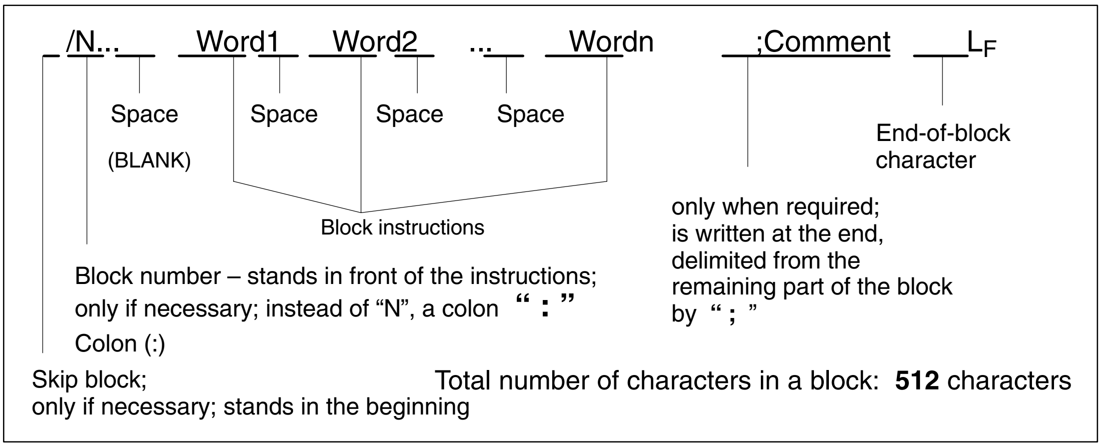

_ /N...__ _Word1__ _Word2__ ...__ _Wordn __;Comment ___LF
Space
(BLANK)
Space
Space
Space
End-of-block character
Block instructions
only when required; is written at the end, delimited from the remaining part of the block by " ; "
Block number – stands in front of the instructions; only if necessary; instead of "N", a colon " : "
Colon (:)
Skip block; only if necessary; stands in the beginning
Total number of characters in a block: 512 characters

Fig. 8-2 Block structure diagram

Word order

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

Note regarding block numbers

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

Block skip

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

Comment, remark

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

Messages

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

Programming example

```
N10                                 ; G&S company Order no. 12A71
N20                                 ; Pump part 17, Drawing no.: 123 677
N30                                 ; Program created by H. Adam, Dept. TV 4
N40 MSG(“DRAWING NO.: 123677”)
:50  G17 G54 G94 F470 S20 D2 M3     ; Main block
N60 G0 G90 X100 Y200
N70 G1 Y185.6
N80 X112
/N90 X118 Y180                      ; Block can be skipped
N100 X118 Y120
N110 G0 G90 X200
N120 M2                             ; End of program
```

8.1.5 Character set

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

Letters, digits

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

Printable special characters

(	Left round bracket	“	Inverted commas
)	Right round bracket	_	Underscore (belonging to a letter)
[	Left square bracket	.	Decimal point
]	Right square bracket	,	Comma, delimiter

<	Less than
>	Greater than
:	Main block, completion of label
=	Assignment; part of equality
/	Division; Block skip
*	Multiplication
+	Addition; positives sign
–	Subtraction; negative sign
;	Start of a comment
%	Reserved; do not use
&	Reserved; do not use
’	Reserved; do not use
$	System-internal variable identifier
?	Reserved; do not use
!	Reserved; do not use

Non-printable special characters

L_F	End-of-block characters
Blank	delimiter between the words, blank
Tabulator	Reserved; do not use

8.1.6 List of instructions

Functions available with SINUMERIK 802D sl plus and pro

Address	Meaning	Value assignments	Information	Programming
D	Tool offset number	0 ... 0 ... 9, only integer, no sign	Contains offset data for a certain tool T... ; D0–>offset values= 0, max. 9 D numbers per tool	D...
F	Feed	0.001 ... 99 999.999	Path velocity of a tool/workpiece; unit: mm/min or mm/revolution depending on G94 or G95	F...
F	Dwell time in block with G4	0.001 ... 99 999.999	Dwell time in seconds	G4 F... ; separate block
G	G function (preparatory function)	Only integer, specified values	The G functions are divided into G groups. Only one G function of a group can be programmed in a block. A G function can be either modal (until it is canceled by another function of the same group) or only effective for the block in which it is programmed (non-modal).	G... or symbolic name, e.g.: CIP
G0	Linear interpolation at rapid traverse rate		**G group 1**: Motion commands (interpolation type) modally effective	G0 X... Y... ; Cartesian In polar coordinates: G0 AP=... RP=... or with additional axis: G0 AP=... RP=... C... ; e.g.: With G17: C axis
G1 *	Linear interpolation at feedrate			G1 X... Y... C... F... In polar coordinates: G1 AP=... RP=... F... or with additional axis: G1 AP=... RP=... F... ; e.g.: With G17: C axis
G2	Circular interpolation CW (in conjunction with a 3rd axis and TURN=... also helix interpolation –>see under TURN)			G2 X... Y... I... J... F... ; Center and end points G2 X... Y... CR=... F... ; Radius and end point G2 AR=... I... J... F... ; Aperture angle and center point G2 X... Y... I... J... F... ; Aperture angle and end point In polar coordinates: G2 AP=... RP=... F... or with additional axis: G2 AP=... RP=... C... F... ; e.g.: With G17: C axis
G3	Circular interpolation CCW			G3 .... ; Otherwise, as with G2
CIP	Circular interpolation through intermediate point			CIP X... Y... I1=... J1=... F...

CT	Circular interpolation; tangential transition		N10 ... N20 CT X... Y... F... ; Circle; tangential transition to the previous path segment
G4	Dwell time	**G group 2:** Special movements, non-modal	G4 F... ; Separate block, F: Time in seconds or G4 S.... ; Separate block, S: in spindle revolutions
G74	Approaching a reference point		G74 X1=0 Y1=0 Z1=0 ; Separate block, (machine axis identifier!)
G75	Approaching a fixed point		G75 X1=0 Y1=0 Z1=0 ; Separate block, (machine axis identifier!)
TRANS	Programmable offset	**G group 3:** Write memory non-modal	TRANS X... Y... ; Separate block
ROT	Programmable rotation		ROT RPL=... ; Rotation in the current plane G17 ... G19, separate block
SCALE	Programmable scaling factor		SCALE X... Y... ; Scaling factor in the direction of the specified axis, separate block
MIRROR	Programmable mirroring		MIRROR X0 ; coordinate axis whose direction is changed; separate block
ATRANS	Additive programmable offset		ATRANS X... Y... ; Separate block
AROT	Additive programmable rotation		AROT RPL=... ; Additive rotation in the current plane G17, separate block
ASCALE	Additive programmable scaling factor		ASCALE X... Y... ; Scaling factor in the direction of the specified axis, separate block
AMIRROR	Additive programmable mirroring		AMIRROR X0 ; Coordinate axis whose direction is changed; separate block
G110	Pole specification relative to the last programmed setpoint position		G110 X... Y... ; Pole specification, Cartesian, e.g.: With G17 G110 RP=... AP=... ; Pole specification, polar, separate block
G111	Pole specification relative to origin of current workpiece coordinate system		G111 X... Y... ; Pole specification, Cartesian, e.g.: With G17 G111 RP=... AP=... ; Pole specification, polar, separate block
G112	Pole specification, relative to the last valid POLE		G112 X... Y... ; Pole specification, Cartesian, e.g.: With G17 G112 RP=... AP=... ; Pole specification, polar, separate block

G17 *	X/Y plane	**G group 6:** Plane selection	G17	; Vertical axis on this plane is the tool length compensation axis
G500 *	Settable work offset OFF	**G group 8:** Settable work offset modally effective		
G54 *	1. settable zero offset			
G55	2nd settable work offset			
G56	3rd settable work offset			
G57	4th settable work offset			
G58	5th settable work offset			
G59	6th settable work offset			
G53	Non-modal skipping of the settable work offset	**G group 9:** Skip settable work offset non-modal		
G153	Non-modal skipping of the settable work offset including base frame			
G60 *	Exact stop	**G group 10:** Approach behavior modally effective		
G64	Continuous-path control mode			
G9	Non-modal exact stop	**G group 11:** Exact stop, non-modal		
G601 *	Exact stop window, fine, with G60, G9	**G group 12:** Exact-stop window modally effective		
G602	Exact stop window, coarse, with G60, G9			
G603	Exact stop at end of interpolation			
G70	Inch dimension input	**G group 13:** Inch / metr. dimension input modally effective		
G71 *	Metric dimension data input			
G700	Inch dimension data input; also for feedrate F			
G710	Metric dimension data input; also for feedrate F			
G90 *	Absolute dimension data input	**G group 14**: Absolute/incremental dimensions modally effective		
G91	Incremental dimension data input			
G94 *	Feed F in mm/min	**G group 15:** Feedrate/spindle modally effective		
CFC *	Feedrate override with circle ON	**G group 16:** Feedrate override modally effective		
CFTCP	Feedrate override OFF			

G450 *	Transition circle	**G group 18:** Corner behavior with tool radius compensation modally effective	
G451	Point of intersection		
BRISK *	Jerking path acceleration	**G group 21:** Acceleration profile modally effective	
SOFT	Jerk-limited path acceleration		
FFWOF *	Feedforward control OFF	**G group 24:** Feedforward control modally effective	
FFWON	Feedforward control ON		
WALIMON *	Working area limitation ON	**G group 28:** Working area limitation modally effective	Applies to all axes activated via setting data; values set via G25, G26
WALIMOF	Working area limitation OFF		
G290 *	SIEMENS mode	**G group 47:** External NC languages modally effective	
The functions marked with * act at program start (CNC variant for the nibbling technology unless otherwise programmed and provided that the machine manufacturer default settings have not been changed).			

Address	Meaning	Value assignments	Information	Programming
H H0= to H9999=	H function	± 0.0000001 ... 9999 9999 (8 decimals) or with specification of an exponent: ± (10^{-300} ... 10^{+300})	Value transfer to the PLC; meaning defined by the machine manufacturer	H0=... H9999=... e.g.: H7=23.456
I	Interpolation parameters	±0.001 ... 99 999.999	Belonging to the X axis; meaning depending on G2, G3 –> circle center point	See G2, G3
J	Interpolation parameters	±0.001 ... 99 999.999	Belonging to the Y axis; otherwise, as with I	See G2, G3
I1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belonging to the X axis; specification for circular interpolation with CIP	See CIP
J1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belonging to the Y axis; specification for circular interpolation with CIP	See CIP
L	Subroutine; name and call	7 decimals; integer only, no sign	Instead of a free name, it is also possible to select L1 ...L9999999; this also calls the subroutine (UP) in a separate block, Please note: L0001 is not always equal to L1. The name “LL6” is reserved for the tool change subroutine.	L781 ; Separate block
M	Additional function	0 ... 99 only integer, without sign	For example, for initiating switching actions, such as “Coolant ON”; max. 5 M functions per block	M...

Address	Meaning	Value assignments	Information	Programming
M0	Programmed stop		The machining is stopped at the end of a block containing M0; to continue, press NC START.	
M1	Optional stop		As with M0, but the stop is only performed if a special signal (Program control: “M01”) is present.	
M2	End of program		Can be found in the last block of the processing sequence	
M30	–		Reserved; do not use	
M17	–		Reserved; do not use	
M6	Tool change		Only if activated with M6 via the machine control panel; otherwise, change directly using the T command	
M70, M19	–		Reserved; do not use	
M...	Remaining M functions		Functionality is not defined by the control system and can therefore be used freely by the machine manufacturer	
N	Block number – subblock	0 ... 9999 9999 only integer, without sign	Can be used to identify blocks with a number; is written in the beginning of a block	N20 ...
:	Block number of a main block	0 ... 9999 9999 only integer, without sign	Special block identification, used instead of N... ; such a block should contain all instructions for a complete subsequent machining step.	**:**20 ...
P	Number of subroutine cycles	1 ... 9999 only integer, without sign	Is used if the subroutine is run several times and is contained in the same block as the call	N10 L781 P... ; Separate block N10 L871 P3 ; Run three times

R0
to
R299
	Arithmetic parameters	± 0.0000001 ... 9999 9999 (8 decimals) or with specification of an exponent:
		± (10^{-300} ... 10^{+300})
				R1=7.9431 R2=4
				with specification of an exponent:
				R1=–1.9876EX9 ; R1=–1 987 600 000

Arithmetic functions			In addition to the 4 basic arithmetic functions using the operands + – * /, there are the following arithmetic functions:	
SIN( )	Sine	Degrees		R1=SIN(17.35)
COS()	Cosine	Degrees		R2=COS(R3)
TAN()	Tangent	Degrees		R4=TAN(R5)
ASIN()	Arc sine			R10=ASIN(0.35) ; R10: 20.487 degrees
ACOS()	Arc cosine			R20=ACOS(R2) ; R20: ... Degrees

Address	Meaning	Value assignments	Information	Programming
ATAN2(,)	Arc tangens2		The angle of the sum vector is calculated from 2 vectors standing vertically one on another. The 2nd vector specified is always used for angle reference. Result in the range: –180 to +180 degrees	R40=ATAN2(30.5,80.1) ; R40: 20.8455 degrees
SQRT()	Square root			R6=SQRT(R7)
POT()	Square			R12=POT(R13)
ABS()	Absolute value			R8=ABS(R9)
TRUNC()	Integer component			R10=TRUNC(R11)
LN()	Natural logarithm			R12=LN(R9)
EXP()	Exponential function			R13=EXP(R1)
RET	Subroutine end		Used instead of M2 – to maintain the continuous-path control mode	RET ; Separate block
S	Dwell time in the block with G4	0.001 ... 99 999.999	Dwell time in spindle revolutions	G4 S... ; Separate block
T	Tool number	1 ... 32 000 only integer, without sign	The tool change can be performed either directly using the T command or only with M6. This can be set in the machine data.	T...
X	Axis	±0.001 ... 99 999.999	Positional data	X...
Y	Axis	±0.001 ... 99 999.999	Positional data	Y...
AC	Absolute coordinate	–	The dimension can be specified for the end or center point of a certain axis, irrespective of G91.	N10 G91 X10 Z=AC(20) ; X – incremental dimension, Z – absolute dimension
ACC[*axis*]	Percentage acceleration override	1 ... 200, integer	Acceleration override for an axis; specified as a percentage	N10 ACC[X]=80 ; 80% acceleration for the X axis
ACP	Absolute coordinate; approach position in the positive direction (for rotary axis)	–	It is also possible to specify the dimensions for the end point of a rotary axis with ACP(...) irrespective of G90/G91.	N10 A=ACP(45.3) ; Approach absolute position of the A axis in the positive direction
ACN	Absolute coordinate; approach position in the negative direction (for rotary axis)	–	It is also possible to specify the dimensions for the end point of a rotary axis with ACP(...) irrespective of G90/G91.	N10 A=ACP(45.3) ; Approach absolute position of the A axis in the negative direction
ANG	Angle for the specification of a straight line for the contour definition	±0.00001 ... 359.99999	Specified in degrees; one possibility of specifying a straight line when using G0 or G1 if only one end-point coordinate of the plane is known or if the complete end point is known with contour ranging over several blocks	N10 G1 G17 X... Y.... N11 X... **ANG=...** or contour over several blocks: N10 G1 G17 X... Y.... N11 **ANG=...** N12 X... Y... **ANG=...**

Address	Meaning	Value assignments	Information	Programming
AP	Polar angle	0 ... ±359.99999	Specification in degrees, traversing in polar coordinates, definition of the pole; in addition: Polar radius RP	see G0, G1, G2, G3, G110, G111, G112
AR	Aperture angle for circular interpolation	0.00001 ... 359.99999	Specified in degrees; one possibility of defining the circle when using G2/G3	See G2, G3
CALL	Indirect cycle call	–	Special form of the cycle call; no parameter transfer; the name of the cycle is stored in a variable; only intended for cycle-internal use	N10 CALL VARNAME ; Variable name
CR	Radius for circular interpolation	0.010 ... 99 999.999 Negative sign – for selecting the circle: greater than semi-circle	One possibility of defining a circle when using G2/G3	See G2, G3
DC	Absolute coordinate; approach position directly (for rotary axis, spindle)	–	It is also possible to specify the dimensions for the end point of a rotary axis with DC(...) irrespective of G90/G91.	N10 A=DC(45.3) ; Approach A axis position directly
DEF	Definition instruction		Defining a user variable of the type BOOL, CHAR, INT, REAL, STRING[n], to be defined directly at the beginning of the program	DEF INT VARI1=24, VARI2 ; 2 variables of the INT type ; name will be def by user DEF STRING[12] VARS3="HELLO" ; max. 12 char.
GOTOB	GoBack instruction	–	A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the program start.	N10 LABEL1: ... ... N100 GOTOB LABEL1
GOTOF	GoTo instruction	–	A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the end of the program.	N10 GOTOF LABEL2 ... N130 LABEL2: ...
IC	Coordinate specified using incremental dimensions	–	The dimension can be specified for the end or center point of a certain axis irrespective of G90.	N10 G90 X10 Y=IC(20) ; Y – incremental dimension, X – absolute dimension
IF	Jump condition	–	If the jump condition is fulfilled, the jump to the block with *Label: is performed;* otherwise, next instruction/block; several IF instructions per block are possible **Comparison operators:** = = Greater than, <> not equal to > greater than, < less than >= greater than or equal to <= less than or equal to	N10 IF R1>5 GOTOF LABEL3 ... N80 LABEL3: ...
$A_DBB[n] $A_DBW[n] $A_DBD[n] $A_DBR[n]	Data byte Data word Data double-word Real data		Reading and writing PLC variables	N10 $A_DBR[5]=16.3 ; Writing the REAL variables with offset position 5 ; (position, type and meaning are agreed between NC and PLC)
$A_MONIFACT	Factor for tool life monitoring	> 0.0	Initialization value: 1.0	N10 $A_MONIFACT=5.0 ; Tool life elapsed 5 times faster

Address	Meaning	Value assignments	Information	Programming
$AA_MM[*axis*]	Measurement result for an axis in the **machine coordinate system**	–	*Axis*: Identifier of an axis (X, Y) traversing when measuring	N10 R1=$AA_MM[X]
$AA_MW[*axis*]	Measurement result for an axis in the **workpiece coordinate system**	-	*Axis*: Identifier of an axis (X, Y) traversing when measuring	N10 R2=$AA_MW[X]
$A..._..._ TIME	Timer for runtime: $AN_SETUP_TIME $AN_POWERON_TIME $AC_OPERATING_TIME $AC_CYCLE_TIME $AC_CUTTING_TIME	0.0 ... 10^{+300} min (value read-only) min (value read-only) s s s	System variable: Time since the control system has last booted Time since the control system has last booted normally Total runtime of all NC programs Runtime of the NC program (only of the selected program) Tool action time	N10 IF $AC_CYCLE_TIME==50.5 ....
$AC_..._ PARTS	Workpiece counter: $AC_TOTAL_PARTS $AC_REQUIRED _PARTS $AC_ACTUAL_PARTS $AC_SPECIAL_PARTS	0 ... 999 999 999, integer	System variable: Total actual count Set number of workpiece Current actual count Count of workpieces – specified by the user	N10 IF $AC_ACTUAL_PARTS==15 ....
$P_ TOOLNO	Number of the act. tool T	–	read-only	N10 IF $P_TOOLNO==12 GOTOF ....
$P_TOOL	Active D number of the active tool	–	read-only	N10 IF $P_TOOL==1 GOTOF ....
$TC_MOP1[t, d]	Tool life warning limit	0.0 ...	in minutes, writing or reading values for tool t, D number d	N10 IF $TC_MOP1[13,1]<15.8 GOTOF ....
$TC_MOP2[t, d]	Residual tool life	0.0 ...	in minutes, writing or reading values for tool t, D number d	N10 IF $TC_MOP1[13,1]<15.8 GOTOF ....
$TC_MOP3[t, d]	Warning limit for count	0 ... 999 999 999, integer	Writing or reading values for tool t, D number d	N10 IF $TC_MOP1[13,1]<15 GOTOF ....
$TC_MOP4[t, d]	Residual quantity	0 ... 999 999 999, integer	Writing or reading values for tool t, D number d	N10 IF $TC_MOP1[13,1]<8 GOTOF ....
$TC_MOP11[t,d]	Required tool life	0.0 ...	in minutes, writing or reading values for tool t, D number d	N10 $TC_MOP11[13,1]=247.5
$TC_MOP13[t,d]	Target quantity	0 ... 999 999 999, integer	Writing or reading values for tool t, D number d	N10 $TC_MOP13[13,1]=715
$TC_TP8[t]	Status of the tool	–	Default status - coding by bits for tool t, (bit 0 to bit 4)	N10 IF $TC_TP8[1]==1 GOTOF ....
$TC_TP9[t]	Type of monitoring of the tool	0 ... 2	Monitoring type for tool t, writing or reading 0: No monitoring, 1: Tool life = 2: Number of workpieces	N10 $TC_TP9[1]=2 ; Select count monitoring

Address	Meaning	Value assignments	Information	Programming
MCALL	Modal subroutine call	–	The subroutine in the block containing MCALL is called automatically after each successive block containing a path motion. The call acts until the next MCALL is called. Application example: Drilling a hole pattern	N10 MCALL CYCLE82(...) ; Separate block, drilling cycle N20 HOLES1(...) ; Row of holes N30 MCALL ; Separate block, modal ; Call of CYCLE82(...) comp.
MSG ()	Message	max. 65 characters	Message text in inverted commas	N10 MSG(“MESSAGE TEXT”) ; Separate block ... N150 MSG() ; Cancel previous message
RP	Polar radius	0.001 ... 99 999.999	Traversing in polar coordinates, pole specification; in addition: Polar angle AP	see G0, G1, G2; G3 G110, G111, G112
RPL	Angle of rotation with ROT, AROT	±0.00001 ... 359.9999	Specification in degrees; angle for a programmable rotation in the current plane G17 to G19	see ROT, AROT
PON	Punching ON		Activates the punching function and deactivates SON; Stroke is only initiated at the end-of-block point	
PONS	Punching ON, activated in the position controller		As with PON, but triggered in the position control cycle	
PDELAYON	Delay when punching ON			
PDELAYOF *	Delay when punching OFF			
SON	Nibbling ON		Activates the nibbling function and deactivates PON; first stroke is performed at the starting point of the activated block	
SONS	Nibbling ON, activated in the position controller		The same behavior as SON, but triggered in the position control cycle	
SPIF1 *	First punching interface active		Fast NCK inputs/outputs for punching/nibbling byte 1	
SPOF *	Punching/nibbling OFF		Quits all punching/nibbling functions	
SPN	Number of path sections per block (stroke/punch number)		Automatic path segmentation into the *programmed number* of path segments	
SPP	Length of path section (stroke/punch path)		Divides the programmed total path automatically into path segments of equal length according to the *programmed distance* to be traversed	

SET(, , ,)
REP()
	Set values for the variable fields		SET: Various values, from the specified element up to: according to the number of values
			REP: the same value, from the specified element up to the end of the field
				DEF REAL VAR2[12]=REP(4.5)
				; All elements with value 4.5
				N10 R10=SET(1.1,2.3,4.4)
				; R10=1.1, R11=2.3, R4=4.4

Address	Meaning	Value assignments	Information	Programming
STOPFIFO	Stops the fast machining step	–	Special function; filling of the buffer memory until STARTFIFO, “Buffer memory full” or “End of program” is detected.	STOPFIFO ; Separate block, start of filling N10 X... N20 X...
STARTFIFO *	Start of fast machining step	–	Special function; the buffer memory is filled at the same time.	N30 X... STARTFIFO ; Separate block, end of filling
STOPRE	Preprocessing stop	–	Special function; the next block is only decoded if the block before STOPRE is completed.	STOPRE ; Separate block
TANG(Fo, Le1,Le2,...)	Tangential control, definition	–	Fo: Name of the following axis (rotary axis) Le1: Name of leading axis 1 Le2: Name of leading axis 2 Further parameters optionally Function only available with SINUMERIK 802D sl pro!	TANG(C,X,Y) ; Separate block TANG(C,X,Y,1“W”,“P”) ; Max. number of parameters
TANGON (Fo,...)	Activate tangential control	–	Fo: Name of the following axis (rotary axis) This function is only available with SINUMERIK 802D sl pro!	TANGON(C) ; Separate block TANGON(C,Angle,Dist,Angletol) ; Max. number of parameters
TANGOF (Fo)	Deactivate tangential control	–	Fo: Name of the following axis (rotary axis) This function is only available with SINUMERIK 802D sl pro!	TANGOF(C) ; Separate block
TANGDEL (Fo)	Tangential control, clear definition	–	Fo: Name of the following axis (rotary axis) This function is only available with SINUMERIK 802D sl pro!	TANGDEL(C) ; Separate block
TLIFT(Fo)	Tangential control, insert intermediate block	–	Fo: Name of the following axis (rotary axis) This function is only available with SINUMERIK 802D sl pro!	TLIFT(C) ; Separate block
TRAILON	Activate coupled-axis motion		Define and activate coupled-axis motion	TRAILON(V,Y,K) ; V=coupled axis, Y=leading axis K=coupling factor
TRAILOF	Deactivate coupled-axis motion		Deactivate coupled axes	TRAILOF(V,Y)
PUNCHACC	Travel-dependent acceleration		Definition of a travel-dependent acceleration characteristic for nibbling/punching	PUNCHACC(S_{min}, A_{min}, S_{max}, A_{max})
CPROT	Activation of protection zones		Protection zones must be defined before they can be activated	CPROT(1,2) ; Activate protection zone 1 CPROT(1,0) ; Deactivate protection zone 1
CAC	Approach position absolutely		Coded value is table index; table value is approached	
CACN	Absolute approach in negative direction of value stored in table.		Permissible as positioning axes for the programming of rotary axes	

Address	Meaning	Value assignments	Information	Programming
CACP	Absolute approach in positive direction of value stored in table.			
CDC	Direct approach of a position			
CIC	Approach position by increments			

8.2 Positional data

8.2.1 Plane selection G17

Functionality

The X and Y axes constitute a plane called G17 plane. Only this is relevant for nibbling (default setting).

The plane selection is described with the individual functions.

The individual planes are also used to define the **direction of rotation of the circle for circular interpolation** CW or CCW. In the plane in which the circle is traversed, the abscissa and the ordinate are designed and thus also the direction of rotation of the circle.

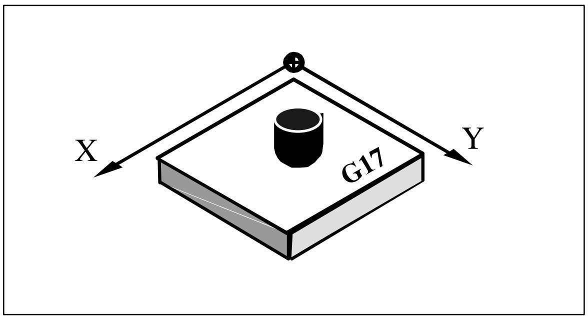


Fig. 8-3 Planes and axis assignment for nibbling

Programming example

```
N10 G17 T... M... ; X/Y plane selected
N20 ... X... Y...
```

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

Functionality

With the instructions G90/G91, the written positional data X, Y, ... are evaluated as a coordinate point (G90) or as an axis position to traverse to (G91). G90/91 applies for all axes. Irrespective of G90/G91, certain positional data can be specified for certain blocks in absolute/incremental dimensions using AC/IC.

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

Programming

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

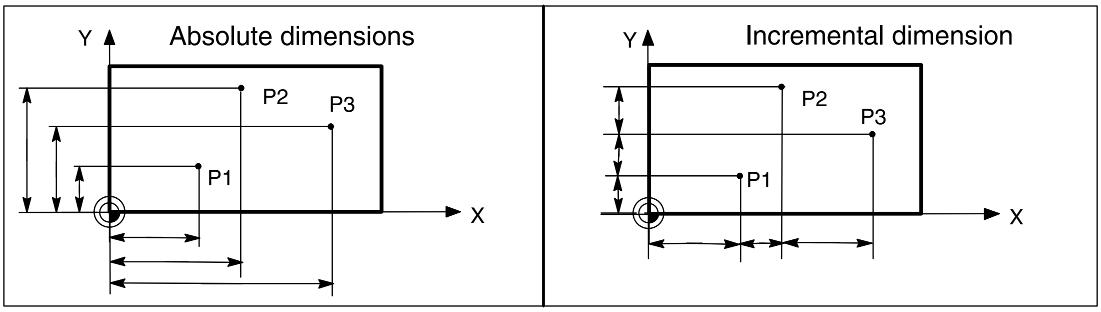


Fig. 8-4 Different dimensioning types in the drawing

Absolute dimensioning G90

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

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

Incremental dimensioning G91

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

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

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

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

Programming example

```
N10 G90 X20 Y90          ; Absolute dimensioning
N20 X75 Y=IC(–32)        ; X dimensioning still absolute, Y incremental dimensioning
...
N180 G91 X40 Y20         ; Switching to incremental dimensioning
N190 X–12 Y=AC(17)       ; X still incremental dimensioning, Y absolute
```

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

Functionality

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

Programming

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

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

Programming example

```
N10 G70 X10 Y30               ; Inch dimensional notation
N20 X40 Y50                   ; G70 remains active
...
N80 G71 X19 Y17.3             ; Metric dimensional notation from here...
```

Information

Depending on the **default setting** you have chosen, the control system interprets all geometric values as either metric **or** inch dimensions. Tool offsets and settable work offsets including their displays are also to be understood as geometrical values; this also applies to the feed F in mm/min or inch/min.
The default setting can be set in machine data. All examples provided in this Manual assume the **metric default setting**.

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

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

All remaining geometrical specifications that are no direct workpiece specifications, such as feedrates, tool offsets, **settable** work offsets, are not influenced by **G70/G71**.
G700/G710, however, influences additionally the feedrate F (inch/min, inch/rev. or mm/min, mm/rev.).

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

Functionality

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

Plane

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

Polar radius RP=...

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

Polar angle AP=...

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

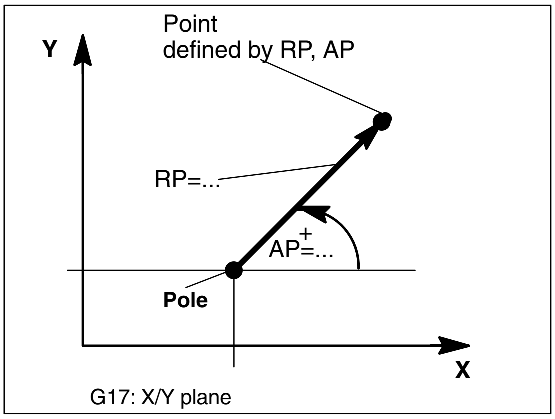


Fig. 8-5 Polar radius and polar angle with definition of the positive direction

Pole definition, programming

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

Information

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

Programming example

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

Traversing with polar coordinates

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

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

(see also Section 8.3 “Axis movements”).

8.2.5 Programmable work offset: TRANS, ATRANS

Functionality

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

Programming

```
TRANS X... Y...        ; Programmable offset; clears old instructions for
                         offset, rotation, scaling factor and mirroring
ATRANS X... Y...       ; Programmable offset,
                         additive to the existing instructions
TRANS                  ; Without values: clears old instructions for offset, rotation,
                       scaling factor and mirroring
```

The instructions with TRANS, ATRANS always require a separate block.

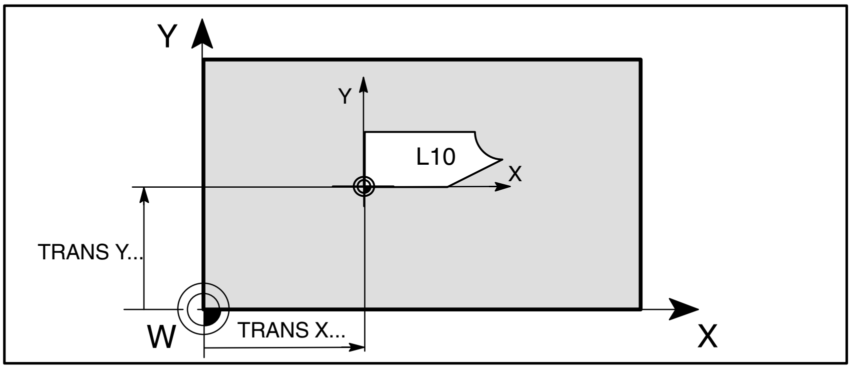


Fig. 8-6 Programmable offset (example)

Programming example

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

Subroutine call – see Section 8.9 “Subroutine technique ”

8.2.6 Programmable rotation: ROT, AROT

Functionality

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

Programming

```
ROT RPL=...      ; Programmable rotation; clears old instructions for offset,
                   rotation, scaling factor and mirroring
AROT RPL=...     ; Programmable rotation, additive to the existing instructions
ROT              ; Without values: clears old instructions for offset, rotation,
                 scaling factor and mirroring
```

The instructions with ROT, AROT always require a separate block.

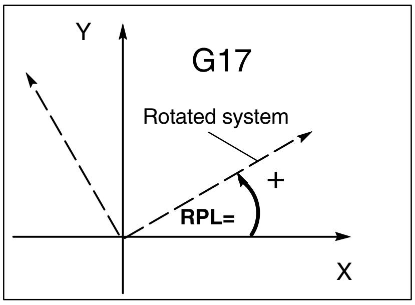


Fig. 8-7 Definition of the positive direction of the angles of rotation

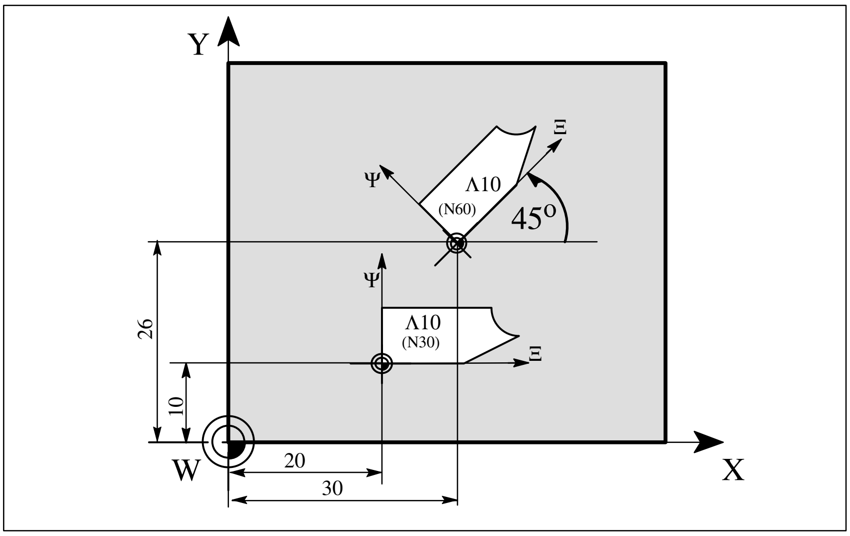


Fig. 8-8 Programming example for programmable offset and rotation

Programming example

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

Subroutine call – see Section 8.9 “Subroutine technique ”

8.2.7 Programmable scaling factor: SCALE, ASCALE

Functionality

A scaling factor can be programmed for all axes with SCALE, ASCALE. The path is enlarged or reduced by this factor in the axis specified.
The currently set coordinate system is used as the reference for the scale change.

Programming

```
SCALE X... Y...         ; Programmable scaling factor; clears old instructions for
                          offset, rotation, scaling factor and mirroring
ASCALE X... Y...        ; Programmable scaling factor, additive to the existing
                          instructions
SCALE                   ; Without values: clears old instructions for offset, rotation,
                        scaling factor, mirroring
```

The instructions with SCALE, ASCALE always require a seprate block.

Information

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

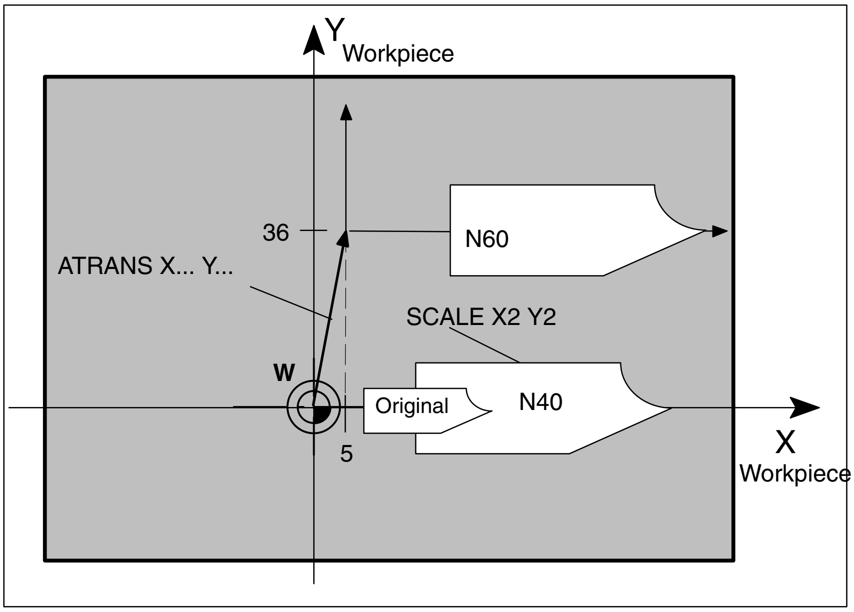


Fig. 8-9 Example for scaling and offset

Programming example

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

Subroutine call – see Section 8.9 “Subroutine technique”

8.2.8 Programmable mirroring: MIRROR, AMIRROR

Functionality

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

Programming

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

The instructions with MIRROR, AMIRROR always require a seprate block. The axis value has no influence. A value, however, must be specified.

Note

The direction of rotation of the circle G2/G3 is also reversed automatically when mirroring.

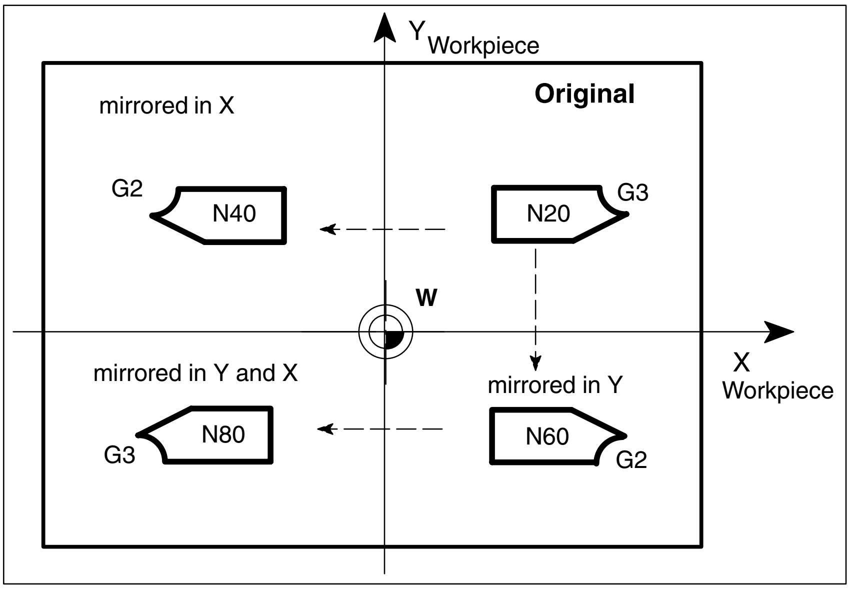


Fig. 8-10 Example for mirroring with the tool position shown

Programming example

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

```
...
N10 G17                    ; X/Y plane,
N20 L10                    ; Programmed contour
N30 MIRROR X0              ; The direction is changed in X
N40 L10                    ; Mirrored contour
N50 MIRROR Y0              ; The direction is changed in Y
N60 L10
N70 AMIRROR X0             ; Mirroring once more, but now in X
N80 L10                    ; Contour mirrored twice
N90 MIRROR                 ; Mirroring OFF.
...
```

Subroutine call – see Section 8.9 “Subroutine technique ”

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

Functionality

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

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

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

Programming

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

G53          ; Settable work offset OFF – non–modal,
             ; Suppresses also the programmable offset
G153         ; The same as G53, but suppresses additionally base frame
```

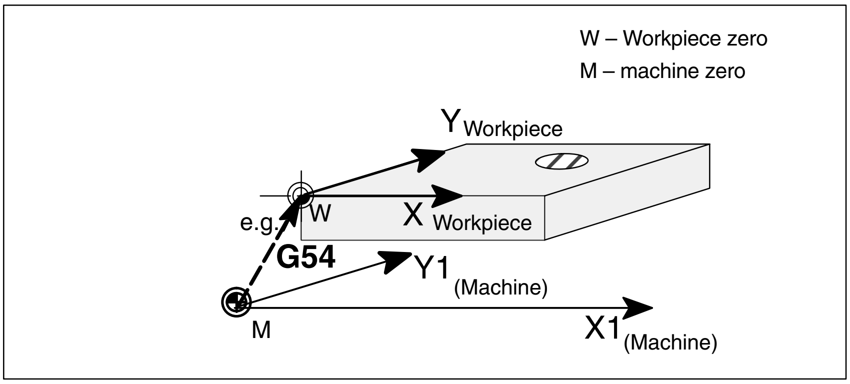


Fig. 8-11 Settable work offset

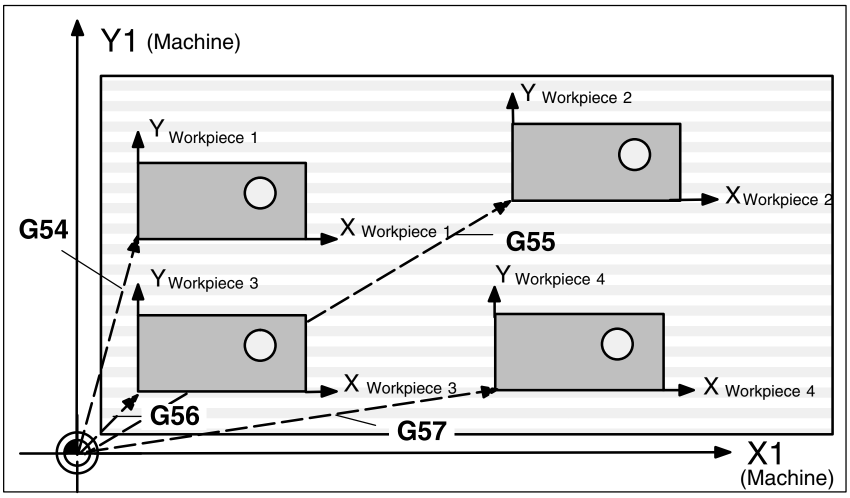


Fig. 8-12 Several workpiece clampings

Programming example

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

Subroutine call – see Section 8.9 “Subroutine technique ”

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

Functionality

The working area limitation defines the working area for all axes. Traversing is allowed in this area only. The coordinate parameters are machine-based.

To be able to use the working area limitation, it must be activated for the axis concerned. This is done using the interactive screenform under <Offset Param.> <Setting data> <Working area limit.>.

There are two possibilities to define the working area:

- Input of the values using the interactive screenform of the control system under <Offset Param> <Setting data> <Working area limit.>
  Thus, the working area limitation is also effective in the JOG mode.
- Programming using G25/G26
  The values for the individual axes can be changed in the part program. The values entered in the interactive screenform (<Offset Param.> <Setting data> <Working area limit.>) are overwritten.

The working area limitation is enabled/disabled in the program by WALIMON/WALIMOF.

Programming

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

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

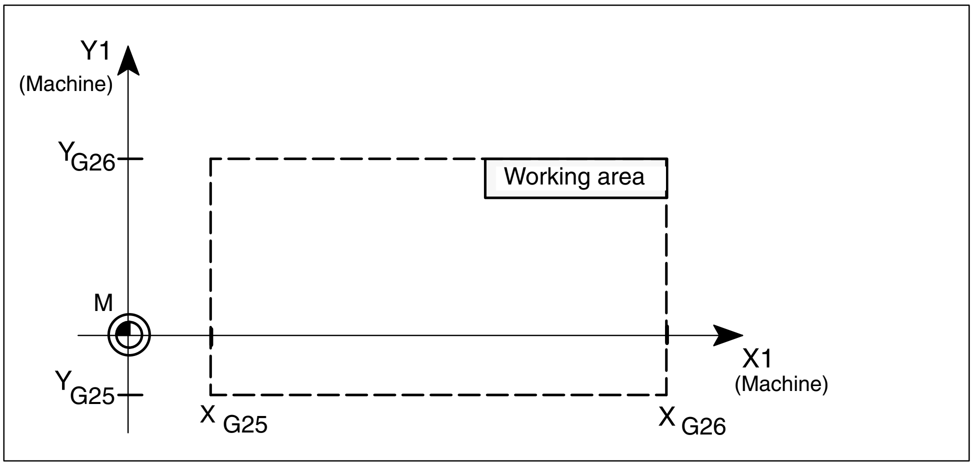


Fig. 8-13 Programmable working area limitation (example: 2-dimensional)

Information

- For G25, G26, the channel axis identifier consisting of machine data 20080 AXCONF_CHANAX_NAME_TAB is to be used. These can be other than the geometry axis identifiers in MD 20060: AXCONF_GEOAX_NAME_TAB.
- A working area limitation can only be activated if the reference point for the relevant axes has been approached.

Programming example

```
N10 G25 X10 Y-20             ; Values for the lower working area limitation
N20 G26 X100 Y110            ; Values for the upper working area limitation
N30 T1 M6
N40 G0 X90 Y100
N50 WALIMON                  ; Working area limitation ON
...                          ; Work only within the limited area
N90 WALIMOF                  ; Working area limitation OFF
```

8.3 Axis movements

8.3.1 Linear interpolation at rapid traverse: G0

Functionality

The rapid traverse movement G0 is used for fast positioning of the tool, but **not directly for machining of the workpiece**.
It is possible to traverse all axes simultaneously – along a straight path.

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

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

Programming

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

Note: Another option for linear programming is available with the angle specification ANG=...

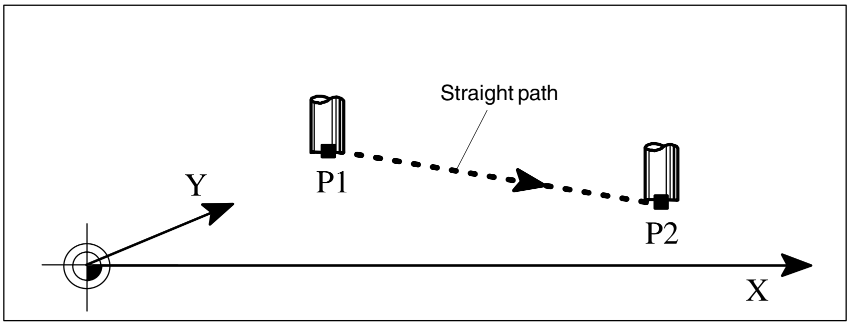


Fig. 8-14 Linear interpolation with rapid traverse from point P1 to P2

Programming example

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

Information

Another group of G functions exists for movement to the position (see Subsection 8.3.11 “Exact stop / continuous-path control mode: G60, G64”).
For G60 exact stop, a window with various precision values can be selected with another G group. For exact stop, an alternative instruction with non-modal effectiveness exists: G9.
You should consider these options for adaptation to your positioning tasks.

8.3.2 Linear interpolation with feedrate: G1

Functionality

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

Programming

```
G1 X... Y...  F...              ; Cartesian coordinates
G1  AP=... RP=... F...          ; Polar coordinates
```

Note: Another option for linear programming is available with the angle specification ANG=...

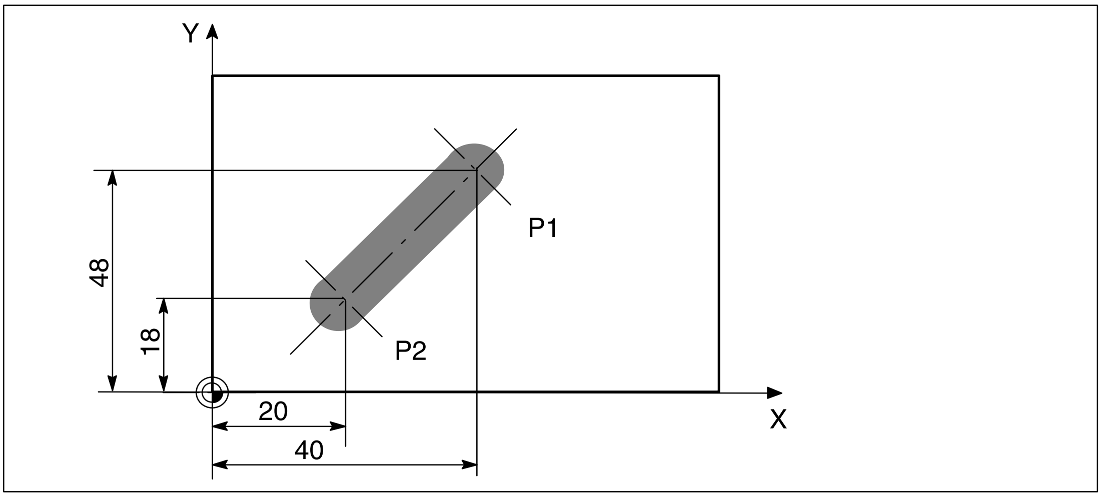


Fig. 8-15 Linear interpolation in three axes using the example of a slot

Programming example

```
N05 G0 G90 X40 Y48           ; Tool traverses at rapid traverse to P1,
                             ; 3 axes simultaneously,
N10 G1  F100                 ; Feedrate 100 mm/min
N15 X20 Y18                  ; Tool traverses along a straight line in the space to P2
N20 X-20 Y80
N30 M2                       ; End of program
```

8.3.3 Circular interpolation: G2, G3

Functionality

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

G2	; CW
G3	; CCW

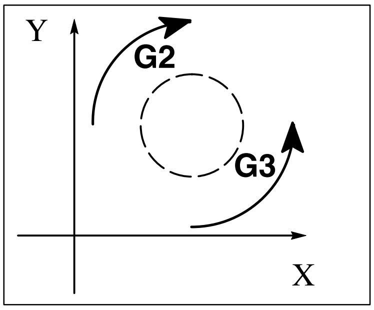


Fig. 8-16 Definition of the circular direction of rotation G2/G3

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

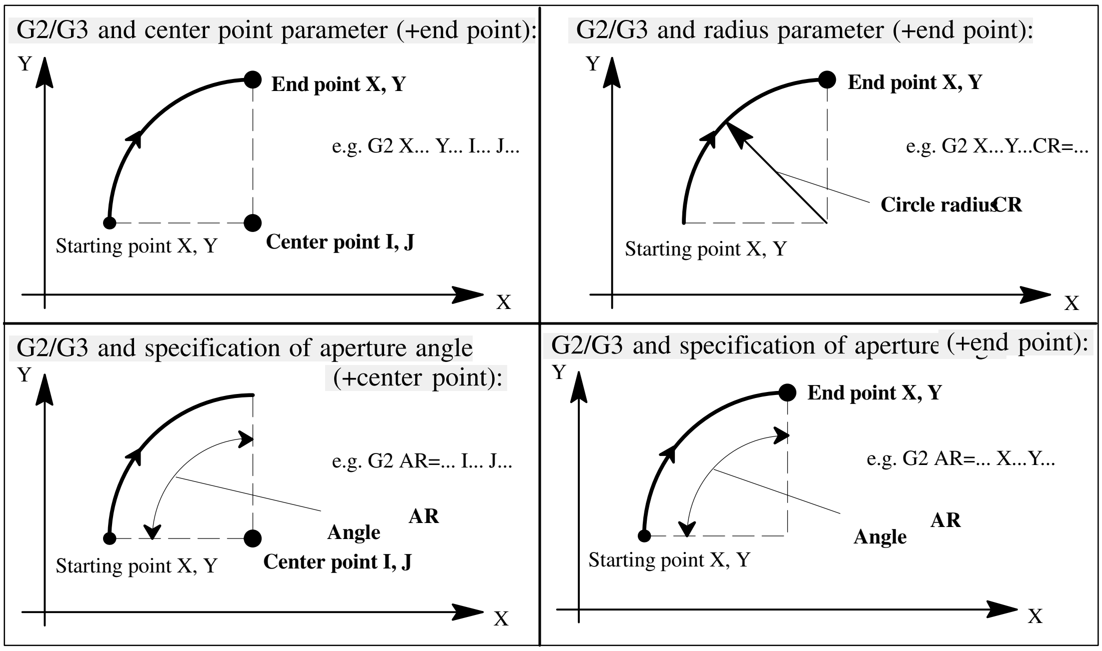


Fig. 8-17 Possibilities of circle programming with G2/G3 using the example of the axes X/Y and G2

G2/G3 are effective until they are canceled by another instruction of this G group (G0, G1, ...).

For the **path velocity**, the programmed **F word** is decisive.

Programming

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

Note

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

Input tolerances for the circle

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

The tolerance value can be set via machine data (see “Start-up Guide” 802Dsl).

Information

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

For circles with radius specification, the arithmetic sign of CR=... is used to select the correct circle. It is possible to program 2 circles with the same starting and end points, as well as with the same radius and the same direction. The negative sign in front of CR=–... determines the circle whose circle segment is greater than a semi-circle; otherwise, the circle with the circle segment is less than or equal to the semi-circle and determined as follows:

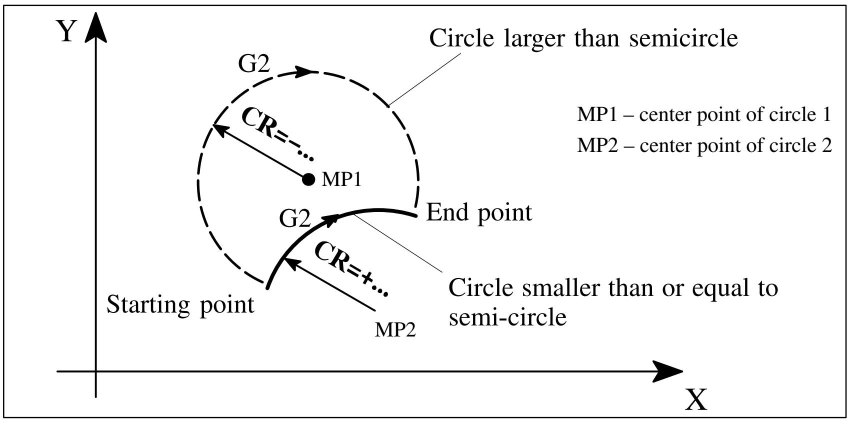

Y
Circle larger than semicircle
G2
CR=–...
MP1 – center point of circle 1
MP2 – center point of circle 2
MP1
G2
End point
CR=+...
Circle smaller than or equal to semi-circle
Starting point
MP2
X

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

Programming example: Definition of center point and end point

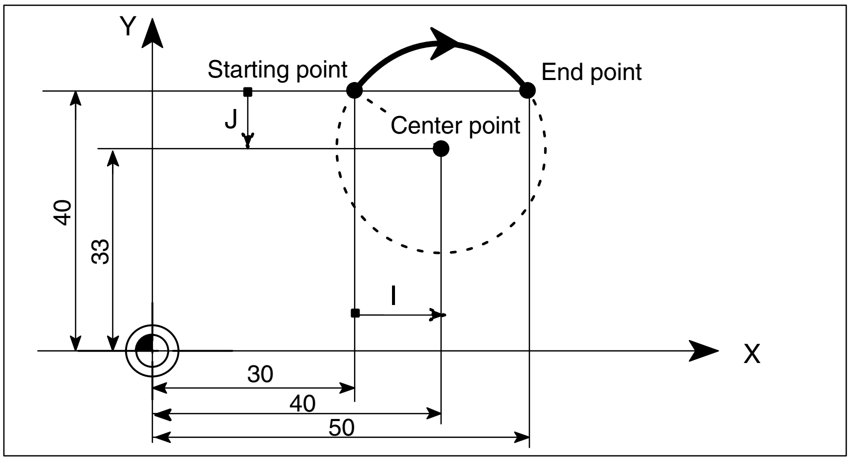


Fig. 8-19 Example for center point and end point specification

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

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

Programming example: End point and radius specification

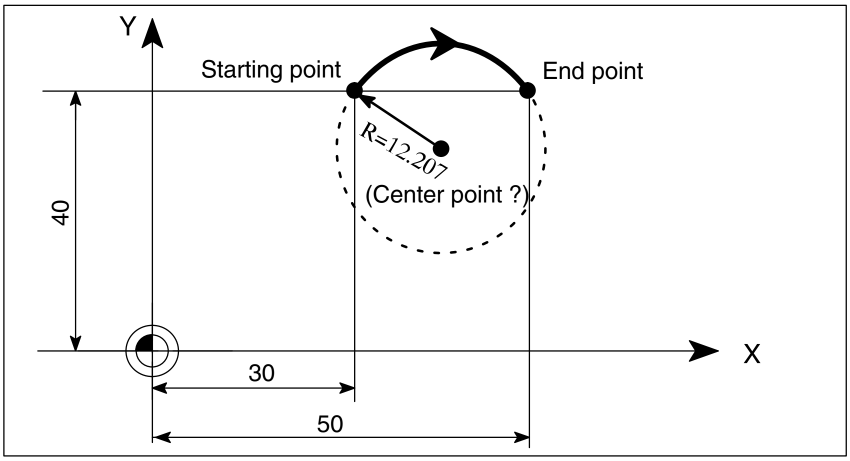


Fig. 8-20 Example for end point and radius specification

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

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

Programming example: Definition of end point and aperture angle

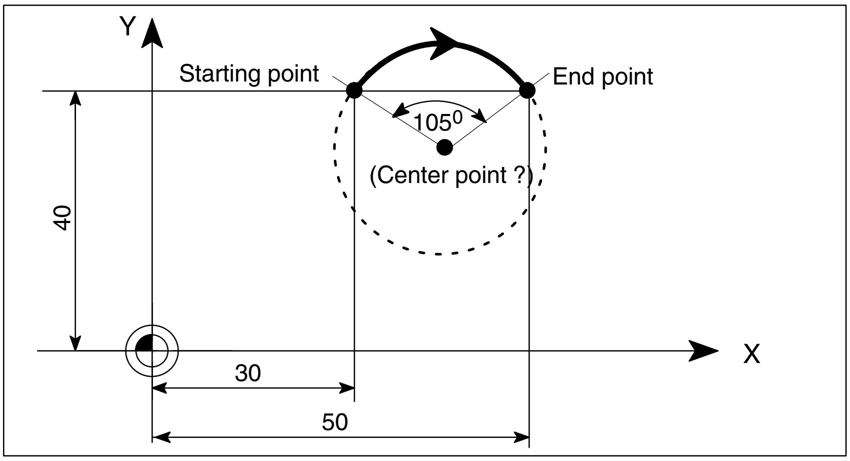


Fig. 8-21 Example for end point and aperture angle specification

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

Programming example: Definition of center point and aperture angle

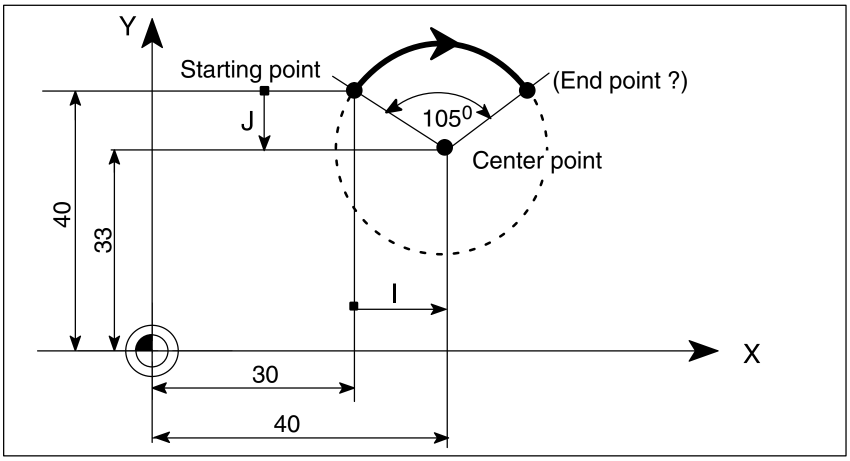


Fig. 8-22 Example for center point and aperture angle specification

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

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

Programming example: Polar coordinates

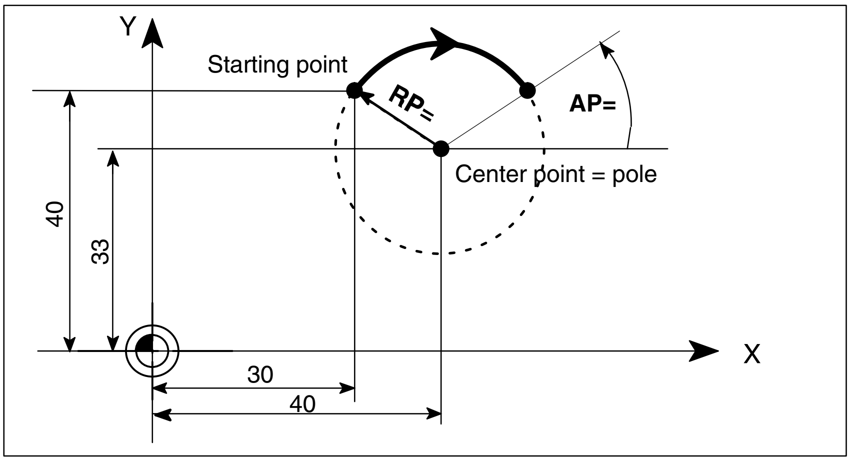


Fig. 8-23 Example for circle with polar coordinates

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

8.3.4 Circular interpolation via intermediate point: CIP

Functionality

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

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

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

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

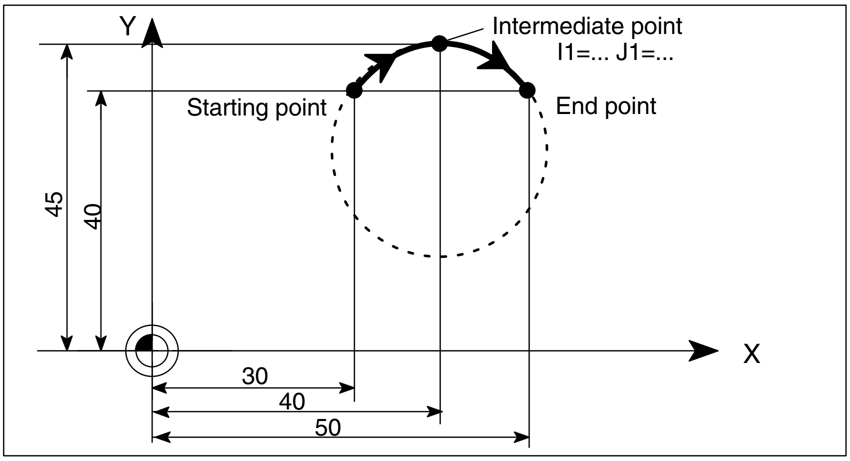


Fig. 8-24 Circle with end point and intermediate point specification using the example of G90

Programming example

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

8.3.5 Circle with tangential transition: CT

Functionality

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

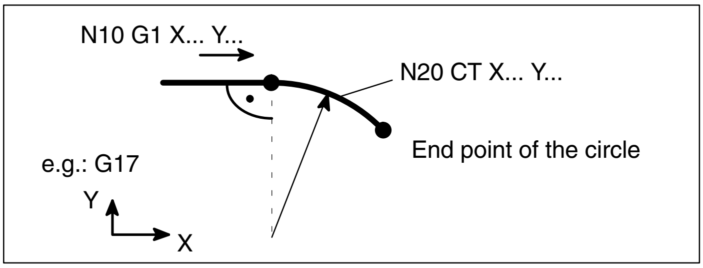


Fig. 8-25 Circle with tangential transition to the previous path section

Programming example

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

8.3.6 Approaching a fixed point: G75

Functionality

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

Programming example

```
N10 G75 X1=0 Y1=0
```

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

8.3.7 Approaching a reference point; G74

Functionality

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

Programming example

```
N10 G74 X1=0 Y1=0
```

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

8.3.8 Approaching coded positions: CAC, CIC, CDC, CACP, CACN

Function

Max. 60 (0 to 59) positions can be entered for 2 axes in position tables using the machine data.

Programming

CAC (n)

or

CIC (n)

or

CACP (n)

or

CACN (n)

Parameters

Table 8-2 Parameters

CAC (n)	Approaching absolutely coded positions
CIC (n)	Approaching a coded position incrementally by n spaces in the plus direction (+) or minus direction (–)
CDC (n)	Approaching a coded position using the shortest possible path (with rotary axes only)
CACP (n)	Approaching a coded position absolutely in the positive direction (with rotary axes only)
CACN (n)	Approaching a coded position absolutely in the negative direction (with rotary axes only)
(n)	Position numbers 1, 2, … max. 60 positions for each axis

Example: Positioning table for linear axis and rotary axis

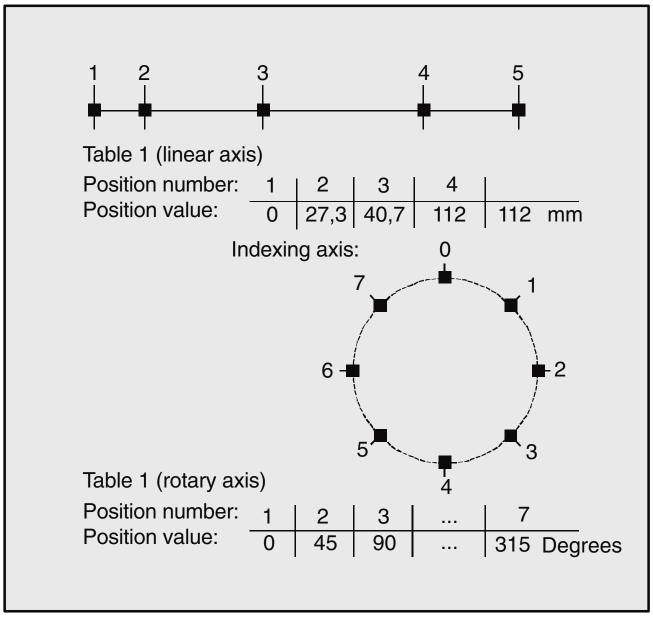

1
2
3
4
5
Table 1 (linear axis)
Position number: 1 | 2 | 3 | 4
Position value: 0 | 27,3 | 40,7 | 112 | 112 mm
Indexing axis:
0
1
2
3
4
5
6
7
Table 1 (rotary axis)
Position number: 1 | 2 | 3 | ... | 7
Position value: 0 | 45 | 90 | ... | 315 Degrees

Fig. 8-26

Note

If an axis is located between two positions, it does not traverse in response to an incremental position command with CIC (...). It is always advisable to program the first travel command with an absolute position value.

Example 2

```
N10 FA[B]= 300             ; Feedrate for position axis B
N20 POS[B]=CAC(10)         ; Approach coded position 10 (absolutely)
N30 POS[B]=CIC(-4)         ; Retract by 4 spaces from the current position
```

8.3.9 Tangential control: TANG, TANGON, TANGOF, TLIFT, TANGDEL

Functionality

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

The tangential control is used if, e.g. a tool is to be guided tangentially to the workpiece contour to be traversed.

The TANG( ) function is used to define coupled axes using a coupling factor. Coupled axes include a following axis (rotary axis) and two leading axes (axes of the machining plane). The following axis is guided along the tangent to the path traversed by the leading axes. TANGON( ) is used to enable the coupling, TANGOF( ) is used to disable the coupling. It is possible to specify an offset angle for the following axis (rotary axis) by specifying an angle programmed in TANGON( ). The TANGDEL( ) instruction can be used to delete a defined coupling in the disabled condition.
Defined parameters or values must be transferred using the relevant functions. If no values are specified for the axes in all parameters, these need not be written.

Programming

```
TANG(FAxis,LAxis1,LAxis2,Couple,CS,Opt)     ; Definition of the tangential coupling
TANGON(FAxis,Angle, Dist, Angletol)         ; Activate tangential control
TANGOF(FAxis)                               ; Deactivate tangential control
TLIFT(FAxis)                                ; Insert intermediate block at contour cor-
ners
TANGDEL(FAxis)                              ; Clear tangential coupling
```

Explanation of the parameters

FAxis – Following axis (tangentially following rotary axis)
LAxis1, LAxis2 – Leading axes 1 and 2 (path axes from which the tangent for tracking of the following axis is determined.)
Couple – Coupling factor (interrelation between change of the angle of the tangent and the following axis.)
Optional specification, default setting = 1
CS – Identifier for the coordinate system, optional specification:
"B" = Basic coordinate system (default setting)
Opt – Optimization: "S" = Default or
"P" = Automatic adaptation of the time characteristic
for
following and leading axes
Angle – Offset angle of the following axis
Dist – Approximation travel of the following axis, necessary for Opt = "P"
Angletol – Angle tolerance of the following axis, optional specification,
(only evaluated if Opt = "P")

Information

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

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

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

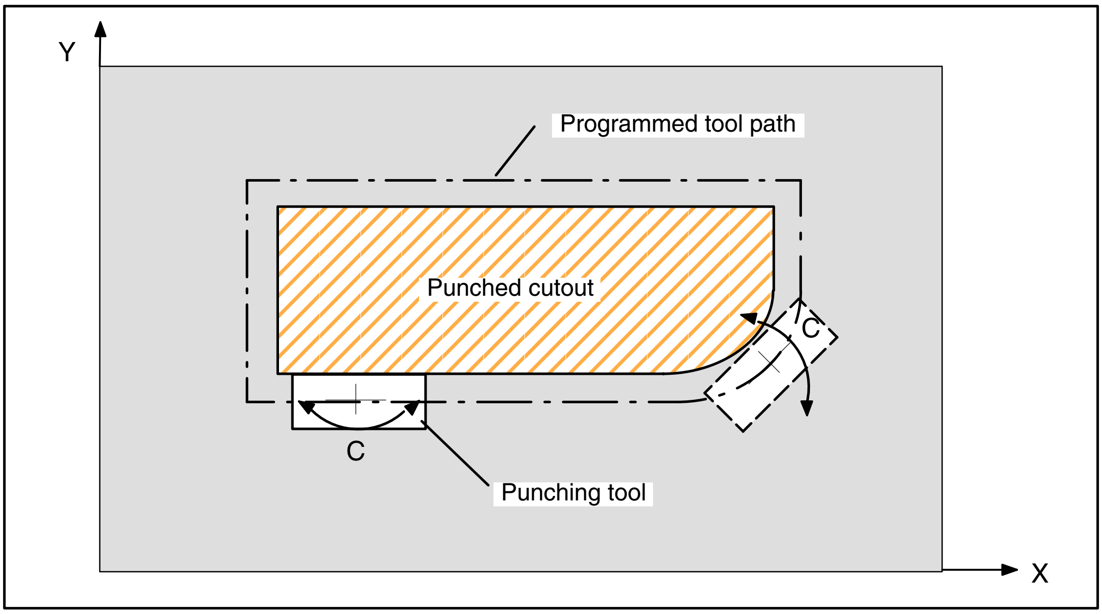


Fig. 8-27 Tangential control shown using the example of a punching tool: X, Y=leading axis, C=following axis

Programming example

```
N10 TANG(C, X, Y,1)          ; Define coupling with tangential control
N20 ...                      , approach starting point
N30 TANGON(C,0)              ; Activate coupling, C axis alignment 0 degrees
N40 G1 F800 X10 Y20          ; Traverse along contour in X, Y
...
N100 TANGOF(C)               ; Deactivate coupling
...
N200 M2
```

8.3.10 Feedrate F

Functionality

The feed F is the **path velocity** and represents the value of the geometric sum of the velocity components of all axes involved. The axis velocities are determined from the share of the axis path in the overall path.

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

Programming

F...

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

Unit of measure for F with G94, G95

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

G94 **F** as the feedrate in**mm/min**

Remark**:**
This unit of measure applies to metric dimensions. According to Section “Metric and inch dimensioning”, settings with inch dimensioning are also possible.

8.3.11 Exact stop / continuous-path control mode: G9, G60, G64

Functionality

G functions are provided for optimum adaptation to different requirements to set the traversing behavior at the block borders and for block advancing.
Example: For example, you would like to quickly position with the axes or you would like to machine path contours over multiple blocks.

Programming

```
G60        ; Exact stop – modally effective
G64        ; Path-control mode

G9         ; Exact stop – non-modal

G601       ; Exact stop fine window
G602       ; Exact stop coarse window
```

Exact stop G60, G9

If the exact stop function (G60 or G9) is active, the velocity for reaching the exact end position at the end of a block is decelerated to zero.

Another modal G group can be used here to set when the traversing movement of this block is considered ended and the next block is started.

- G601 ; Exact stop window fine
  Block advance takes place when all axes have reached the “Exact stop window fine” (value in the machine data).
- G602 ; Exact stop window fine
  Block advance takes place when all axes have reached the “Exact stop window fine” (value in the machine data).

The selection of the exact stop window has a significant influence on the total time if many positioning operations are executed. Fine adjustments require more time.

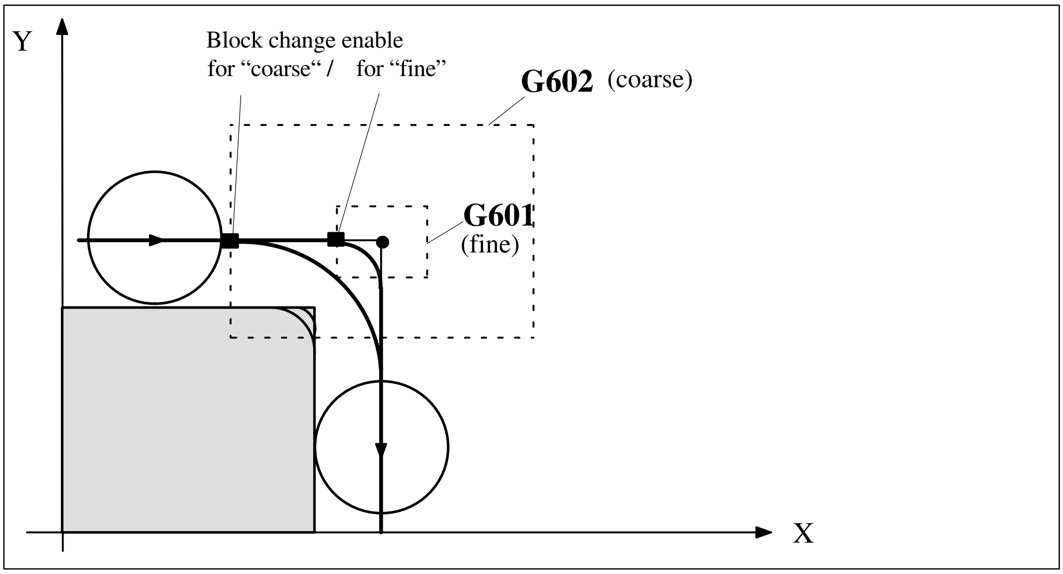


Fig. 8-28 Exact stop window coarse or fine, in effect for G60/G9; enlarged display of the windows

Programming example

```
N5 G602                      ; exact-stop window, coarse
N10 G0 G60  X...             ; Exact stop, modal
N20 X... Y...                ; G60 continues to be effective
...
N50 G1 G601 ...              ; Exact-stop window, fine
N80 G64 X...                 ; Switching to continuous path-control mode
...
N100 G0 G9 X...              ; Exact stop is only effective for this block
N111 ...                     ; Continuous-path control mode again
```

Remark: The G9 command only generates exact stop for the block in which it is programmed; G60, however, is effective until it is canceled by G64.

Continuous-path control mode G64

The objective of the continuous-path control mode is to avoid deceleration at the block boundaries and to switch **to the next block** with **a path velocity as constant as possible** (in the case of tangential transitions). The function works with **look-ahead velocity control** over several blocks.

For non-tangential transitions (corners), the velocity can reduced rapidly enough so that the axes are subject to a relatively high velocity change over a short time. This may lead to a significant jerk (acceleration change). The size of the jerk can be limited by activating the SOFT function.

Programming example

```
N10 G64 G1 X... F...          ; Continuous-path control mode
N20  Y..                      ; Continuous-path control mode continued
...
N180 G60 ...                  ; Switching to exact stop
```

Look-ahead velocity control

In the continuous-path control mode with G64, the control system automatically determines the velocity control for several NC block in advance. This enables acceleration and deceleration across multiple blocks with approximately tangential transitions. For paths that consist of short travels in the NC blocks, higher velocities can be achieved than without look ahead.

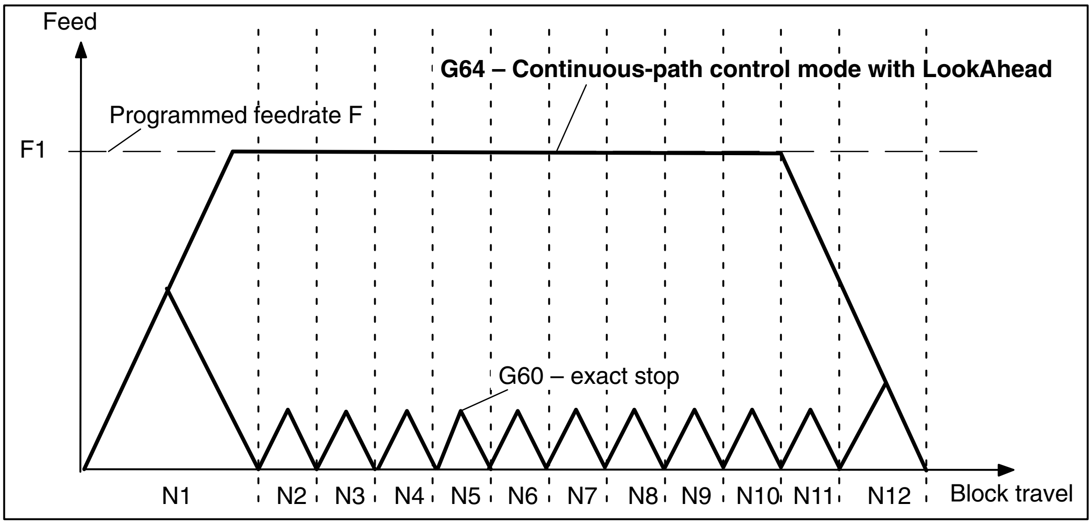


Fig. 8-29 Comparison of the G60 and G64 velocity behavior with short travels in the blocks

8.3.12 Acceleration behavior: BRISK, SOFT

BRISK

The axes of the machine change their velocities using the maximum permissible acceleration value until reaching the final velocity. BRISK allows time-optimized working. The set velocity is reached in a short time. However, jumps are present in the acceleration pattern.

SOFT

The axes of the machine accelerate along a non-linear, constant characteristic until reaching the final velocity. With this jerk-free acceleration, SOFT allows for reduced machine load. The same behavior can also be applied to braking procedures.

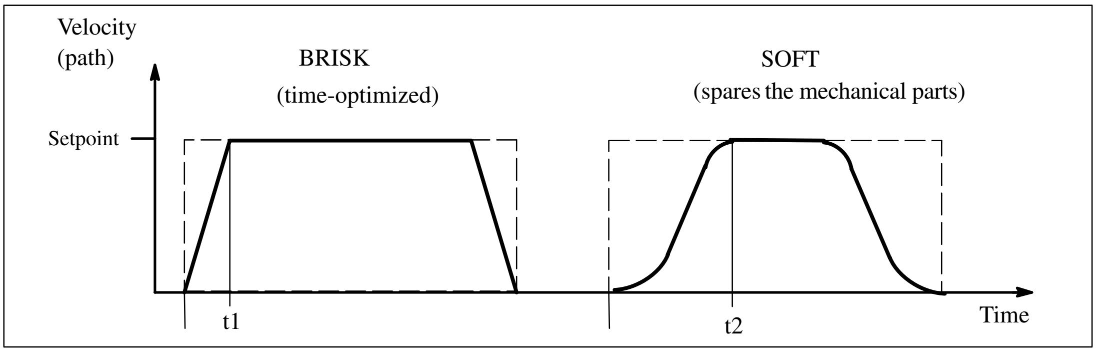


Fig. 8-30 Principle course of the path velocity when using BRISK/SOFT

Programming

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

Programming example

```
N10 SOFT G1 X30 Y84 F650      ; Path acceleration with jerk limitation
...
N90 BRISK  X87 Y104           ; Continue with stepped path acceleration
...
```

8.3.13 Percentage acceleration override: ACC

Functionality

Certain program sections can require the axis and spindle acceleration set via the machine data to be changed using the program. This programmable acceleration is a percentage acceleration override.

can be programmed to specify a percentage value of > 0% and ≤ 200% for each machine axis. The axis interpolation is then carried out with this proportional acceleration. Reference value (100%) is the valid machine data value for the acceleration.

Programming

ACC[*axis name*] = percentage ; for the axis

Programming example

```
N10 ACC[X]=80                ; 80% acceleration for the X axis
...
N100 ACC[X]=100              ; Disable override for the X axis
```

Activation

The limitation is in effect with all types of interpolation in the AUTOMATIC and MDA modes, but **not** in the JOG mode and when approaching the reference point.

The value assignment ACC [...] = 100 deactivates the override (100% of the MD values) as well as RESET and program end.

The programmed override value is also active with dry run feedrate.

Caution

A value greater than 100% may only be programmed if this load is permissible for the machine mechanics and the drives have the corresponding reserves. Failure to adhere to the limits can lead to damage to the mechanical parts and/or error messages.

8.3.14 Traversing with feedforward control: FFWON, FFWOF

Functionality

The feedforward control reduces the following error in the traversed path towards zero. Traversing with feedforward control provides higher path accuracy and thus better manufacturing results.

Programming

```
FFWON              ; Feedforward control ON
FFWOF              ; Feedforward control OFF
```

Programming example

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

8.3.15 3rd or 4th axis

Functionality

Depending on the machine design, a 3rd and even a 4th can be required. This axis can be designed either as a linear or as a rotary axis (e.g. rotary axis for the tool magazine). The names for these axes are specified by the machine manufacturer (e.g. U, C or A).

For rotary axes, the traversing range can be configured between 0 ...<360 degrees (modulo behavior).

With an appropriate machine design, the 4th axis can be traversed linear simultaneously with the remaining axes. If the axis is traversed together with the remaining axes (X, Y) in a block that contains G1 or G2/G3, no component of the feedrate F is assigned to this axis; its velocity will depend on the time for traveling along the X and Y axes, and its motion starts and ends with the remaining path axes. The velocity, however, cannot be greater than the defined limit value.
When traversed separately with G1 in the block, the axis will traverse with the activate feedrate F. If it is a rotary axis, the unit for F will be correspondingly degrees/min with G94 or degrees/revolution of the spindle with G95.
For these axes, offsets can be set (G54 ... G59) and programmed (TRANS, ATRANS).

Programming example

Supposed the 4th axis is a rotary axis and the name of the axis is A:

```
N5 G94                        ; F in mm/min or degrees/min
N10 G0 X10  Z30 A45           ; Traverse X-Z path at rapid traverse, and A at the same time
N20 G1 X12  Z33 A60 F400      ; Traverse X-Z path with 400 mm/min, A at the same time
N30 G1 A90 F3000              ; A axis traverses to 90 deg. position at a
                                velocity of 3,000 deg./min
```

Special instructions for rotary axes: DC, ACP, ACN

e.g. for rotary axis A:

```
A=DC(...)          ; Abs. dimensioning; approach position directly (on the shortest path)
A=ACP(...)         ; Absolute dimensioning; approach position in the positive direction
A=ACN(...)         ; Absolute dimensioning; approach position in the negative direction
```

Example:

```
N10 A=ACP(55.7)    ; Absolute position 55.7 degrees; approach position in the positive
direction
```

8.3.16 Dwell time: G4

Functionality

You can interrupt the machining for a defined time between two NC blocks by inserting a **separate block** with G4, e.g. backing off.
The words with F... or S... are for the time specifications in this block only. Any feedrate F programmed beforehand is maintained.

Programming

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

Programming example

```
N5 G1 F200       ; Feedrate F,
N10 G4 F2.5      ; Dwell time 2.5 s
N20 Z70
N30 G4           ;  and 100 %, speed override:  t=0.1 min
N40 X...         ; Feed continues to be active
```

8.3.17 Coupled motion: TRAILON, TRAILOF

Note

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

Function

When a defined leading axis is moved, the trailing axes (= following axes) assigned to it traverse through the distances described by the leading axis, allowing for a coupling factor.

Together, the leading axis and following axis represent coupled axes.

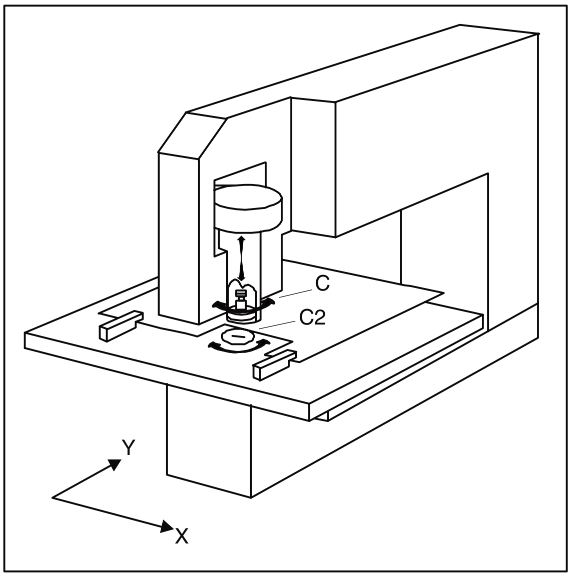


Fig. 8-31 Coupled axes (leading axis C, following axis C2)

Field of application

Traversal of an axis by means of a simulated axis. The leading axis is a simulated axis and the coupled axis a real axis. In this way, the real axis can be traversed as a function of the coupling factor.

Leading axis	C
Coupled-motion axis	C2

Programming

TRAILON(FAxis, LAxis, Couple)

or

TRAILOF(Faxis,Laxis,LAxis2)

or deactivation without specifying the leading axis

TRAILOF(FAxis)

TRAILON and `TRAILOF` act modal.

Parameters

TRAILON	Activating and defining a coupled-axis grouping Example: C2 = coupled-motion axis, C = leading axis TRAILON(C2,C)
TRAILOF	Deactivate coupled-axis grouping TRAILOF(C2,C) Deactivate coupling to leading axis C. TRAILOF(C2) Deactivate the coupling without specifying the leading axis.
FAchse	Axis name of the coupled-motion axis A coupled-motion axis can also act as the leading axis for other coupled-motion axes. In this way, it is possible to create a range of different coupled–axis groupings.
LAchse	Axis name of the leading axis
Couple	Coupling factor = Path of the coupled-motion axis/path of the leading axis Default = 1

Note

The coupled motion is performed in the machine coordinate system (MCS).

The number of coupled-axis groupings which may be simultaneously activated is limited only by the maximum possible number of combinations of axes on the machine.

8.3.18 Activating, deactivatingprotection zones: CPROT

Function

Activating and preactivating previously defined protection zones for collision monitoring and deactivating protection zones.

The maximum number of protection zones, which can be active simultaneously on the same channel, is defined in machine data.

If no toolrelated protection zone is active, the tool path is checked against the workpiece-related protection zones.

Note

If no workpiece-oriented protection zone is active, protection-zone monitoring does not take place.

Programming

CPROT (n, state, xMov, yMov)

Parameters

CPROT	Call of channel-specific protection zones
n	Number of protection zone
state	Status specification 0 = Deactivate protection zone 1 = Preactivate protection zone 2 = Activate protection zone 3 = Preactivate protection zone with conditional stop
xMove, yMove	Moving the protection zone already defined along the geometry axes

8.4 Tool T

Functionality

The tool selection takes place when the T word is programmed. Whether this is a **tool change** or only a **preselection**, is defined in the machine data:

- The tool change (tool call) is performed either directly using the T word or
- the change takes place after the preselection with the T word by an additional instruction **M6** (see also Section 8.5 “Miscellaneous functions M”).

Please note:
If a certain tool has been activated, it also remains stored as the active tool even after the end of the program and after turning off/turning on the control system.
If you change a tool manually, enter the change also in the control system to make sure that the control system knows the relevant tool. For example, you can start a block with the new T word in the MDA mode.

Programming

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

Note

The following is the maximum that can be stored simultaneously in the control system:

• SINUMERIK 802D sl plus:	64 tools
• SINUMERIK 802D sl pro:	128 tools.

Programming example

```
; Tool change without M6:
N10 T1                  ; Tool 1
...
N70 T588                ; Tool 588

; Tool change using M6:
N10 T14  ...            ; Preselect tool 14
...
N15 M6                  ; Carry out tool change; thereafter, T14 is active
```

8.5 Miscellaneous function (M)

Functionality

The special function M initiates switching operations, for example, such as “Coolant ON/ OFF” and other operations on the machine.

Various M functions have already been assigned a fixed functionality by the CNC manufacturer. The functions not yet assigned fixed functions are reserved for free use of the machine manufacturer.

Note:
An overview of the M functions used and reserved in the control system can be found in Subsection 8.1.6 “List of instructions”.

Programming

```
M...            ; Max. 5 M functions per block
```

Effect

Effect in blocks with axis movements:
If the functions **M0, M1, M2** are contained in a block with traversing movements of the axes, then these M functions become effective **after the traversing movements.**

The remaining M functions are output to the PLC **with** the traversing movements.

If you would like to program an M function directly before or after an axis movement, insert a separate block with this M function. **Please note**: This block interrupts the G64 continuous–path control mode and generates exact stop.

Programming example

```
N10 ...
N20 X...                    ; M function in a block with axis movement
N180 M78 M67 M10 M12 M37    ; Max. 5 M functions per block
```

Note

In addition to M and H functions, T, D, and S functions can also be transferred to the PLC (programmable logic controller). In all, a maximum of 10 such function outputs are possible in a block.

8.6 H function

Functionality

H functions can also be used to transfer floating point data (REAL data type, such as arithmetic parameters, see Section “Arithmetic parameters R”) to the PLC.
Which meanings the values have for a certain H function is defined by the machine manufacturer.

Programming

```
H0=...   to    H9999=...              ; Max. 3 H functions per block
```

Programming example

```
N10 H1=1.987 H2=978.123 H3=4        ; 3 H functions in a block
N20 G0 X71.3 H99=–8978.234          ; with axis motions in the block
N30 H5                              ;  corresponds to: H0=5.0
```

Note

In addition to M and H functions, T, D, and S functions can also be transferred to the PLC (programmable logic controller). In all, a maximum of 10 such function outputs are possible in a block.

8.7 Arithmetic parameters R, LUD and PLC variables

8.7.1 Arithmetic parameters R

Functionality

The arithmetic parameters are used if an NC program is not only to be valid for values assigned once, or if you must calculate values. The required values can be set or calculated by the control system during program execution.
The arithmetic parameter values can also be set by operator inputs. If values have been assigned to the arithmetic parameters, they can be assigned to other NC addresses in the program with variable values.

Programming

```
R0=... to R299=...      ; Assign values to the arithmetic parameters
R[R0]=...               ; Indirect programming: Assign the arithmetic parameter R whose
                          number is to be found, e.g. in R0 a value
X=R0                    ; Assign the NC addresses an arithmetic parameter, e.g. the X axis
```

Value assignments

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

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

The decimal point can be omitted for integer values. A plus sign can always be omitted.

Example:
R0=3.5678 R1=–37.3 R2=2 R3=–7 R4=–45678.123

Use the **exponential notation** to assign an extended range of numbers:

± (10^{-300} ... 10^{+300}).

The value of the exponent is written after the **EX** characters; maximum total number of characters: 10 (including the leading sign and decimal point)
Value range of EX: –300 to +300

Example:

```
R0=–0.1EX–5            ; Meaning: R0 = –0.000 001
R1=1.874EX8            ; Meaning: R1 = 187 400 000
```

Remark: There can be several assignments in one block incl. assignments of arithmetic expressions.

Assignments to other addresses

The flexibility of an NC program lies in assigning these arithmetic parameters or expressions with arithmetic parameters to other NC addresses. Values, arithmetic expressions and arithmetic parameters can be assigned to all addresses; **Exception: The addresses N, G, and L**.

When assigning, write the “ = ” sign after the address character. It is also possible to have an assignment with a minus sign.

A separate block is required for assignments to axis addresses (traversing instructions).

Example:

```
N10 G0 X=R2                    ; Assignment to the X axis
```

Arithmetic operations/arithmetic functions

When operators/arithmetic functions are used, it is imperative to use conventional mathematical notation. Machining priorities are set using round brackets. Otherwise, the following rule applies: Multiplication and division first, and then addition and subtraction.
The trigonometric functions are to be specified using degrees.
For permissible arithmetic functions, please refer to: see Section “List of instructions”

Programming example: Calculating with R parameters

```
N10 R1= R1+1                   ; The new R1 is obtained from the old R1 plus 1
N20 R1=R2+R3   R4=R5–R6   R7=R8* R9   R10=R11/R12
N30 R13=SIN(25.3)              ; R13 results in a sine value of 25.3 degrees
N40 R14=R1*R2+R3               ; Note the order of operations R14=(R1*R2)+R3
N50 R14=R3+R2*R1               ; Same result as for block N40
N60 R15=SQRT(R1*R1+R2*R2)      ; Explanation:  $R15 = \sqrt{R1^2 + R2^2}$
N70 R1= –R1                    ; The new R1 value is the negative old R1 value
```

Programming example: Assign R parameters to the axes

```
N10 G1 G91 X=R1 Y=R2 F300      ; Separate blocks (traversing blocks)
N20  Y=R3
N30 X= –R4
N40 Y= SIN(25.3)–R5            ; with arithmetic operations
...
```

Programming example: Indirect programming

```
N10 R1=5                       ; Assign value 5 (integer) directly to R1
...
N100 R[R1]=27.123              ; Assign value 27.123 indirectly to R5
```

8.7.2 Local User Data (LUD)

Functionality

The operator/programmer (user) can define his/her own variable in the program from various data types (LUD = Local User Data). These variables are only available in the program in which they were defined. The definition takes place immediately at the start of the program and can also be associated with a value assignment at the same time. Otherwise, the initial value is zero.
The name of a variable can be defined by the programmer himself. The naming is subject to the following rules:

- A maximum of 32 characters can be used.
- It is imperative to use letters for the first two characters; the remaining characters can be either letters, underscore or digits.
- Do not use a name already used in the control system (NC addresses, keywords, names of programs, subroutines, etc.).

Programming / data types

```
DEF BOOL  varname1        ; "Bool" type, values: TRUE (=1), FALSE (=0)
DEF CHAR  varname2        ; "Char" type, 1 character in the ASCII code: „a„, „b„, ...
                          ; Numerical code value: 0 ... 255
DEF INT  varname3         ; Integer type, integer values, 32-bit value range:
                          ;  –2 147 483 648 to +2 147 483 648 (decimal)
DEF REAL  varname4        ; "Real" type, natural number (like arithmetic parameters R),
                          ; Range of values:  ±(0.000 0001 ... 9999 9999)
                          ; (8 decimal places and leading sign and decimal point) or
                          ; Exponential notation: ± ( 10^–300  ... 10^+300 )
DEF STRING[string length]  varname41   ; STRING type,  [string length]: Maximum number
of characters
```

Each data type requires its own program line. However, several variables of the same type can be defined in one line.

Example:

```
DEF INT PVAR1, PVAR2, PVAR3=12, PVAR4      ; 4 variables of the INT type
```

Example for the STRING type with assignment:

```
DEF STRING[12]  PVAR="Hello"               ; Define PVAR variable with maximum
                                            character length 12 and assign string Hello
```

Fields

In addition to individual variables, it is also possible to define one- or two-dimensional variable fields of these data types:

```
DEF INT PVAR5[n]              ; One-dimensional field of the INT type, n: Integer
DEF INT PVAR6[n,m]            ; Two-dimensional field of the INT type, n, m: Integer
```

Example:

```
DEF INT PVAR7[3]              ; Field with 3 elements of the INT type
```

Within the program, the individual field elements can be reached via the field index and can be treated like individual variables. The field index runs from 0 to a small number of the elements.

Example:
N10 PVAR7[2]=24 ; The third field element (with index 2) is assigned the value of 24.

Value assignment for a field having a SET instruction:
N20 PVAR5[2]=SET(1,2,3) ; From the 3rd field element on, different values are assigned.

Value assignment for a field having a REP instruction:
N20 PVAR7[4]=REP(2) ; Starting from field element [4], all elements are assigned the same value, here 2.

8.7.3 Reading and writing PLC variables

Functionality

To allow rapid data exchange between NC and PLC, a special data area exists in the PLC user interface with a length of 512 bytes. In this area, PLC data are compatible in data type and position offset. In the NC program, these compatible PLC variables can be read or written.
To this end, special system variables are provided:

```
$A_DBB[n]     ; Data byte  (8-bit value)
$A_DBW[n]     ; Data word (16-bit value)
$A_DBD[n]     ; Data double-word (32-bit value)
$A_DBR[n]     ; REAL data   (32-bit value)
```

“n” stands for the position offset (beginning of data area to beginning of variable) in bytes

Example:
R1=$A_DBR[5] ; Read REAL value, offset 5 (starts at byte 5 of the range)

Information

- The reading of variables generates a preprocessing stop (internal STOPRE).
- A maximum of 3 variables can be written simultaneously (in a block).

8.8 Program jumps

8.8.1 Jump destination for program jumps

Functionality

A **label** or a **block number** serve to mark blocks as jump destinations for program jumps. Program jumps can be used to branch to the program sequence.

Labels can be freely selected, but must contain a minimum of 2 and a maximum of 8 letters or numbers, and the **first two characters** must be **letters** or underscores.

Labels that are in the block that serves as the jump destination are **ended by a colon**. They are always at the start of a block. If a block number is also present, the label is located **after the block number**.

Labels must be unique within a program.

Programming example

```
N10 LABEL1: G1 X20                    ; LABEL1 is the label, jump destination
...
TR789: G0 X10 Z20                     ; TR789 is the label; jump destination
                                        – no block number exists
N100 ...                              ; Block number can be jump destination
...
```

8.8.2 Unconditional program jumps

Functionality

NC programs process their blocks in the sequence in which they were arranged when they were written.

The processing sequence can be changed by introducing program jumps.

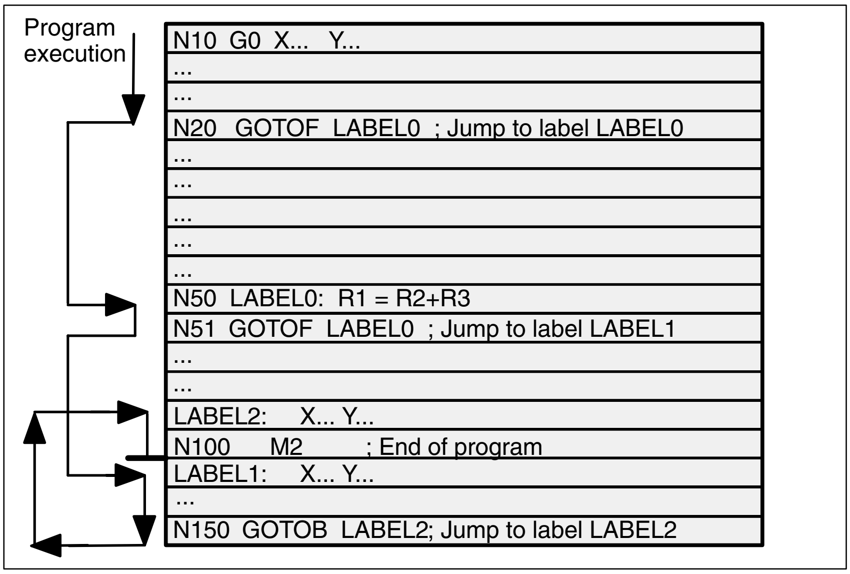
The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

The unconditional jump instruction requires a separate block.

Programming

```
GOTOF Label     ; Jump forward (in the direction of the last block in the program)
GOTOB Label     ; Jump backwards (in the direction of the first block in the program)

Label           ; Selected string for the label (jump label) or for the block number
```



```
N10  G0  X...   Y...
...
...
N20   GOTOF  LABEL0  ; Jump to label LABEL0
...
...
...
...
...
N50  LABEL0:  R1 = R2+R3
N51  GOTOF  LABEL0  ; Jump to label LABEL1
...
...
LABEL2:     X... Y...
N100      M2         ; End of program
LABEL1:     X... Y...
...
N150  GOTOB  LABEL2; Jump to label LABEL2
```

Fig. 8-32 Unconditional jumps using an example

8.8.3 Conditional program jumps

Functionality

Jump conditions are formulated after the **IF instruction**. If the jump condition (**value not zero**) is satisfied, the jump takes place.
The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

Conditional jump instructions require a separate block. Several conditional jump instructions can be located in the same block.

By using conditional program jumps, you can also considerably shorten the program, if necessary.

Programming

IF *condition* GOTOF *label* ; Jump forward
IF *condition* GOTOB *label* ; Jump backwards

GOTOF	; Jump direction forward (in the direction of the last block in the program)
GOTOB	; Jump direction backwards (in the direction of the first block in the program)
Label	; String selected for the label (jump label) or block number
IF	; Initiation of the jump condition
condition	; Arithmetic parameters, arithmetic expression for formulating the condition

Comparison operations

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

The comparison operations support formulating of a jump condition. Arithmetic expressions can also be compared.

The result of comparison operations is “satisfied” or “not satisfied.” “Not satisfied” sets the value to zero.

Programming example for comparison operators

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

Programming example

```
N10 IF R1 GOTOF  LABEL1              ; If R1 is not equal to zero, then go to the block with LABEL1
...
N90 LABEL1: ...
N100 IF R1>1 GOTOF  LABEL2           ; If R1 is greater than 1, then go to the block with LABEL2
...
N150 LABEL2: ...
...
N800 LABEL3: ...
...
N1000 IF R45==R7+1 GOTOB  LABEL3     ; If R45 is equal to R7 plus 1, then go to the block with
                                     ; LABEL3
...
```

Several conditional jumps in the block:

```
N10 MA1: ...
...
N20 IF R1==1 GOTOB  MA1  IF R1==2 GOTOF  MA2 ...
...
N50 MA2: ...
```

Remark: The jump is executed for the first fulfilled condition.

8.8.4 Program example for jumps

Task

Approaching points on a circle section:

Given:	Starting angle:	30°	in R1
	Circle radius:	32 mm	in R2
	Distance between the positions:	10°	in R3
	Number of points:	11	in R4
	Position of the circle center point in Z:	50 mm	in R5
	Position of the circle center point in X:	20 mm	in R6

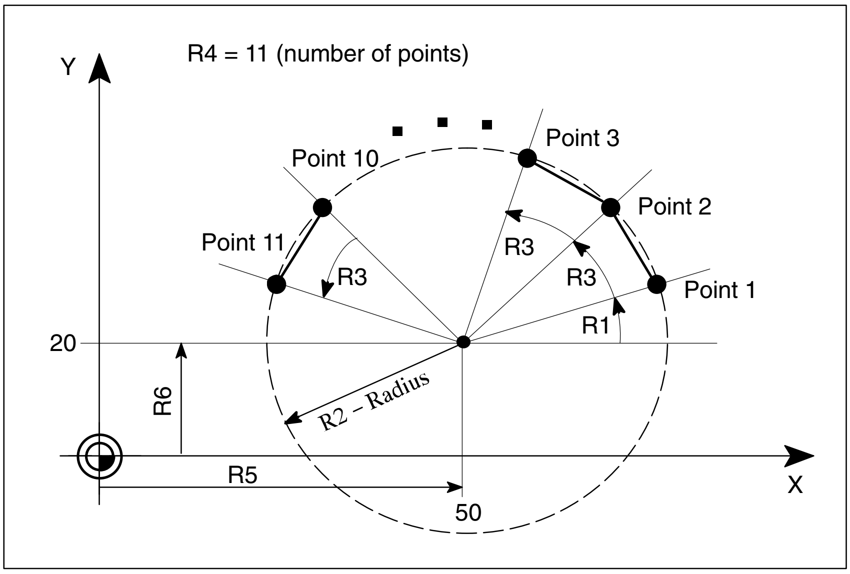


Fig. 8-33 Linear approach of points on a circle segment

Programming example

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

Explanation

In block N10, the starting conditions are assigned to the corresponding arithmetic parameters. The calculation of the coordinates in X and Y and the processing takes place in N20.

In block N30, R1 is incremented by the clearance angle R3; R4 is decremented by 1.
If R4 > 0, N20 is reprocessed; otherwise, N50 with end of program.

8.9 Subroutine technique

Usage

Basically, there is no difference between a main program and a subroutine.

Frequently recurring machining sequences are stored in subroutines, e.g. certain contour shapes. These subroutines are called at the appropriate locations in the main program and then executed.

One form of a subroutine is the **machining cycle**. Machining cycles contain universally valid machining scenarios. By assigning values via included transfer parameters, you can adapt the subroutine to your specific application.

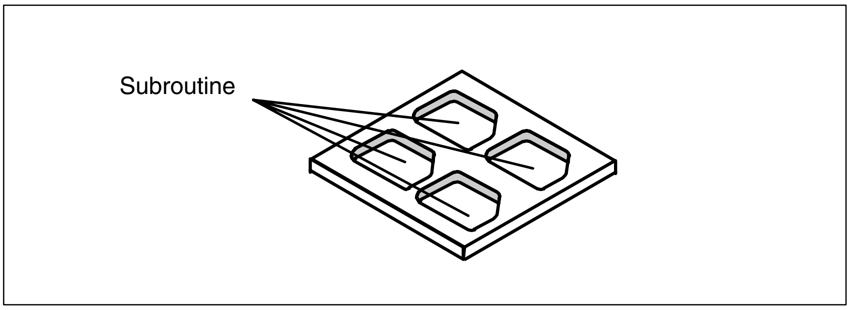


Fig. 8-34 Example for using a subroutine for a workpiece four times

Program structure

The structure of a subroutine is identical to that of a main program (see Subsection 8.1.2 “Program structure”). Like main programs, subroutines contain **M2 (end of program)** in the last block of the program sequence. This means a return to the program level where the subroutine was called from.

End of program

The end instruction **RET** can also be used instead of the M2 program end in the subroutine.

RET must be programmed in a separate block.

The RET instruction is used when G64 continuous-path mode is not to be interrupted by the return. With M2, G64 is interrupted and exact stop is initiated.

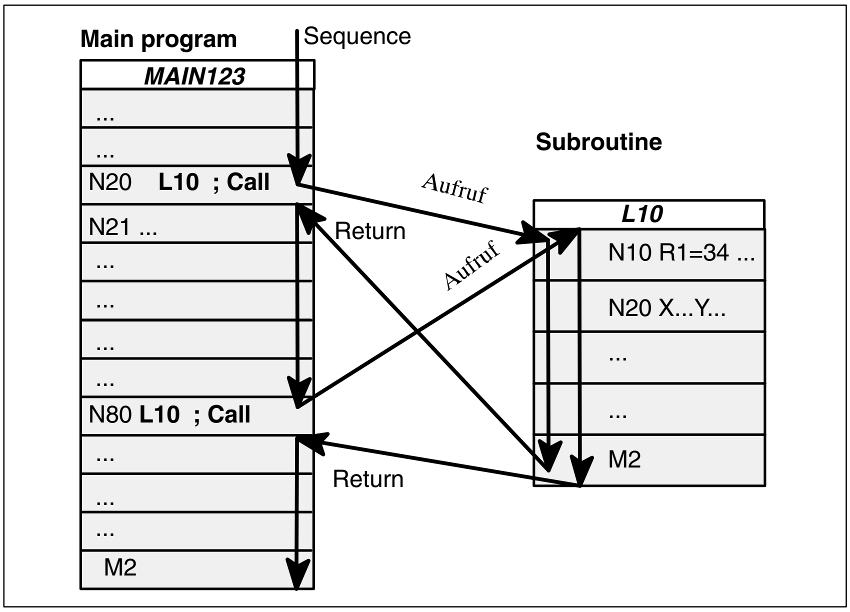


Fig. 8-35 Example of sequence when calling a subroutine twice

Subroutine name

The subprogram is given a unique name allowing it to be selected from several subroutines. When you create the program, the program name may be freely selected provided the following conventions are observed:

The same rules apply as for the names of main programs.

Example: **LRAHMEN7**

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

Please observe: With address L, leading zeros are meaningful for differentiation.

Example: **L128** is not **L0128** or **L00128** !
These are 3 different subroutines.

Subroutine call

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

Example:

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

Program repetition P...

If a subroutine is to be executed several times in succession, write the number of times it is to be executed in the block of the call after the subroutine name under the **address P**. A maximum of **9,999 cycles** are possible (P1 ... P9999).

Example:

```
N10 L785 P3                    ; Call of subroutine L785, 3 cycles
```

Nesting depth

Subroutines can also be called from a subroutine, not only from a main program. In total, up to **8 program levels** are available for this type of nested call, including the main program level.

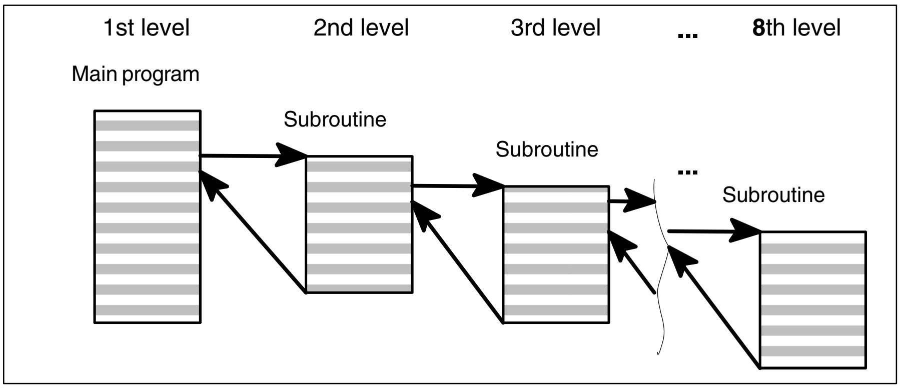


Fig. 8-36 Execution with 8 program levels

Information

Modal G functions can be changed in the subroutine, e.g. G90 –> G91. When returning to the calling program, ensure that all modal functions are set the way you need them to be.

Please make sure that the values of your arithmetic parameters used in upper program levels are not inadvertently changed in lower program levels.

8.10 Timers and workpiece counters

8.10.1 Runtime timers

Functionality

The timers are provided as system variables ($A...) which can be used either for monitoring of the technological processes in the program or only in the display.
These timers can only be read. There are timers that are always active. Others can be deactivated via machine data.

Timers – always active

- **$AN_SETUP_TIME** – Time since the last “Control boot with default values” (in minutes)
  It is automatically set to zero upon “Control boot with default values”.
- **$AN_POWERON_TIME** – Time since the last booting of the control (in minutes)
  It is automatically set to zero upon every booting of the control.

Timers that can be deactivated

The following timers are activated via machine data (default setting).
The start is timer-specific. Each active execution time measurement is interrupted automatically in the stopped program status or at feedrate override =zero.
The behavior of the activated time measurements in active dry run feedrate and program test can be defined by machine data.

- **$AC_OPERATING_TIME** – Total operating time of NC programs in the AUTOMATIC mode (in seconds)
  The operating times of all programs between NC start and program end / reset are totaled in the AUTOMATIC mode. The timer is zeroed with each power-up of the control system.
- **$AC_CYCLE_TIME** – Operating time of the selected NC program (in seconds)
  The runtime between NC start and program end / reset is measured in the selected NC program. The timer is reset with the start of a new NC program.
- **$AC_CUTTING_TIME** – Tool operating time (in seconds):
  The runtime of the path axes without active rapid traverse in all NC programs between NC start and program end / reset with active tool is measured.
  The measurement is also interrupted with active dwell time.
  The timer is zeroed automatically at a “Controller run-up with default values”.

Programming example

```
N10 IF $AC_CUTTING_TIME>=R10 GOTOF WZZEIT      ; Tool action time limit value
...
N80 WZZEIT:
N90 MSG("Tool action time: Limit value reached")
N100 M0
```

Display

The contents of the active system variables is displayed on the screen under <OFFSET PARAM> <Setting data> <Timers/counters>:

Total runtime	= $AC_OPERATING_TIME
Time of program execution	= $AC_CYCLE_TIME
Time of feedrate action	= $AC_CUTTING_TIME
Time since cold start	= $AN_SETUP_TIME
Time since warm restart	= $AN_POWERON_TIME

"Part count" is visible in addition in the AUTOMATIC mode in the "Position" operating area in the information line.

8.10.2 Workpiece counter

Functionality

The "Workpiece counter" function provides counters which can be used for counting of workpieces.
These counters exist as system variables with read-write access from the program or through operation (observe protection level for writing!).
The counter activation, the time of zeroing and the counter algorithm can be influenced via machine data.

Counters

- **$AC_REQUIRED_PARTS** – Number of required workpieces (workpiece target)
  The number of workpieces at which the number of current workpieces $AC_ACTUAL_PARTS is set to zero can be defined in this counter.
  The generation of the display alarm 21800 "Workpiece setpoint reached" can be activated via machine data.
- **$AC_TOTAL_PARTS** – Total number of workpieces actually produced (total actual)
  The counter indicates the total number of workpieces produced since the starting time.
  The counter is automatically set to zero upon every booting of the control system.
- **$AC_ACTUAL_PARTS** – Number of current workpieces (current actual)
  This counter records the number of all workpieces produced since the starting time.
  When the workpiece setpoint is reached ($AC_REQUIRED_PARTS, value greater than zero), the counter is automatically zeroed.

- **$AC_SPECIAL_PARTS** – Number of workpieces specified by the user
  This counter allows user-defined workpiece counting. Alarm output can be defined for the case of identity with
  $AC_REQUIRED_PARTS (workpiece target). Users must reset the counter themselves.

Programming example

```
N10 IF $AC_TOTAL_PARTS==R15 GOTOF SIST      ; Required number of workpieces reached?
...
N80 SIST:
N90 MSG("Workpiece target reached")
N100 M0
```

Display

The contents of the active system variables is displayed on the screen under <OFFSET PARAM> <Setting data> <Timers/counters>:

Parts, total	= $AC_TOTAL_PARTS
Parts requested	= $AC_REQUIRED_PARTS
Number of parts	= $AC_ACTUAL_PARTS
	$AC_SPECIAL_PARTS Cannot be displayed

“Part count” is visible in addition in the AUTOMATIC mode in the “Position” operating area in the information line.

8.11 Language commands for tool monitoring

8.11.1 Overview of tool monitoring

Functionality

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

The tool monitoring is activated via machine data.
The following monitoring types are possible for the active cutting edge of the active tool:

- Monitoring of the **tool life**
- Monitoring of the **workpiece count**

The above-mentioned types of monitoring can be activated for one tool (T) simultaneously.

The control / data input for the tool monitoring is preferably done through the operation. In addition, various functions can be programmed.

Monitoring counter

Monitoring counters are provided for each monitoring type.
The monitoring counters run from a set value > 0 towards zero. Once a monitoring counter has reached the value ≤ 0, the limit value is deemed to be reached. A corresponding alarm message is issued.

System variable for type and condition of the monitoring

- **$TC_TP8[t]** ; Status of the tool with the number t:
  Bit 0 =1: Tool is **active**
  =0: Tool is not active

  Bit 1 =1: Tool is **enabled**
  =0: not enabled

  Bit 2 =1: Tool is **disabled**
  =0: not disabled

  Bit 3 : reserved

  Bit 4 =1: **Warning limit reached**
  =0: not reached
- **$TC_TP9[t]** ; Type of the monitoring function for the tool with the number t:
  = 0: No monitoring
  = 1: (Tool) life-monitored tool
  = 2: Quantity-monitored tool

These system variables can be read/written in the NC program.

System variables for tool monitoring data

Table 8-3 Tool monitoring data

Identifier	Description	Data type	Default value
$TC_MOP1[t,d]	Warning limit for tool life in minutes	REAL	0.0
$TC_MOP2[t,d]	Residual tool life in minutes	REAL	0.0
$TC_MOP3[t,d]	Warning limit for count	INT	0
$TC_MOP4[t,d]	Residual unit quantity	INT	0
...	...		
$TC_MOP11[t,d]	Tool life setpoint	REAL	0.0
$TC_MOP13[t,d]	Unit quantity setpoint	INT	0

t for tool number T, d for D number

System variables for active tool

The following can be read in the NC program via system variables:

- $P_TOOLNO – number of the active tool T
- $P_TOOL – active D number of the active tool

8.11.2 Tool life monitoring

Tool life monitoring is performed for the tool cutting edge that is currently in use (active cutting edge D of the active tool T).
Once the axes traverse (G1,G2, G3, ... but not for G0), the residual tool life ($TC_MOP2[t,d]) of this tool cutting edge is updated. When the remaining tool life of a cutting edge is taken as the “Tool life warning” ($TC_MOP1[t,d]) value during machining it is indicated to the PLC using an interface signal.
If the remaining tool life is ≤ 0, an alarm is issued and an additional interface signal set. The tool changes to the “disabled” condition and cannot be programmed again until this condition changes. The operator must intervene: The operator must change the tool or ensure that he has an operational tool for machining.

$A_MONIFACT system variable

The **$A_MONIFACT** system variable (REAL data type) allows the monitoring clock to be run slower or faster. This factor can be set before using the tool, in order to take the different kinds of wear into consideration according to the workpiece material used, for example.

After booting of the control system, reset / program end, the factor $A_MONIFACT has the value 1.0. Real-time operation applies.
Calculation examples:

$A_MONIFACT=1	1 minute real time = 1 minute tool life that is decremented
$A_MONIFACT=0.1	1 minute real time = 0.1 minute tool life that is decremented
$A_MONIFACT=5	1 minute real time = 5 minutes tool life that are decremented

Setpoint update with RESETMON( )

The RESETMON(state, t, d, mon) function sets the actual value to the setpoint:

– for all or only one certain edge of a certain tool
– for all or only a certain monitoring types.

Transfer parameter:

INT state Status of command execution:
= 0 Successful execution
= –1 The cutting edge with the specified D number d does not exist.
= –2 The tool with the specified T number t does not exist.
= –3 The specified tool t does not possess a defined monitoring function.
= –4 The monitoring function is not activated; the command is not executed.

INT t Internal T number:
= 0 For all tools
<> 0 For this tool (t < 0 : Absolute-value generation |t|)

INT d *optional:* D number of the tool with the number t:
> 0 For this D number
Without d / = 0 All cutting edges of tool t

INT mon *optional:* Bit-coded parameter for the monitoring type (values analogous to $TC_TP9):
= 1: Tool life
= 2: Count
Without mon or = 0: **All** actual values of the monitoring functions active for tool t are set to their setpoints.

Notes:

– RESETMON( ) has no effect when “Program test” is active.

– The variable for the **state** status feedback must be defined at the beginning of the program using a DEF statement: DEF INT state
You can also define a different name for the variable (instead of state, with a maximum of 15 characters, beginning with 2 letters). The variable is only available in the program if it was defined in this program.
This also applies to **mon.** If a data is required at all, this can also be directly transferred as a number (1 or 2).

8.11.3 Workpiece count monitoring

The workpiece count of the active cutting edge of the active tool is monitored.
The workpiece count monitoring records all tool cutting edges used to produce a workpiece. If the count is changed due to new values specified, the monitoring data are adapted to all tool cutting edges that became active since the last workpiece count.

Updating the workpiece count by operator input or SETPIECE( )

The workpiece count can be updated by an operator input (HMI) or in the NC program through the SETPIECE( ) language command.

With the **SETPIECE** function, the user can update the workpiece count monitoring data of the tools involved in the machining process.
If SETPIECE(n) is programmed, the internal setpiece memory is searched. If this “memory” is set for one cutting edge of a tool, the part quantity (remaining part quantity – $TC_MOP4) of the respective cutting edge is decremented by the specified value and the respective “memory” (setpiece memory) is deleted.

SETPIECE(n)
n : = 0... 32000 Number of workpieces that have been produced since the last execution of the
SETPIECE function. The counter status for the remaining part quantity ($TC_MOP4[t,d])
is reduced by this value.

Programming example

```
N10 G0 X100
N20 ...
N30 T1
N40 M6
N50 D1
...                          ; Machining using T1, D1
N60 SETPIECE(1)              ; $TC_MOP4[1,1 ] (T1,D1) is decremented by 1
N90 T2
N100 M6
N110 D2
...                          ; Machining using T2, D2
N200 SETPIECE(1)             ; $TC_MOP4[2,2 ] (T2,D2) is decremented by 1
...
N300 M2
```

Notes:

- The SETPIECE( ) command is not active during the block search.
- Direct writing of $TC_MOP4[t,d] is recommended only in simple cases. A subsequent block with the STOPRE command is required.

Setpoint refreshing

Setpoint refreshing, i.e. setting the remaining workpiece counters ($TC_MOP4[t,d]) to the workpiece count setpoint ($TC_MOP13[t,d]) is typically performed via operator input (HMI). It can, however, also be performed through the RESETMON (state, t, d, ,mon) function as already described for the tool life monitoring.

Example:

```
DEF INT state                    ; Define variable for status feedback at the beginning of the pro-
gram
...
N100 RESETMON(state,12,1,2)      ; Workpiece counter setpoint refreshed for T12, D1; setpoint 2
...
```

Programming example

```
DEF INT state                                 ; Define variable for the status checkback mes-
sage from
                                                RESETMON()
;
G0 X...                                       ; Retract
T7                                            ; Load new tool using M6 as necessary
$TC_MOP3[$P_TOOLNO,$P_TOOL]=100               ; Warning limit, 100 workpieces
$TC_MOP4[$P_TOOLNO,$P_TOOL]=700               ; Remaining quantity
$TC_MOP13[$P_TOOLNO,$P_TOOL]=700              ; Required quantity (setpoint)
; Activation after setting:
$TC_TP9[$P_TOOLNO,$P_TOOL]=2                  ; Activate monitoring of required quantity, active
tool
STOPRE
ANF:
BEARBEIT                                      ; Subroutine for workpiece machining
SETPIECE(1)                                   ; Refresh counter
M0                                            ; Next workpiece; press NC START to continue
IF ($TC_MOP4[$P_TOOLNO,$P_TOOL]]>1) GOTOB ANF
MSG("Tool T7 worn – please change")
M0                                            ; After tool change, press NC START to continue
RESETMON(state,7,1,2)                         ; Workpiece counter setpoint refreshed
IF (state<>0) GOTOF ALARM
GOTOB ANF
ALARM:                                        ; Display errors:
MSG("error RESETMON: " <<state)
M0
M2
```

8.12 Language commands for nibbling and punching

Functionality

Punching and nibbling functions are activated and deactivated via configurable language commands.

Groups

The language commands are subdivided into the following groups:

Group 35	
The actual punching and nibbling-specific functions are activated and deactivated by means of the following language commands:	
PON	= punching ON
SON	= nibbling ON
PONS	= punching ON, activated in the position controller cycle
SONS	= nibbling ON, activated in the position controller cycle
SPOF	= punching/nibbling OFF

Group 36	
This group includes the commands which have only a preparatory character and which determine the real nature of the punching function:	
PDELAYON	= punching with delay ON
PDELAYOF	= punching with delay OFF
Since the PLC normally needs to perform some preliminary tasks with respect to these preparatory functions, they are programmed before the activating commands.	

Group 38	
This group contains the commands for switching over to a second punch interface. It can be used, for example, for a second punching unit or set of hammer shears. A second I/O pair which can be used for punching functionality is defined via machine data.	
SPIF1	= first interface is active

Note

Only one function at a time can be active within a G code group (similar, for example, to the various interpolation modes G0, G1, G2, G3, etc., which are also mutually exclusive).

SPOF

Punching and nibbling OFF

The SPOF function terminates all punching and nibbling functions. In this state, the NCK responds neither to the “Stroke active” signal nor to the PLC signals specific to punching and nibbling functions.

If SPOF is programmed together with a travel command in one block (and in all further blocks, if punching/nibbling is not activated with SON or PON), the machine approaches the programmed position without the initiation of a punching operation. SPOF deselects SON, SONS, PON and PONS and is equivalent to a RESET state.

Programming example for SPOF

```
N20 G90 X100 SON              ; Activate punching
N25 X50 SPOF                  ; Deactivate punching; positioning without stroke initiation
```

SON

Nibbling ON

SON activates the nibbling function and deselects the other functions in G group 35 (e. g. PON).

In contrast to punching, the first stroke is made at the starting point of the block with the activating command, i. e. before the first machine motion is performed.

SON has a modal action, i. e. it remains active until either SPOF or PON is programmed or until the program end is reached.

The stroke initiation is suppressed in blocks without traversing information relating to the axes designated as punching or nibbling axes (typically those in the active plane). If a stroke nevertheless is to be initiated, then one of the punching/nibbling axes must be programmed with 0 traversing path. If the first block with SON is a block without traversing information of the type mentioned, then only one stroke takes place in this block since the start and end points are identical.

Programming example for SON

```
;
N70 X50 SPOF                  ; Positioning without punching initiation
N80 X100 SON                  ; Activate nibbling; stroke initiation before the
                                motion (X=50) is started and at the end of the programmed
                                motion (X=100)
```

SONS

Nibbling ON (in the position controller cycle)

SONS behaves in the same way as SON. The function is activated in the position control cycle, thus allowing time-optimized stroke initiation and an increase in the punching rate per minute.

PON

Punching ON

PON activates the punching function and deactivates SON.

PON has a modal action like SON.

In contrast to SON, however, a stroke is only executed at the end of the block or, in the case of automatic path segmentation, at the end of a path segment. PON has an identical action to SON in the case of blocks which contain no traversing information.

Programming example for PON

```
;
N100 Y30 SPOF                 ; Positioning without stroke initiation
N110 X100 PON                 ; Activate punching, stroke initiation at the end of the
                                positioning process (X=100)
```

PONS Punching ON (in the position controller cycle)

PONS behaves in the same way as PON. For explanation, please refer to SONS.

PDELAYON Punching with delay

PDELAYON is a preparatory function. This means that PDELAYON is programmed prior to PON. The punching stroke is output with a delay when the programmed end position is reached.

The delay time can be defined in seconds using the following setting data:
SD42400 PUNCH_DWELLTIME.

If the defined value cannot be divided as an integer into the interpolation clock cycle, then it is rounded to the next divisible integer value.

The function has a modal action.

PDELAYOF Punching OFF with delay

PDELAYOF deactivates punching with delay, i. e. the punching process is continued normally. PDELAYON and PDELAYOF form a G code group.

Programming example for PDELAYOF

```
;
N170 PDELAYON X100 SPOF       ; Positioning without punch initiation; activation of
                              ; punch initiation with delay
;
N180 X800 PON                 ; Activate punching. When the end position is reached,
                              ; a punching stroke is output with delay
;
N190 PDELAYOF X700            ; Activate punching with delay; standard
                              ; punch initiation ON. End of programmed motion
```

SPIF1 Activation of the first punching interface

SPIF1 activates the first punching interface. The stroke is controlled through the first pair of fast I/Os (see machine data:
MD26004[0] NIBBLE_PUNCH_OUTMASK,
MD26006[0] NIBBLE_PUNCH_INMASK).

The first punching interface is always active after a reset or control system power up. If only one interface is used, then it need not be programmed.

8.12.1 Expansions to punching and nibbling functions

Automatically activated pre-initiation time

Dead times due to the reaction time of the punching unit can be minimized if the stroke can be initiated before the interpolation window of the axes is reached. The reference time for this is the interpolation end. The stroke is automatically initiated with G603 and delayed by the set value in relation to the time that the end of interpolation is reached.

The machine data: MD26018 NIBBLE_PRE_START_TIME
can be used to set an appropriate delay time in seconds for the stroke initiation.

Select a value of 0.018 s if e. g. a stroke is to be initiated aft an IPO cycle of 9 ms 2 cycles after reaching the end of the interpolation.

In addition, the pre-initiation time can be set in the setting data:
SD42402 NIBPUNCH_PRE_START_TIME.

This setting is only active if the following machine data has been entered:
MD26018 NIBBLE_PRE_START_TIME = 0

Thus, the pre-initiation time set in MD26018 has higher priority.

Monitoring of the input signal

If the “Stroke active” signal is fluctuating between strokes due to plunger overshoots, for example, the message “Illegal punching signal” is output, and the interpolation is stopped.

This message is generated independently of the machine data MD26020 NIBBLE_SIGNAL_CHECK:

MD26020 NIBBLE_SIGNAL_CHECK = 0 → No alarm

Minimum period between two strokes

The setting data SD42404 MINTIME_BETWEEN_STROKES can be used to set a minimum time interval between two consecutive strokes.

If, for example , the interval between two stroke initiations is to be at least 1.3 seconds irrespective of the spatial distance, then set SD42404 MINTIME_BETWEEN_STROKES=1.3.

If a punching dwell time (PDELAYON) is also programmed, then the two times are applied additively. If a pre-initiation time at G603 is programmed, it will be effective only if the end of interpolation is reached before the time set in SD 42404:

The programmed time becomes operative immediately. Depending on the size of the block buffer, strokes that have already been programmed can be executed with this minimum interval. The following programming measures (example) can be taken to prevent this:

```
N..
N100 STOPRE
N110 $SC_MINTIME_BETWEEN_STOKES=1.3
```

The function is not active when SD42404 = 0.

Travel-dependent acceleration

An acceleration characteristic can be defined with PUNCHACC (Smin, Amin, Smax, Amax). This command can be used to define different acceleration rates depending on the distance between holes.

Example 1

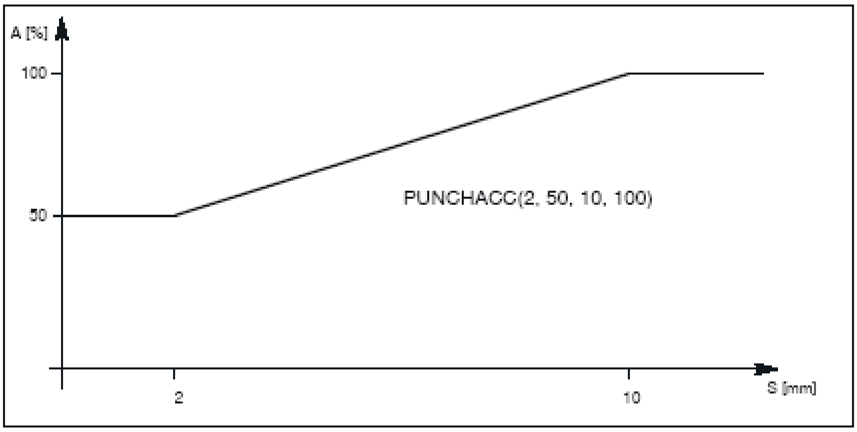


Fig. 8-37 Proportional increase in acceleration from 50 to 100 % with a distance between holes of 2 mm to 10 mm

The characteristic shown in the illustration above defines the following acceleration rate:

- Distance between the holes less than 2 mm: The axis accelerates at a rate corresponding to 50 % of maximum acceleration.
- Distance between the holes between 2 mm and 10 mm: Acceleration is increased to 100 %, proportional to the spacing.
- Distance between the holes greater than 10 mm: The axis accelerates at the maximum rate (100 %).

Example 2

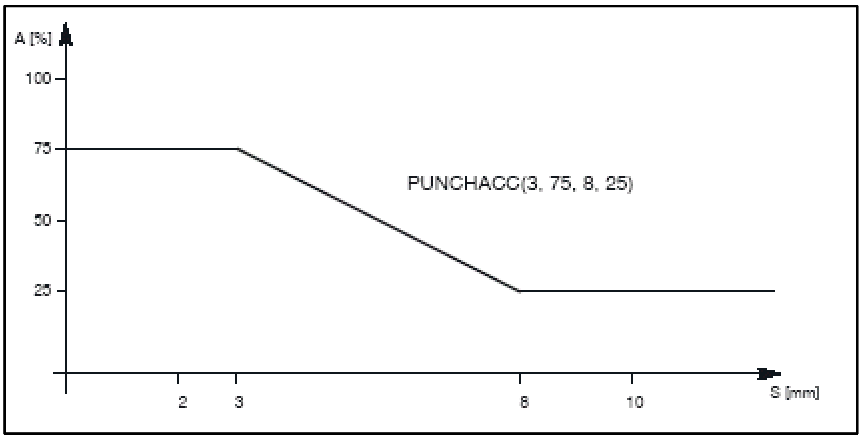


Fig. 8-38 Proportional increase in acceleration from 75 to 25 % with a distance between holes of 3 mm to 8 mm

The characteristic shown in the illustration above defines the following acceleration rate:

- Distance between the holes less than 3 mm: The axis accelerates at a rate corresponding to 75 % of maximum acceleration.
- Distance between the holes between 3 mm and 8 mm: Acceleration is increased to 25 %, proportional to the spacing.
- Distance between the holes greater than 8 mm: The axis accelerates at the maximum rate (25 %).

If a reduced acceleration rate has already been programmed via ACC, then the acceleration limits defined with PUNCHACC refer to the reduced acceleration rate.

The function is deselected with:

Smin = Smax = 0

The acceleration rate programmed beforehand with ACC remains operative.

8.12.2 Compatibility with earlier systems

Use of M functions

As in earlier versions, macro technology allows special M functions to be used instead of language commands (compatibility).

Table 8-4 Compatibility between M function and language command

M function	Language command
M20, M23	SPOF
M22	SON
M25	PON
M26	PDELAYON

Note

M functions can be configured in machine data.

When M functions are assigned to language commands, it must be noted that M functions are organized in auxiliary function groups.

Examples

```
DEFINE M20 AS SPOF                  ; Punching/nibbling OFF
or
DEFINE M20 AS SPOF M=20             ; Punching with auxiliary function output

DEFINE M20 AS SPOF PDELAYOF         ; Punching/nibbling OFF and punching OFF with delay

DEFINE M22 AS SON                   ; Nibbling ON
or
DEFINE M22 AS SON M=22              ; Nibbling ON with auxiliary function output
DEFINE M25 AS PON                 ; Punching ON
or
DEFINE M25 AS PON M=25            ; Punching ON with auxiliary function output

DEFINE M26 AS PDELAYON            ; Punching with delay
or
DEFINE M26 AS PDELAYON M=26       ; Punching and auxiliary function output
```

Programming example

```
:
N100 X100 M20              ; Positioning without punch initiation
N110 X120 M22              ; Activate nibbling; stroke initiation prior to and after the movement
:
N120 X150 Y150 M25         ; Activate punching; stroke initiation at the end of the movement
```

8.13 Automatic path segmentation

Functionality

Path segmentation

When punching or nibbling is active, SPP and SPN cause the total traversing distance programmed for the path axes to be divided into a number of path sections of equal length (equidistant path segmentation). Each path segment corresponds internally to a block.

Number of strokes

When punching is active, the first stroke is executed at the end of the first path segment. In contrast, the first nibbling stroke is executed at the start of the first path segment. The number of punching/nibbling strokes over the total traversing path is thus as follows:

Punching: Number of strokes = number of path segments

Nibbling: Number of strokes = number of path segments + 1

Auxiliary functions

Auxiliary functions are executed in the first of the generated blocks.

Programming

SPP=

or

SPN=

Parameters

Table 8-5

SPP	Size of path section (maximum distance between strokes); modal
SPN	Number of path sections per block; non-modal

Example 1

The programmed nibbling paths must be divided automatically into equidistant path segments.

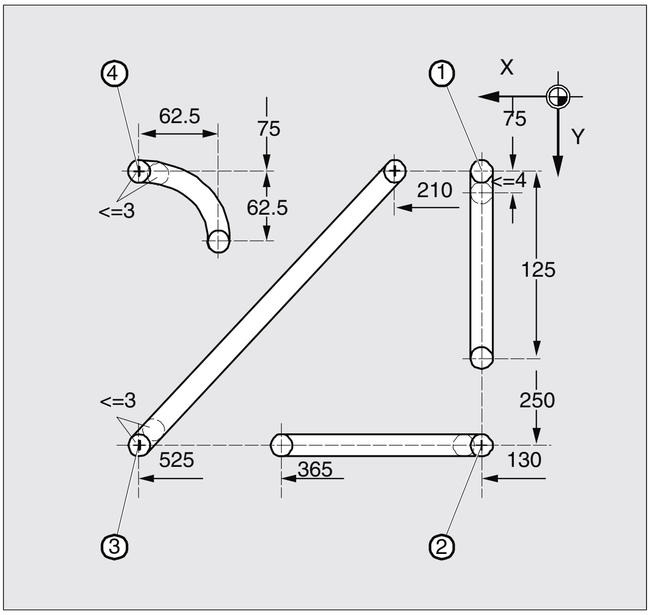


Fig. 8-39

```
N100 G90 X130 Y75 F60 SPOF                ; Positioning to starting point 1
N110 G91 Y125 SPP=4 SON                   ; Nibbling ON; maximum path segment length
                                          ; for automatic path segmentation: 4 mm
N120 G90 Y250 SPOF                        ; Nibbling OFF; positioning to starting point 2
N130 X365 SON                             ; Nibbling ON; maximum path segment length
                                          ; for automatic path segmentation: 4 mm
N140 X525 SPOF                            ; Nibbling OFF; positioning to starting point 3
N150 X210 Y75 SPP=3 SON                   ; Nibbling ON; maximum path segment length
                                          ; for automatic path segmentation: 3 mm
N160 X525 SPOF                            ; Nibbling OFF; positioning to starting point 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON   ; Nibbling ON; maximum path segment length
                                          ; for automatic path segmentation: 3 mm
N180 G00 G90 Y300 SPOF                    ; Nibbling OFF
```

Example 2

Automatic path segmentation is to be used to create the individual rows of holes. The maximum path segment length (SPP value) is specified in each case for segmentation purposes.

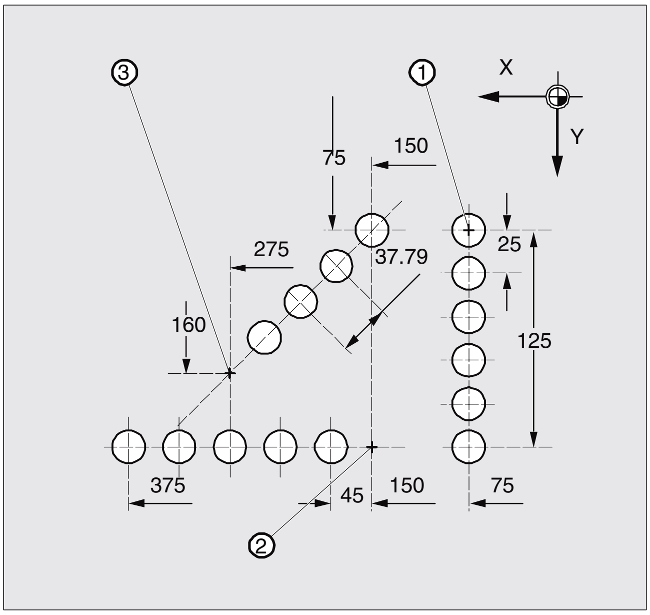


Fig. 8-40

```
N100 G90 X75 Y75 F60 PON          ; Positioning to starting point 1;
                                  ; Punching an individual hole
N110 G91 Y125 SPP=25              ; Maximum path segment length for
                                  ; automatic path segmentation: 25 mm
N120 G90 X150 SPOF                ; Nibbling OFF; positioning to starting point 2
N130 X375 SPP=45 PON              ; Nibbling ON; maximum path segment length
                                  ; for automatic path segmentation: 45 mm
N140 X275 Y160 SPOF               ; Punching OFF; positioning to starting point 3
N150 X150 Y75 SPP=40 PON          ; Punching ON; instead of the programmed
                                  ; path segment length of 40 mm, the calculated
                                  ; path segment length of 37.79 mm is used.
N160 G00 Y300 SPOF                ; Punching OFF; positioning
```

8.13.1 Path segmentation for path axes

Length of SPP path segment

SPP is used to specify the maximum distance between strokes and thus the maximum length of the path segments in which the total traversing distance is to be divided. The command is deactivated with `SPOF` or `SPP=0`.

Example:

```
N10 SON X0 Y0
N20 SPP=2 X10
```

The total traversing distance of 10 mm will be divided into five path sections each of 2 mm (SPP=2).

Note

The path segments effected by SPP are always equidistant, i.e. all segments are equal in length. In other words, the programmed path segment size (SPP setting) is valid only if the quotient of the total traversing distance and the SPP value is an integer. If this is not the case, the size of the path segment is reduced internally such as to produce an integer quotient.

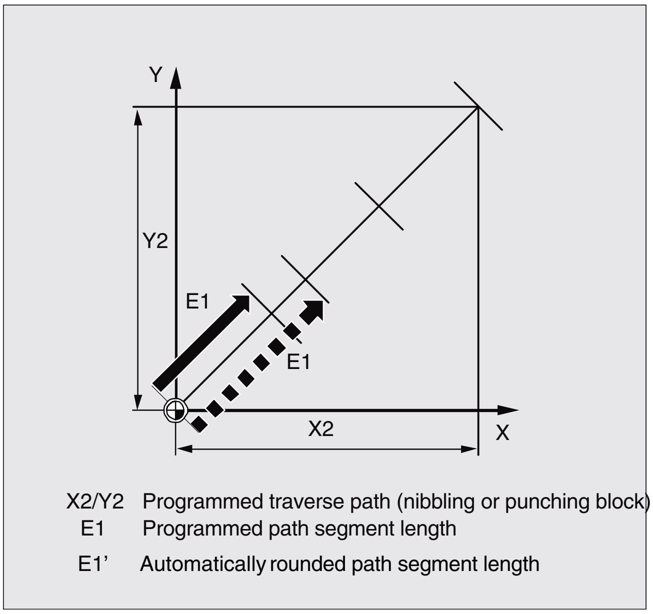

Y
Y2
E1
E1
X2
X
X2/Y2 Programmed traverse path (nibbling or punching block)
E1 Programmed path segment length
E1' Automatically rounded path segment length

Fig. 8-41

Example:

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

When the total traversing distance is 15 mm and the path segment length 3.5 mm, the quotient is not an integer value (4.28). In this case, the SPP value is reduced down to the next possible integer quotient. The result in this example would be a path segment length of 3 mm.

Number of SPN path segments

SPN defines the number of path segments to be generated from the total traversing distance. The length of the segments is calculated automatically. Since `SPN` is non-modal, punching or nibbling must be activated beforehand with `PON` or `SON` respectively.

SPP and SPN in the same block

If you program both the path segment length (SPP) and the number of path segments (`SPN`) in the same block, then `SPN` applies to this block and `SPP` to all the following blocks. If `SPP` was activated before `SPN`, then it takes effect again after the block with `SPN`.

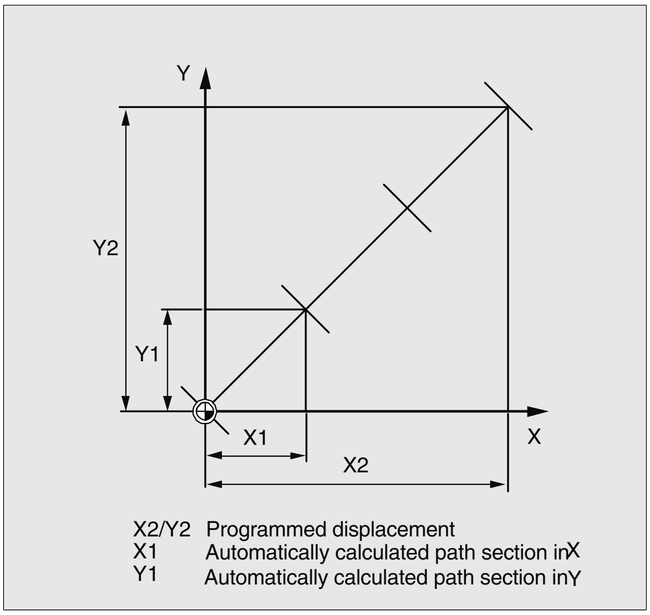

Y
Y2
Y1
X1
X2
X
X2/Y2 Programmed displacement
X1 Automatically calculated path section inX
Y1 Automatically calculated path section inY

Fig. 8-42

Note

Provided that punching/nibbling functions are available in the control, then it is possible to program the automatic path segmentation function with SPN or SPP even when the punching/nibbling functions are not in use.

8.13.2 Path segmentation for single axes

If single axes are defined as punching/nibbling axes in addition to path axes, then the automatic path segmentation function can be activated for them.

Response of single axis to SPP

The programmed path segment length (SPP) basically refers to the path axes. For this reason, the SPP value is ignored in blocks which contain a single axis motion and an SPP value, but not a programmed path axis.

If both a single axis and a path axis are programmed in the block, then the single axis responds according to the setting of the appropriate machine data.

1. Standard setting

   The path traversed by the single axis is distributed evenly among the intermediate blocks generated by SPP.

   Example:

   N10 G1 SON X10 A0
   N20 SPP=3 X25 A100

   As a result of the programmed distance between strokes of 3 mm, five blocks are generated for the total traversing distance of the X axis (path axis) of 15 mm.

   The A axis thus rotates through 20° in every block.

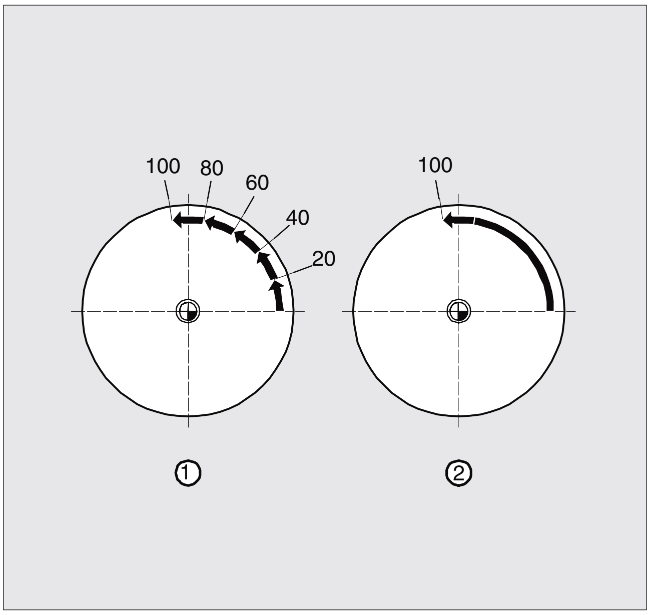


Fig. 8-43

2. Single axis without path segmentation

   The single axis traverses the total distance in the first of the generated blocks.

3. With/without path segmentation

   The response of the single axis depends on the interpolation of the path axes:

   – Circular interpolation: Path segmentation

   – Linear interpolation: No path segmentation

Response to SPN

The programmed number of path segments is applicable even if a path axis is not programmed in the same block.

Prerequisite: The single axis is defined as a punching/nibbling axis.

Index

Notes

To
SIEMENS AG
A&D MC BMS
Postfach 3180

91050 ERLANGEN, GERMANY

(Tel. +49 (0) 180 5050 – 222 [Hotline]
Fax +49 (0) 9131 98 – 63315 [Documentation]
email: motioncontrol.docu@siemens.com

Suggestions

Corrections

For Publication/Manual:

SINUMERIK 802D sl

User Documentation

Operation and Programming
Nibbling
Order No.: 6FC5398-3CP10-0BA0
06/06 Edition

From

Name

Company/dept.

Address

Postal code: City:

Phone: /

Fax: /

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